SPRING '89 COMPUTER DEALERS EXPO • COMDEX

FDC 50078

ST LOG

THE ATARI ST MONTHLY MAGA

AUGUST 1989 ISSUE 35

DISK VERSION $12.95

ST • BINGO

CALORIE COUNTER

MENUS WITH PASCAL

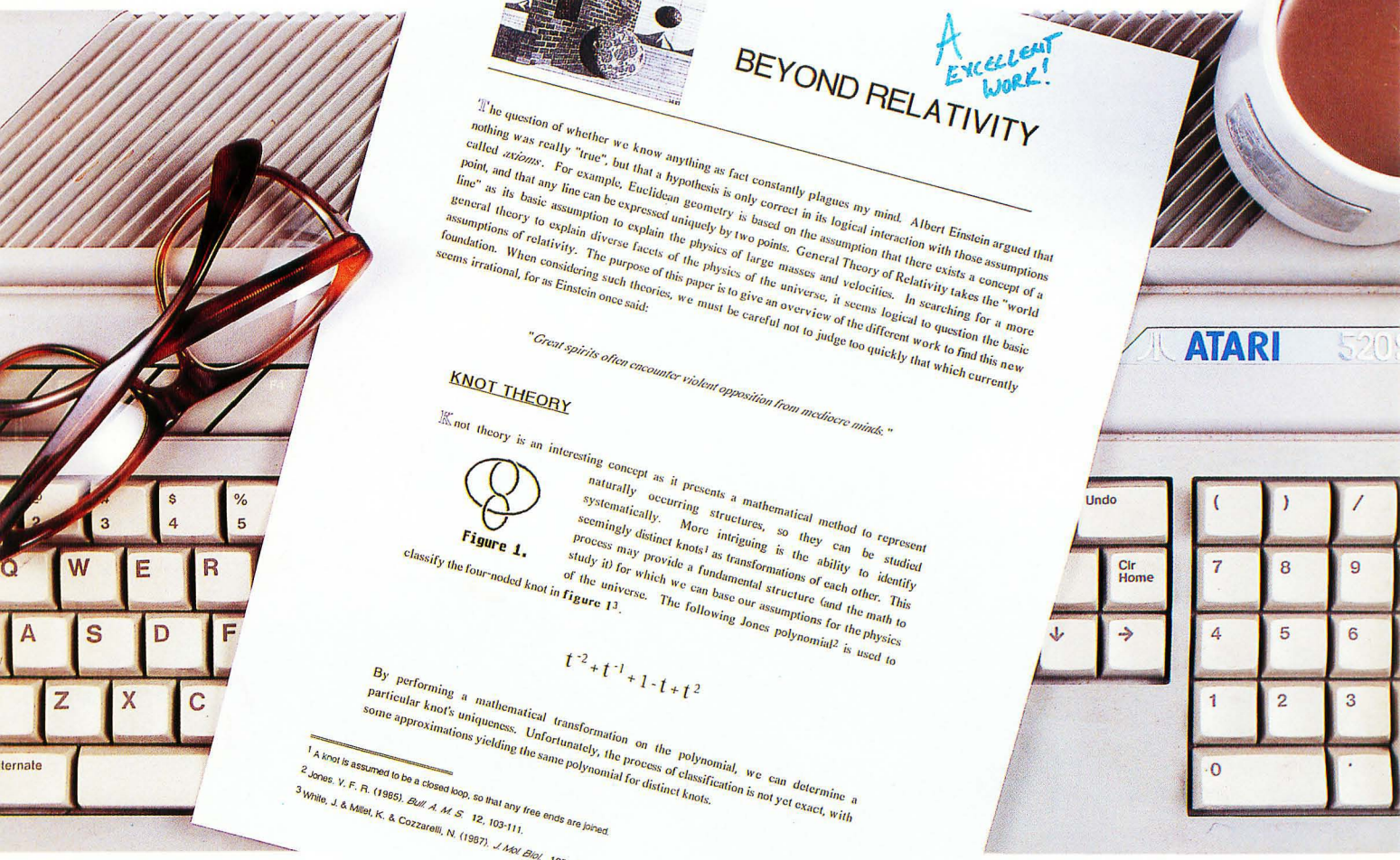
REVIEWS

Fleet Street Publisher
R/C Aerochopper
Operation Cleanstreets
Zynaps
Roadwars

08
74369 50078 5

WordUp...

NEW VERSION
STILL $79.95



BEYOND RELATIVITY

A EXCELLENT WORK!

The question of whether we know anything as fact constantly plagues my mind. Albert Einstein argued that nothing was really "true", but that a hypothesis is only correct in its logical interaction with those assumptions called *axioms*. For example, Euclidean geometry is based on the assumption that there exists a concept of a point, and that any line can be expressed uniquely by two points. General Theory of Relativity takes the "world line" as its basic assumption to explain the physics of large masses and velocities. In searching for a more general theory to explain diverse facets of the physics of the universe, it seems logical to question the basic assumptions of relativity. The purpose of this paper is to give an overview of the different work to find this new foundation. When considering such theories, we must be careful not to judge too quickly that which currently seems irrational, for as Einstein once said:

"Great spirits often encounter violent opposition from mediocre minds."

KNOT THEORY

Knot theory is an interesting concept as it presents a mathematical method to represent naturally occurring structures, so they can be studied systematically. More intriguing is the ability to identify seemingly distinct knots[1] as transformations of each other. This process may provide a fundamental structure (and the math to study it) for which we can base our assumptions for the physics of the universe. The following Jones polynomial[2] is used to classify the four-noded knot in **figure 1**[3].

Figure 1.

$$t^{-2} + t^{-1} + 1 - t + t^{2}$$

By performing a mathematical transformation on the polynomial, we can determine a particular knot's uniqueness. Unfortunately, the process of classification is not yet exact, with some approximations yielding the same polynomial for distinct knots.

[1] A knot is assumed to be a closed loop, so that any free ends are joined.

[2] Jones, V. F. R. (1985). *Bull. A. M. S.* 12, 103-111.

[3] White, J. & Millet, K. & Cozzarelli, N. (1987). *J Mol Biol* 197, 585-603.

...Impresses

WordUp is the only word processor for the Atari ST/MEGA™ that can integrate multiple fonts and pictures.

For the dealer nearest you
Call (805) 482-4446

CIRCLE #101 ON READER SERVICE CARD.

NEOCEPT, Inc.
547 Constitution #A • Camarillo • CA 93010

The language of the above sample document is not intended to be factual. Document was created with WordUp and printed with an Atari Laser. WordUp is a trademark of Neocept, Inc.

EDITORIAL

BY CLAYTON WALNUM

Twice a year, the people and companies that make up the computer marketplace get together to show off their new ideas and wares at the Computer Dealers Exposition (COMDEX). This show differs from that other famous semiannual event, the Consumer Electronics Show (CES), in that the exhibitors at COMDEX are made up solely of computer-related dealers, both software and hardware. And while COMDEX is a smaller show than the CES, which covers the entire electronics industry, it is more important to a computer manufacturer. CES more typically focuses on the recreational electronics, which in the case of computers means games; whereas COMDEX focuses on the hardware itself. For a computer manufacturer who's serious about his business, COMDEX is the place to be.

At this year's spring COMDEX in Chicago, Atari made a better impression than they did at the winter show. Although the new products that Atari displayed in Chicago were not exactly surprises, at least the company had a strong presence and gained some notice from industry pundits.

Of course, ST-LOG made an appearance at the spring COMDEX, as well, and you'll find our report in this issue. We're sure you'll be pleased to read of the new products Atari will offer in the near future, as well as some of the goods being shown by the many third-party developers present at the show.

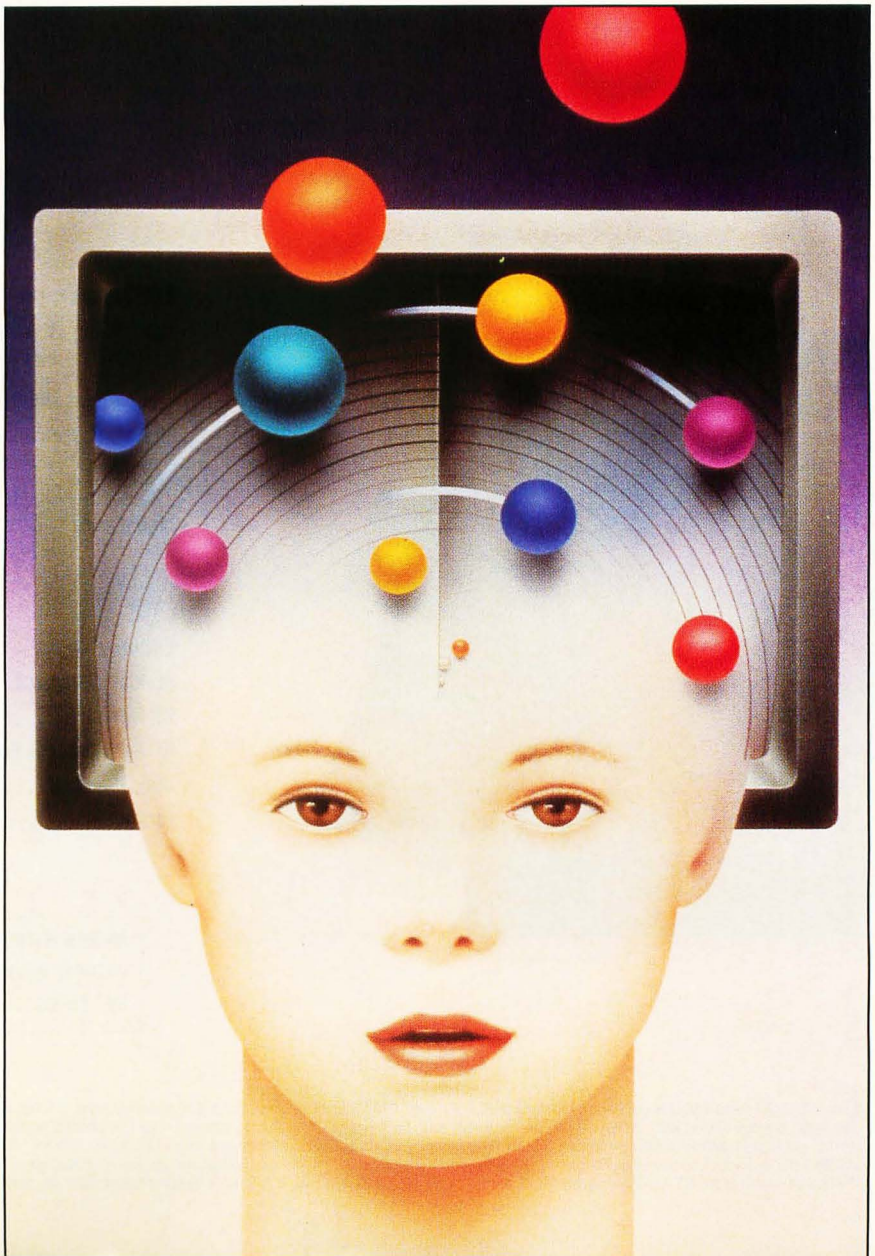
Also in this issue, we have the second installment of our new column, *The Compukid Connection*, by D.A. Brumleve. Ms. Brumleve's sensational children's programs have gained her praises from both here and abroad, and we're proud to have her as a contributing editor. Each month she will discuss some topic on children and their interaction with computers and present a commercial-quality program that will delight the little ones in your household.

In addition, Karl Wiegers continues his popular "Software Engineering" series with a discussion of structured programming, and frequent contributor, Albert Baggetta, gives us his computerized version of the venerable game Bingo.

There are plenty of reviews this time around, too, both for the serious computerist and the fun-loving gamer. And simulation fans should be sure to read Steve Panak's "Test Diving Four Submarines," wherein he gives a brief overview of four software titles that put you in command of your own submarine.

Whether you use your computer to generate newsletters for your business or to blast evil aliens from some faraway galaxy, you'll find much to delight you in this issue of ST-LOG.

Let's get started, shall we? ▪

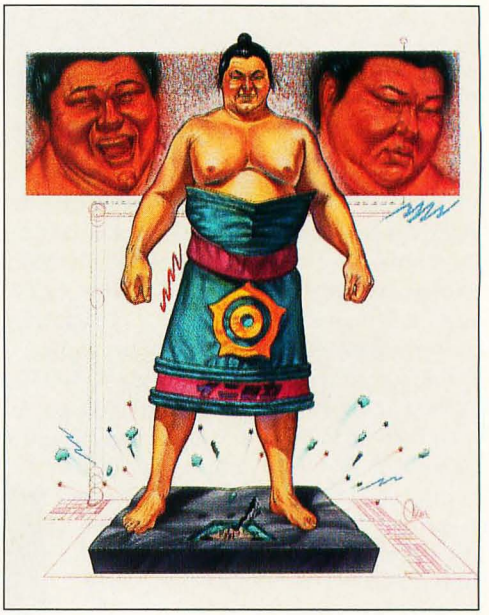


IN THIS ISSUE

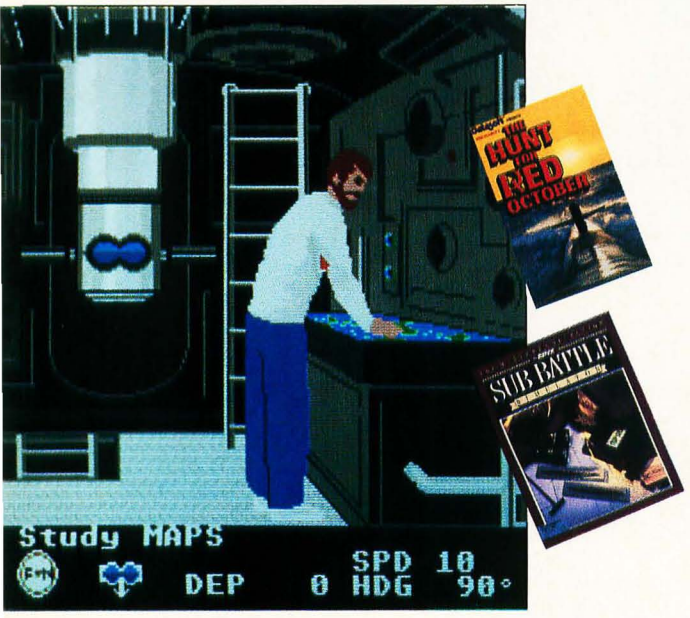
FEATURES

COLUMNS

CALORIE COUNTER 12



TEST DIVING REVIEW 82



REVIEWS

DEPARTMENTS

PROGRAM · LISTING · GUIDE

ST-LOG (ISSN 0890-9601) is published monthly by L.F.P., Inc., 9171 Wilshire Blvd., Suite 300, Beverly Hills, CA 90210.

U.S. subscription $28 for one year (12 issues), $52 for two years (24 issues), $76 for three years (36 issues). Foreign subscriptions: add $10 per subscription per year. Single copy $3.50 (add $1 for postage). Change of address: six weeks advance notice, and both old and new addresses are needed. POSTMASTER, send change of address to: **ST-LOG** Magazine, P.O. Box 16928, North Hollywood, CA 91615. Second-class postage paid at Beverly Hills, CA, and additional mailing offices.

AUGUST 1989
ISSUE 34

ST LOG

STAFF

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

ADVERTISING SALES

Correspondence, letters and press releases should be sent to: Editor, **ST-LOG**, 9171 Wilshire Blvd., Suite 300, Beverly Hills, CA 90210.

Correspondence regarding subscriptions, including problems and changes of address, should be sent to: **ST-LOG,** P.O. Box 16928, North Hollywood, CA 91615 or call (818) 760-8983.

Correspondence concerning a regular column should be sent to our editorial address (see Authors below), 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.

J.E. Publishers Representatives — Los Angeles: (213) 467-2266.
6855 Santa Monica Blvd., Suite 200, Los Angeles, CA 90038.
San Francisco: (415) 864-3252. Chicago: (312) 445-2489.
Denver: (303) 595-4331.
New York: (212) 724-7767.

Address all advertising materials to: Paula Thornton — Advertising Director, **ST-LOG**, 9171 Wilshire Blvd., Suite 300, Beverly Hills, CA 90210.

PERMISSIONS

SUBSCRIPTIONS

ST-LOG, P.O. Box 16928, North Hollywood, CA 91615 or call (818) 760-8983. Payable in U.S. funds only. U.S.: $28.00-1 year; $52.00-2 years; $76.00-3 years. Foreign: add $7.00 per year per subscription. 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 with your manuscript to **ST-LOG**, P.O. Box 1413-MO, Manchester, CT 06040-1413.

COVER PHOTOGRAPHY:
MICHAEL TCHEREVKOFF

INTERIOR GRAPHICS:
DIGITAL ART

ILLUSTRATORS:
ALAN HUNTER
JOHN BERADO
JAVIER ROMERO
STANISLAW FERNANDES

THANKS TO:
IMAGE BANK WEST
WESTLIGHT

READER COMMENT

Skipping columns

This letter is to congratulate you on a fine magazine, a magazine that I thoroughly enjoy each month. However, I've noticed that you've been skipping some of your tutorial columns in the past few months—most notably *C-manship* and *Assembly Line.*

I know you've received some mail from nonprogrammers stating they don't like these kinds of columns and telling you to leave the technical stuff out. I'd like to voice my opinion as strongly as possible in the opposite direction. *C-manship* is one of the best tutorial columns I've ever seen. It does a thorough job of explaining things that most of the books and user's manuals leave out. The same has been true for *Assembly Line.*

I know you can't please all of your readers, but I think there are a lot of us who study the tutorials, type in the examples and try to use the technical help you give us each month. No other Atari magazine does this to my knowledge. If you do anything, please expand your programming tutorials and make it a priority that they appear in ST-LOG each month.

—Chris Wareham
Spearfish, SD

Thank you, Chris, for the kind words. We print the programming tutorials as often as possible, but space and time restraints frequently force us to skip over them. For example, because editing this magazine takes such a huge amount of my time, I frequently find myself unable to complete a* C-manship *every month. The same holds true for* Assembly Line, *which is written by Charles F. Johnson. Charles is heavily involved in his own software company, CodeHead Software, as well as being a professional guitarist with frequent studio responsibilities.

Also, because* ST-LOG *must cover such a wide variety of subjects, even if Charles and

***I could manage to write a column every month, there wouldn't always be space for it. For example, this issue, due to the COMDEX special report, you'll find that* C-manship *and* Assembly Line *are again missing. But we have included several program listings which should be invaluable to people wanting to learn the languages they represent. There's also a tutorial on using menu bars in Pascal. We hope you'll find that these items more than satisfy your technical curiosity until next month, when, if all goes well,* C-manship *and* Assembly Line *will return again.* ■

Games and hard disks

I am one of many ST users and ST-LOG subscribers here in Saudi Arabia. I recently purchased a Supra 20-Meg hard drive and am having a problem with it.

I am unable to boot game programs that have AUTO-boot folders with the hard drive plugged into the computer. It doesn't matter if the drive is turned on or not. The ST Drive A doesn't read the floppy if the hard disk is connected to the computer. The GEM desktop comes up in low resolution with no disk icons. Even when I disable the hard drive, I cannot play my games. The only way I can get the computer to load my game disks is to physically disconnect the hard drive from the computer. Other friends of mine do not seem to have this problem. Can you offer any help? —Gilbert Snell
Saudi Arabia

It sounds to us like you're using Supra's self-booting software. This program is provided for Supra owners so that they can boot the computer from Drive C (the hard drive itself) rather than from floppy Disk A, thus speeding up the loading of AUTO programs and desk accessories. In order to load AUTO programs from Drive A, you have to disable the self-booting software. But don't despair; this is easy to do.

If you have any ST other than a MEGA, all you have to do is hold down the Alternate, Shift and Control keys when turning on the computer. The hard disk will then be ignored and the computer will boot from Floppy A. If you own a MEGA, you must wait until the Drive A light comes on before pressing the three above keys. ■

Outline Plus update

I recently discovered a bug in the ST version of *Outline Plus.* If the vertical scroll bar is used to reposition the window, and the new cursor position is on an empty line, the program will crash. Scrolling the window with the arrow keys, page-up or page-down keys, or clicking in the shaded part of the vertical scroll bar to page-up or page-down all work fine.

One line needs to be added to the program to prevent this problem. In the procedure *Wind_VSlide,* the following line should be added:

```
if XPos[WindNo] < 1 then XPos[WindNo] := 1 ;
```

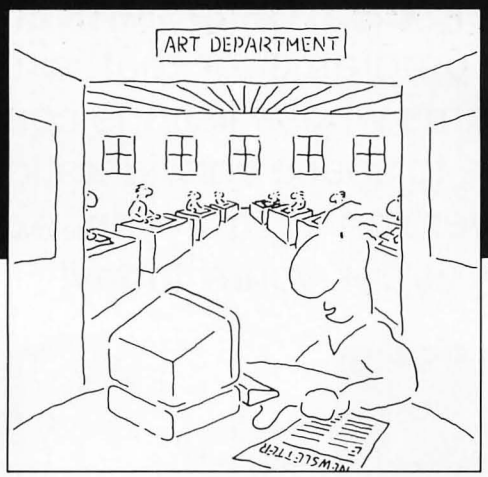
The above code should be added immediately following this line:

```
if XPos[WindNo] > Length(Current[WindNo]^.NewWord) then
   XPos[WindNo] := Length(Current[WindNo]^.NewWord) ;
```

—James Maki
Indianapolis, IN

ALL LETTERS TO BE CONSIDERED FOR PUBLICATION SHOULD BE ADDRESSED TO:
ST-LOG, READER COMMENT
P.O. BOX 1413-M.O.
MANCHESTER, CT 06040-1413

Open your own art department.



If you're a desktop publisher with big ideas and a small crew, let Migraph staff the desktop art department of your dreams.

Picture this: Professionally drawn images and illustrations at your fingertips. Powerful drawing tools, extensive editing tools, and a complete paint program at your command.

Plus high-quality printouts. Every time.

All that and compatibility, too. Migraph files load easily into your favorite Atari ST publishing programs.

Powerful. Versatile. And easy to use. Migraph's the ideal candidate for every job in your graphics department.

Touch-Up™ The complete design tool for high-resolution monochrome images.

Easy-Draw™ The professional object-oriented drawing program.

Supercharged Easy-Draw™ Easy-Draw plus basic publishing features.

Easy-Tools™ A 5-tool GEM desk accessory to enhance Easy-Draw.

DrawArt Professional™ A library of over 150 professional line art drawings.

Scan Art™ A library of over 100 high-resolution, bit-mapped images.

Border Pack A library of over 40 attention-getting border designs.

OSpooler A configurable background file spooler and print buffer.

Whatever desktop graphics project you have in mind—be it big or small, simple or ornate, traditional or avant-garde—Migraph's got you covered.

See your Atari ST™ dealer today for more details.

MIGRAPH INC™

200 S. 333rd St., Suite 220 Federal Way, WA 98003 800/223-3729 206/838-4677

CIRCLE #102 ON READER SERVICE CARD.

YOU NEED THE DISK!

If you want to get the most out of ST-LOG, you're also going to want to get your copy of the disk. Each issue's disk contains all the exciting programs for that issue, including the programs whose listings could not be included due to space considerations. The ST-LOG disk version is truly an excellent software value. Order yours today!

ONLY $9.95 EACH!

AUGUST 1989	ST Bingo, Calorie Counter, Menus with Pascal and more!	JUNE 1989	Ballbuster, MIDI Capture Safe Keeping and more!	APRIL 1989	Multi-Paint, Font ID Editor, Monochrome Gray and more!
JULY 1989	MicroCheck ST, Game Cupboard, Animated GFA Input and more!	MAY 1989	Line Attack, Outline Plus, Monochrome Gray II and more!	MARCH 1989	Picture-Puzzle, Sounds-A-Like, ChemCalc and more!

YES I DO WANT THE DISK! ONLY $9.95 EACH

- ☐ ST-LOG March 1989 DISK
- ☐ ST-LOG April 1989 DISK
- ☐ ST-LOG May 1989 DISK
- ☐ ST-LOG June 1989 DISK
- ☐ ST-LOG July 1989 DISK
- ☐ ST-LOG August 1989 DISK

Name_____

Address_____

City_____ State_____ Zip_____

☐ Payment Enclosed—Charge My ☐ VISA ☐ MC

#_____ Exp._____

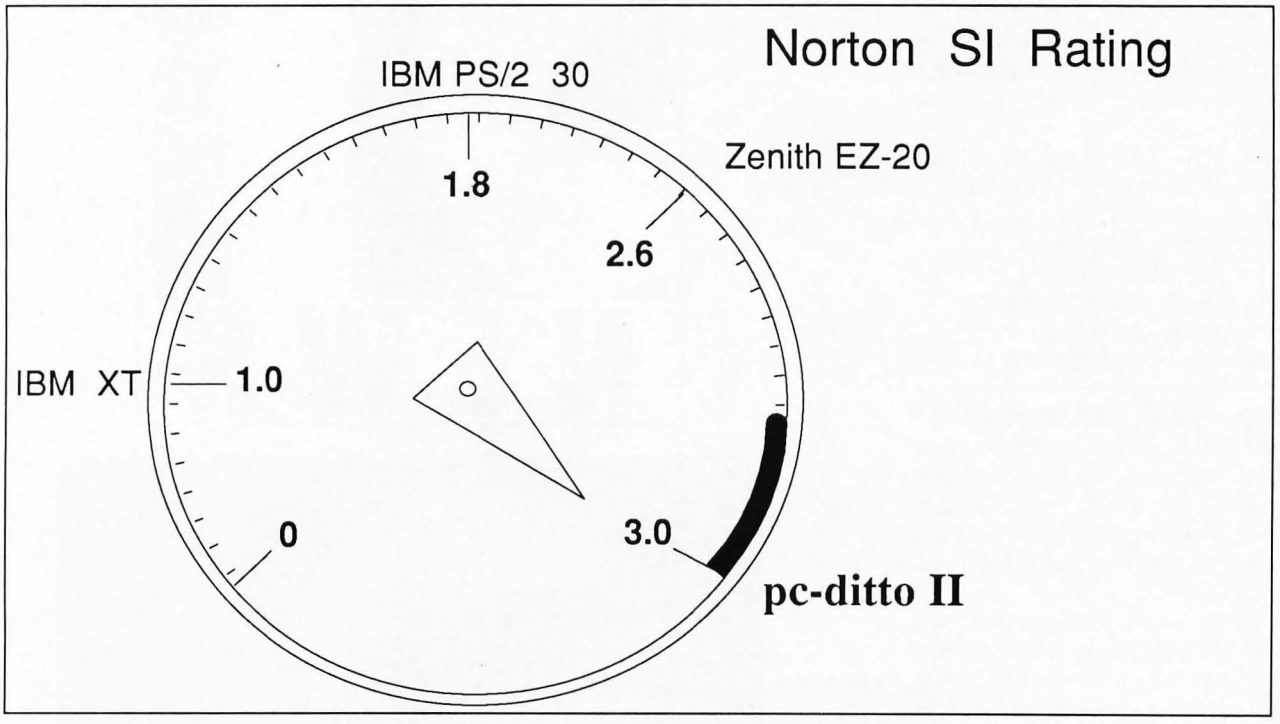
Signature_____

Add $1.50 postage and handling for each disk ordered.
Make check payable to: LFP, Inc. P.O. Box 67068, Los Angeles, CA 90067
California residents add 6.5%

ST LOG
THE ATARI ST MONTHLY MAGAZINE

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**_____

CIRCLE #103 ON READER SERVICE CARD.

The ST goes video

Regent Software has just taken on an unusual project. They are currently producing a series of training videos to help ST users with popular ST software and hardware. The 30- to 60-minute videotapes will provide highlights of such products as *Word Up*, *G+Plus*, *PageStream*, *Revolver*, *GFA BASIC 3.0* and *HiSoft BASIC*.

"The project started as an easy method of delivering news and information to ST users' groups," says Frank Cohen, president of Regent Software. "In the past, Regent has received a bunch of letters from ST users' groups asking for a videotape that could be shown at users' group meetings."

Dubbed "The Computer Tutor," the videotapes will be priced at $19.95.

Regent Software
P.O. Box 14628
Long Beach, CA 90803-1208
(213) 439-9664

CIRCLE #130 ON READER SERVICE CARD.

State-of-the-art paint program

ArtisTech Development has announced the coming release of their paint program, *Da Vinci Designer*, which provides computer artists with more functions than any other paint program currently on the market.

Some of *Da Vinci*'s features include a "real time" magnify mode that works with every mode and tool. When using real-time magnify, the magnified portion of the picture is constantly updated according to the position of the mouse, almost as if a magnifying glass were actually being moved across the picture.

Da Vinci's stencil function provides the ability for specific colors to be "protected"; that is, subsequent shapes that are drawn on the screen will not affect the protected colors, giving the appearance that they are being drawn behind the protected colors.

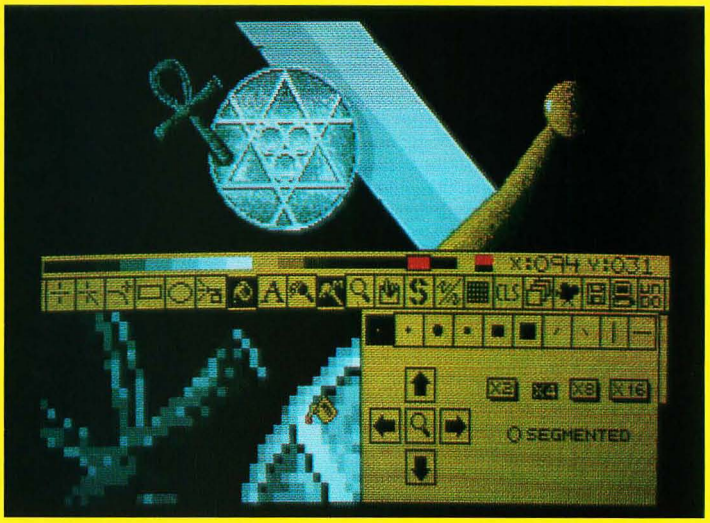
The many powerful functions provided by *Da Vinci* are too numerous to list here. But, as well as all the more common graphics functions, artists are also provided with coordinate and grid lock, 3-D distort, bend, rotate, flip, smear, cycle, blend, tint and filter.

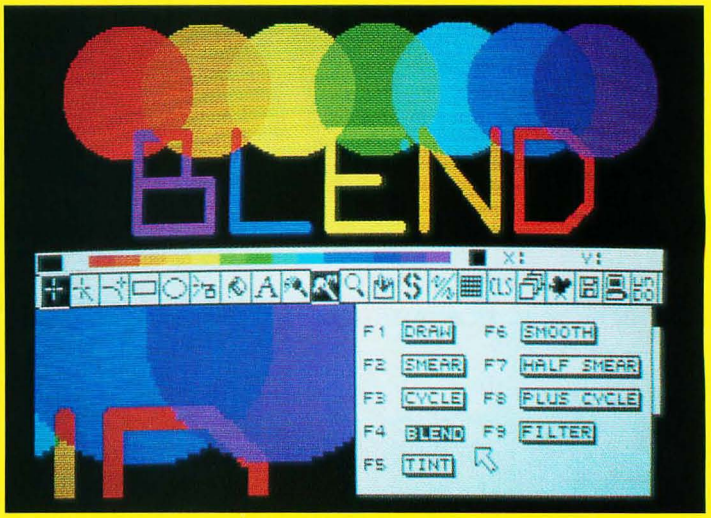
Developed completely in assembly language and ignoring almost all system calls, this program is *fast*. *Da Vinci Designer* will be available July 31 at a retail price of $99.95.

ArtisTech Development
P.O. Box 214830
Sacramento, CA 95821
(916) 488-6844

CIRCLE #131 ON READER SERVICE CARD.

ST NEWS





Space pirates

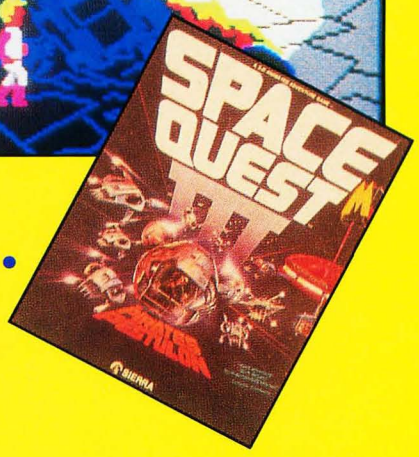
Now available from Sierra On-line, Inc. is *Space Quest III: The Pirates of Pestulon*, starring space janitor extraordinaire, Roger Wilko.

In this chapter of the continuing space drama, the authors, Mark Crowe and Scott Murphy, have written themselves into the game as the two guys who must be rescued before they are forced to work for the evil software manufacturer ScumSoft.

The music for *Space Quest III* was written and performed by Bob Siebenberg, a member of the rock group Supertramp. As with other games created with Sierra's SCI (Sierra Creative Interpreter), MIDI output directly supports the Roland MT-32 and the Casio CZ-101 and MT-540.

According to Sierra, "You need not have played *Space Quest I* or *II* to play *Space Quest III*, but Scott and Mark could use some extra royalty money."

Space Quest III, on three 3½-inch disks, sells for $59.95



Sierra On-line, Inc.
P.O. Box 485
Coarsegold, CA 93614
(209) 683-4468

CIRCLE #132 ON READER SERVICE CARD.

Professional CAD

BeckerCAD ST, published by Abacus, is the next in a long line of computer-aided design programs for the ST. This program boasts a multilayer capability with up to 300 individually selected layers; a fully integrated programming language for exact object definition; symbol libraries; text capabilities; and plotter, dot-matrix printer, laser printer and PostScript support.

According to Abacus, "*BeckerCAD*'s well-thought-out interface makes it easy to design anything from simple sketches, to detailed small parts to large complex buildings. *BeckerCAD*'s standard features include polygons with any number of sides, freehand drawing, automatic line measurement, definable grid, multiple line widths, trim, divide, zoom and undo operations."

The suggested retail price for *BeckerCAD ST* is $395.

Abacus
5370 52nd Street S.E.
Grand Rapids, MI 49508
(616) 698-0330

CIRCLE #133 ON READER SERVICE CARD.

More from Artworx

Artworx Software, publisher of the popular *Strip Poker*, has just released three new games for the ST.

Colossus Chess X provides four different sets of chess pieces, as well as allows players to view the board from any angle. The program also features six different playing modes and an opening book of over 11,000 records. One thing that makes *Colossus* unique is its ability to learn from past playing experiences. *Colossus Chess X* is priced at $34.95.

Tank Attack is a combination of a strategy board game and a real-time action game, and allows two, three or four players. Players try to capture their opponent's headquarters by strategically deploying their forces. The results of battles, as well as the weather predictions for the next day, are given after each move in a computerized daily newspaper. To add some spice to play, *Tank Attack* allows players to form alliances, which allows them to share rebuilding facilities. *Tank Attack* is $29.95.

Finally, *World Snooker* offers five different types of pool-type games: two versions of snooker, two types of pool and two types of billiards. Players may zoom in to align shots, as well as take back missed shots. Also, for trick shots the table may be set up in any way the players like. Further, *World Snooker* provides instant replays and four modes of play, including a practice mode that supplies shot prediction, where the computer will predict a ball's path. *World Snooker* can be purchased for $24.95.

Artworx Software Company, Inc.
1844 Penfield Road
Penfield, NY 14526
(716) 385-6120

CIRCLE #134 ON READER SERVICE CARD.

ILLUSTRATION: **ALAN HUNTER**

ORIE COUNTER

BY
RON SCHAEFER, M.D.
AND
KATHY SCHAEFER, M.D.

Approximately 40% of Americans are overweight, and they spend billions of dollars each year trying to shed these unwanted pounds. In addition to a decrease in self esteem, obesity is a tremendous health problem that can cut years from your life. One of the keys to successful dieting is to limit the number of calories that you eat. But totaling up the number of calories that you have eaten can be a tiresome process.

Calorie Counter to the rescue! This short program will make counting your calories a snap—just point and click. *Calorie Counter* will work in medium or high resolution on any Atari ST.

To run the program, double-click on the CAL__CNT.PRG file (or run Listing 1 with GFA BASIC 2.0). The opening screen will appear, giving you the choice of: Count = counting calories, Help = display a help screen, or Quit = end the program. If you select Count, the program will automatically load the file CALORIE.DAT if it is in the same directory as the program. If it is not, an item selector will appear, asking you to select the file that contains the calorie database.

Calorie Counter allows you to plan a menu for a meal, a day or a week at a time. After the program has read in the data from the file CALORIE.DAT, you will be asked to input the number of desired calories. This number will serve as a goal, and the computer will categorize the calories in terms of grams of protein, fat and carbohydrates.

Next the Menu Planner screen will be displayed. At the top you will see your goal number of calories, grams of protein, fat and carbohydrate, and below this the running total. Your goal is based on the ideal American diet and has the following breakdown: protein (12%), fat (30%) and carbohydrate (58%). In the center of

the screen are the food items that can be selected. Selection is done with the mouse, clicking the left mouse button to add an item and clicking the right button to subtract the item from the total. Once an item is selected it will turn red if you are using a color monitor or bold face if you are using monochrome. More than one of a given item can be added to the total by clicking repeatedly on that item; the quantity column will tell you how many of an item you have selected.

At the bottom of the Menu Planner screen are four other options: NEXT PAGE, LAST PAGE, LIST and QUIT. Clicking on NEXT PAGE will display the next page of food items, LAST PAGE will display the previous page of food items. LIST will allow you to create a list of the food items that you have selected. This list can then be displayed on the screen or printed out so that you can take it with you when you go shopping. The QUIT option will take you back to the opening screen.

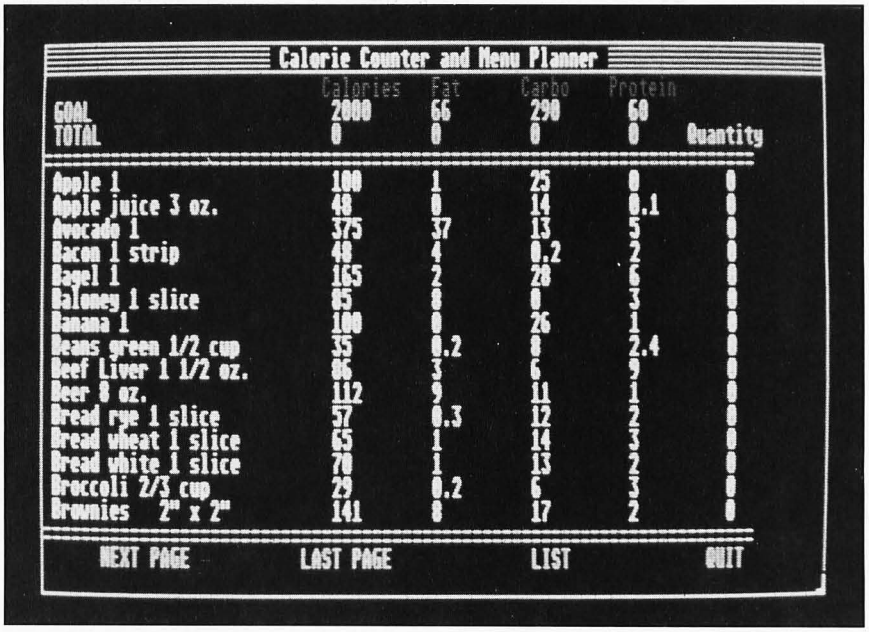
The food item data file, CALORIE.DAT, can be easily added to by using any word processor with an ASCII mode. (This month's disk version contains the CALORIE.DAT file; if you don't have the disk version, you'll have to create this file yourself.) The first line of the file contains the number of food items in the file. This is followed by a listing of the various food items. It is important that the same format be used when adding a food item to the list. The first column contains the food item and must be 20 characters long; add spaces to the end of the name if necessary. This is followed by the calories, the grams of protein, the grams of fat and the grams of carbohydrates; each of these items separated by commas. This type of dietary information can be found in almost all books on dieting and nutrition and allows you to individualize the list to include your food preferences.

The following is a short sample of a CALORIE.DAT file:

```
6
Apple 1                ,100,0,1,25
Apple juice 3 oz.      ,48,.1,0,14
Avocado 1              ,375,5,37,13
Bacon 1 strip          ,48,2,4,.2
Bagel 1                ,165,6,2,28
Baloney 1 slice        ,85,3,8,0
```

I would recommend keeping the food items in alphabetical order. The same item could be placed under two different names; for instance, Big Mac under the letter B and also under the letter H for hamburger. The only factor limiting the number of food items that can be in this list is the amount of memory in your computer.

Calorie Counter to the rescue! This short program will make counting your calories a snap—just point and click. *Calorie Counter* will work in medium or high resolution on any Atari ST.



About the source code

This program was written and compiled with GFA BASIC 2.0. If you have Version 3.0, you will need to modify the open window commands as explained in your GFA 3.0 manual. I have tried to put enough comments in the program listing to make it easy to follow.

Basically, the file CALORIE.DAT is loaded into multiple arrays:

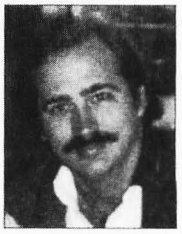
N$(T%) = food item name
CALS(T%) = number of calories
FATS(T%) = grams of fat
CARBOS(T%) = grams of carbohydrates
PROTS(T%) = grams of protein
QUANT(T%) = the quantity of a given food item

This information is then displayed on the screen. The position of the mouse is determined in relation to which food item it is over, and when a button is pressed the quantity of that item is updated. To print out a list of the selected food items on a printer the LPRINT command is used.

To allow the program to run in both medium and high resolution, the variables *REZ%* and *TF%* are used. The resolution is determined with the *XBIOS(4)* command, and *REZ%* is set based on the return, 1 for medium resolution and 2 for high resolution. *REZ%* is then used to modify the Y position of items on the screen. The Y coordinate has a maximum value of 200 in medium resolution and 400 in high resolution; for high resolution every Y coordinate is multiplied by 2.

TF% is used to adjust the size of the text displayed on the screen. The DEFTEXT command sets the text size; a size of 6 in medium resolution is equivalent to 13 in high resolution. So for high resolution, *TF%* is set to 7 and added to the size of the text in medium resolution, 6 + 7 = 13.

We hope that you'll use *Calorie Counter* in good health. ■



***Ron** and **Kathy** are both practicing internal medicine in Los Angeles. Together they run Schaefer SuperGraphics, which has a line of medically oriented software for the Atari ST.*

▶ PROGRAM LISTINGS BEGIN ON PAGE 30

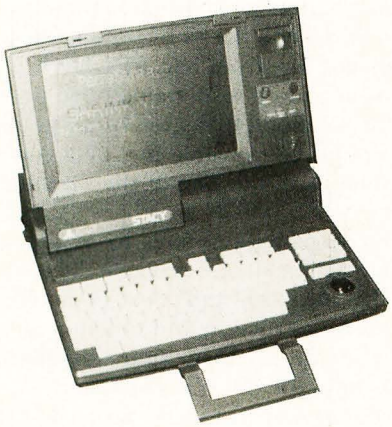
ST GOSSIP FROM Hollywood, USA

by TG

A date with Stacy

While attending the Atari World show in Anaheim, California, in late April, I met a most charming young lady whom a great many attendees of the show did not get to see. Many expected her to be there when the show opened, but, alas, she was nowhere to be found. However, the evening after the first day of the show, in a private suite in the Disneyland Hotel, this shy young lady made an appearance. She was very attractive; small, a bit on the short side, wider than she was tall, with ebony skin as black as Darth Vader's heart. She was beautiful!

I am referring, of course, to Stacy, Atari's laptop ST computer. The one and only prototype was brought in so that the select few in that suite could get a sneak



peek at this latest entry in the ST family. And, although Stacy did appear at the show near the end of the second day, users were allowed only to look, not touch. Fortunately, no such restrictions were in place in the hotel suite the previous night.

The Stacy prototype is not immediately recognizable as an ST, the most obvious difference being the radical departure from the Atari-gray color scheme. Stacy is sleek and jet black, and features a hinged top that contains the screen and covers the keyboard when the machine is closed. The screen on this prototype was an amber liquid-crystal display, but we were told that the final machines would feature blue- on-white backlit supertwist LCD displays (which are nice). The LCD displays the 640×400 monochrome resolution nicely. Understandably, but unfortunately, this LCD screen cannot display color.

To the right of the screen were volume and contrast controls, and just above those was a small "clipboard" clip that allows you to hang notes next to your screen. The keyboard was of standard ST gray-white color, but the keys actually had a better feel than those of the Mega ST. The main keyboard is full size, but the function, cursor pad and numeric pad keys are all miniaturized. The keyboard was angled perfectly for laptop typing although the angle feels a little wrong when you have it on a table in front of you.

Since the computer is designed to be used on the go, when even a desk or table is rare, using a mouse might prove inconvenient. The route Atari chose was to provide a small trakball (with two buttons) to the right of the keyboard and just below the numeric keypad.

The prototype featured a battery pack (said to allow the machine to run for about four hours on a charge), adapter (in the event you have access to an electrical outlet) and two double-sided 3.5-inch floppy-disk drives. It was stated that the final models will be offered in two configurations, the first with a single floppy drive, the second with one floppy drive and an internal mini 20-megabyte hard disk. Both models will come with one megabyte of RAM and TOS 1.4 on ROM (the prototype was running TOS 1.2).

Except for the miniaturized accessory keys on the keyboard, Stacy could be used just like a 1040ST. The prototype featured all of the ports found on a standard ST: MIDI IN and OUT, printer, modem, DMA, monitor, cartridge. It also featured both mouse and joystick ports so users won't be forced to use a trakball if they want to use a mouse (a switch is used to select between them). Removable panels covered many of the ports, although I overheard someone say all ports would have such panels in the release models. My biggest surprise came when I popped off the largest of these panels, on the left side of the machine, and found the 64-pin processor bus connector, usually only found inside the Mega STs.

And how did she run? Very nicely, as far as I could tell. There wasn't much software on hand at the suite, so I couldn't put Stacy through any hard paces. At first I found the screen updating on the LCD a little too slow and difficult to look at, but I got used to it in a hurry (the final screen will be better). Everything seems to run at normal speed.

The one thing that bothered me was talk of a name change. If they're going to do it, it will have happened by the time you read this. I'll be disappointed if they change the name to "ST Laptop" or "Portable ST," because for once Atari has a computer with a name—and it's a friendly one.

As for price and availability, the floppy-disk-only version of Stacy is planned to retail for $1,495. I did not get a price for the hard-disk-equipped model, though I have heard it will be in the $2,500 range. The first Stacy machines are expected to ship in August, but knowing Atari's policy on marketing new products in the US. . .don't hold your breath.

The mysterious TT

For years rumors have buzzed within the Atari community that the successor to the current ST line was something referred to as the TT. ST stands for Sixteen/Thirty-Two, referring to the 16/32-bit Motorola 68000 microprocessor; so what does TT stand for? Thirty-Two, as in 32-bit microprocessor, and now specifically for the Motorola 68030. The 68030 is, of course, a more powerful and much faster chip than the 68000, used in the current STs, so the TT will consequently be a more powerful piece of hardware.

Beyond the name, what do we know about the TT? Some rumors claimed to be fact have been circulating around the Atari community. I've gathered the most reliable of these, but remember, Atari has changed its mind more often than I can count, so even if these rumors are true today, they may be wrong tomorrow.

The TT will be offered in two models—the standard TT and the TTx. Both will feature a 68030 running at 16 megahertz, two megabytes RAM, and will run under TOS 1.4 and offer all the standard ST ports and connectors. It will also have internal card slots for expandability. The expansion slots will be 32-bit VME bus connectors (as used in Sun and other workstations), allowing non-Atari cards to be utilized. VME has been a standard expansion bus for many years, so many boards exist that would potentially be plugged into the TT "as is." The TT may also feature SCSI and ASCI hard-disk interface ports.

The TTx differs in that it may have more expansion slots, an internal hard disk (perhaps 40 to 80 megabytes), an Ethernet networking port and will run AT&T's Unix System 5.3 with X-Windows, in addition to TOS 1.4. Unix is provided to make the TTx more attractive to businesspeople as a potential workstation, and the Ethernet capabilities will make it easier to integrate TTx computers into existing networks than is possible with the ST series.

Sound capability is unknown, though graphics seem a little more concrete. All the usual ST graphics modes are supported in addition to "boosted" low- and medium-resolution modes. The boosted low-resolution screen offers 256 colors at once (320 × 200 pixel mode), and the medium-resolution screen allows 16 colors at once (either 640 × 400 or perhaps a 640 × 480 screen). A new super-high-monochrome mode has been added as well, resulting in a 1280 × 960-pixel display (which will require a special monitor). In addition to this, the TT rectifies a deficiency in the ST, giving each of the primary colors in RGB sixteen levels instead of 8 (as on the ST), bringing the total palette up from the ST's mere 512 colors to a more respectable 4,096 possible hues. Beyond that, even better graphics would be possible by plugging a graphics card into one of the machine's slots.

ST compatibility seems to be a goal, though a difficult one to obtain due to the fact that the 68000 and 68030 do not handle every instruction and function absolutely identically (such as the exception stack and the MOVE command). I've heard that Atari is putting a special new system variable into the TOS that will be in the TT, which will allow programmers to check which processor the machine is running so the software will be able to compensate for the differences. Expect a lot of updates to current ST software when the TT ships!

And the price? Expect the TT to run about $2,499, and the TTx to be closer to $5,000. When? Atari says soon, but I'll be surprised if the TT machines materialize before Christmas. ■

After a long and relaxing stay at the Institute for the Potentially Nervous, **TG** *has decided to give up his favorite vice—police chief's daughters—and his favorite sport, van dodging. Because he's found that fresh air stimulates his creativity, clears his complexion and prevents nosebleeds, he now writes this column while hang-gliding over the Pacific Ocean.*

VideoGames & 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_____

SIGNATURE_____

NAME_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

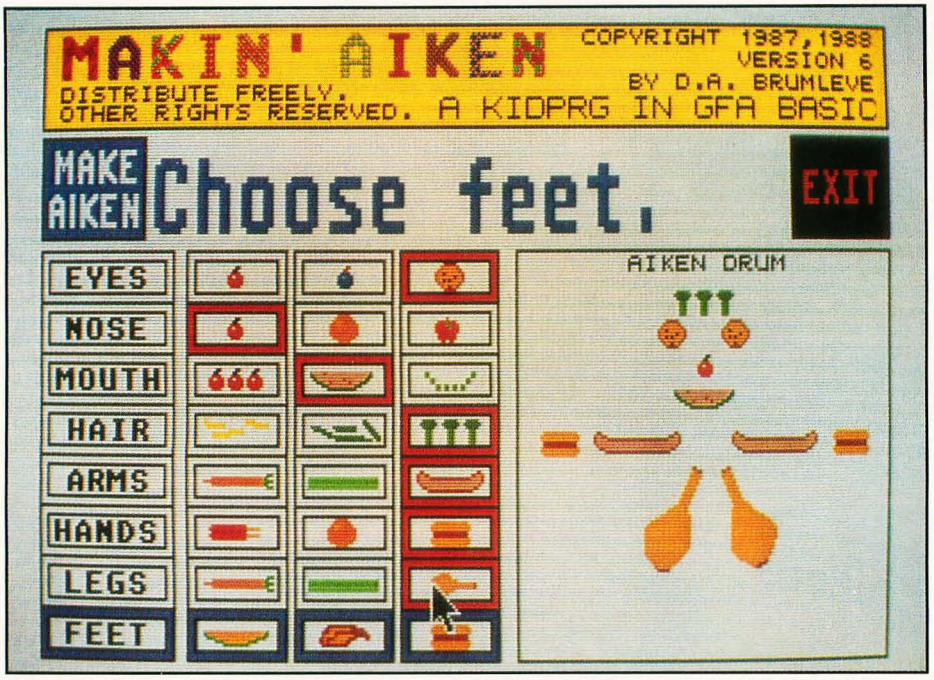
Money back on unused subscriptions if not satisfied!
Foreign—add $10 for postage.

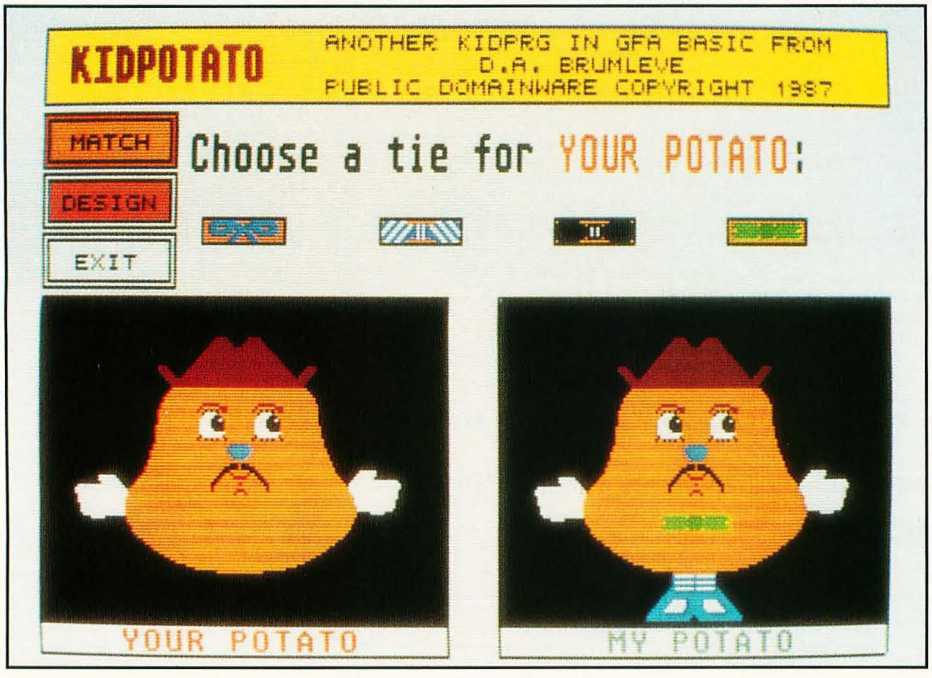
Your first issue will arrive in 6 to 8 weeks.
WATCH FOR IT!!
Offer expires Oct. 25, 1989

CJUWA

PD PARADE

BY GEORGE L. SMYTH





Remember when the major criticism leveled against the Atari ST series of computers stemmed around the apparent lack of software? The range of programs currently available to ST users is now quite exceptional. Programs covering everything from spreadsheets to adventure games, databases to utilities, are numerous in both the public domain and commercial areas. One of the few weak areas, however, concerns software designed for preschool children. Fortunately, this need was recognized and is in the process of being filled by D.A. Brumleve, who has written a wide range of educational games and fun exercises for children. This month I offer two of my five-year-old's favorites from her catalog.

TOP:

Offered to the four- to ten-year-old range, *Makin' Aiken* is an interactive song that allows the child to build a silly picture of Aiken, the man in the moon, from a choice of fruits and vegetables.

BOTTOM:

When in my preschool years, Mr. Potato Head was a game all of my friends owned. Now *Kidpotato*, a similar game, is available on the ST, and you don't even need to crawl under the couch in search of that lost ear.

Note: *The programs described in this article are available on this month's disk version, as well as on BBSs and commercial online services throughout the country.*

Makin' Aiken

Offered to the four- to ten-year-old range, *Makin' Aiken* is an interactive song that allows the child to build a silly picture of Aiken, the man in the moon, from a choice of fruits and vegetables.

The program is based on the children's song, "There was a man lived in the moon." The object of the program is to choose food from a picture row of edibles that will represent the eight body parts of Aiken: his eyes, nose, mouth, hair, arms, hands, legs and feet.

As each choice is made, the selection is displayed on the right side of the screen as the chorus of the song is played. For instance, if the picture of cherries is selected for Aiken's hair, the cherries are drawn in the box labeled AIKEN DRUM where his hair should be. As the song is played, the words are printed in large type so that the child can sing along ("His hair was made of cherries, cherries, cherries. His hair was made of cherries, and his name was Aiken Drum").

Selections are registered by clicking the mouse on the appropriate choice. My five-year-old had no trouble playing with this program, although I did need to do some coaching to get him to sing the song. Fun for the little ones, *Makin' Aiken* should be kept on your games disk.

Kidpotato

When in my preschool years, Mr. Potato Head was a game all of my friends owned. Several years ago the game was reintroduced and once again gained a degree of popularity. Now *Kidpotato,* a similar game, is available on the ST, and you don't even need to crawl under the couch in search of that lost ear.

The *Kidpotato* screen is divided into two major sections: the left portion of the screen labeled YOUR POTATO and the right portion labeled MY POTATO. A choice of MATCH or DESIGN determines the direction of the program, whether the object will be to match an existing potato or design a new one.

If the MATCH option is chosen, a potato in the MY POTATO section is drawn with a randomly chosen hat, eyes, nose, mouth, tie and shoes. The object of this game is for the child to match each of these objects. One at a time, a selection of four types is offered. For instance, the first object to be matched is the potato's hat. Four hats are placed within buttons at the top of the screen. The mouse is then used to determine which hat was chosen on the MY POTATO side.

A correct choice displays WOW on the screen and happy musical tones are played while drawing that part of the potato on the MY POTATO side. An incorrect guess honks at the user and prompts him to TRY AGAIN.

After the correct hat has been chosen, the selection process continues with the eyes, the nose and so on. When all of the choices have been completed the potato waves his hand.

The other option, DESIGN, allows the child to design his/her own potato. This section of the program is similar to the above, with the exception that the child designs the potato by choosing the parts from a picture menu. One attribute at a time, the choices of hat, eyes, nose, etc., are made until the potato is complete.

◆

Remember when the major criticism leveled against the Atari ST series of computers stemmed around the apparent lack of software?

◆

This program is excellent for the two- to six-year-old who can tolerate only a minimum of frustration—a key concern for this age group. The play allows simple and fun interaction with the computer for even the youngest child.

I hope your children enjoy the two programs on this month's disk as much as mine. Additional programs from D.A. Brumleve can be found on DELPHI, GEnie, CompuServe, your local bulletin board and each month in ST-LOG as part of the *Compukid Connection* column.

***George L. Smyth** has 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.*

Menu Management with Pascal

by William B. Busby

Even though Personal Pascal (PP) generally provides easy access to the features of GEM, some assistance is needed beyond the manual to begin writing applications that run under GEM. This article will discuss one of those GEM topics: the construction and management of menu bars within PP.

The largest job we're faced with when building our menu is planning. Several considerations must be observed. The total width of the menu titles must not exceed the width of the resolution in which our application will run. If our application is to run in all resolutions, we must limit ourselves to 40 characters, including the "DESK" title provided by GEM, to fit the width of the low-resolution screen. This width restriction also applies to the menu items found under the titles, or our menu items will wrap around the screen when the menu is activated.

Applications designed for medium and high resolution can use the full 80-column width of the screen. It is usually recommended that a space be placed on each side of a menu title. When planning the items that will be found under each title, be careful to group them logically. For example, if you have a title called "FILE," place all disk functions under that title, including the option to exit the application. It would probably be wise to review several commercial applications which you find easy to use and observe how the menus are constructed. We are limited to 24 items per title and/or less than 25% of the total screen area. GEM maintains a buffer of up to 25% of the screen area, which is used to redraw the screen when the drop-down menu is removed from the screen and when alert boxes disappear.

To build a menu with PP, the following steps must be taken:

1) Use *New__Menu* to reserve menu space and obtain a pointer to the menu tree.

2) Add all titles to the menu with *Add__MTitle*.

3) Add the items found under the titles with *Add__MItem* in the same order the titles were added. Add all items to the first title before adding any items to the second, etc. Unexpected results may occur if this caution is not observed. If any of the items are to be preceded by a check mark, all items should be preceded by two spaces. All items under a title should be the same length. Shorter items should be padded with blanks to make their lengths equal to the longest found under any one title.

After our menu structure is defined in memory we probably need to do some touch-up work before we actually show it to the user. Seldom are all options in an application available to the user at the same time. It is standard practice, and common sense, to use the menu to show the options currently active with check marks and to disable any options not available at any given time.

The procedure *Menu__Check* is used to show or remove check marks preceding items. The procedure *Menu__Disable* will make each disabled item show up in half-intensity, and mouse clicks on these disabled items won't cause menu events to be placed in the event queue. When all the preparation has been done, we're ready to show the results of our efforts to the user with a call to *Draw__Menu*.

With the menu on the screen, the user could take several actions. He could press a key, choose a menu item, move the mouse to another portion of the screen or manipulate any displayed windows or icons. GEM provides a central procedure to monitor these events. In PP, this call is implemented in the function *Get__Event*, which returns an integer defining the event(s) and any data associated with the event in the parameter definitions. GEM applications are designed to monitor this event call and take actions based on the events detected. Through a call to *Get__Event*, we can detect keyboard events, mouse-button events, mouse-pointer events, timer events and messages. Messages can be further broken down into menu events and window events.

That's a lot of stuff for anyone to keep track of. Fortunately, we can tell GEM to notify us of only certain events, making our application more selective and simplifying our task. In the sample program, we requested only message events by passing the event mask parameter as *E__Message*. We then went a little farther and acted only on the *MN__Selected* message (menu item selected). When the function returns, we will find the title and item indexes in message-buffer elements 3 and 4, respectively.

From there, managing our menu is simply a matter of passing these indexes to our *Do__Menu* procedure and testing them. The first thing we want to determine is if the *Desk__Title* has been selected. If so, we need to show the information about our application. Typically, we would display a dialog box describing our application. In the sample program, I just slipped an alert box in here.

If the *Desk__Title* wasn't selected, we need to test the item numbers to determine which was selected. Since each menu item is assigned a unique integer number, we don't need to concern ourselves with the title, but the title number is there if we choose to use it.

Depending on the item chosen, we will probably want to make changes to the status of our menu. Fortunately, PP has provided some comprehensive procedures to keep the menu properly updated. The usage of each of these is documented in our application and for the most part covered well in the manual. The following procedures are available.

Menu__Check. This procedure will place or remove a check mark, which precedes the item text. You should place two spaces in front of the text to make room for this check mark or the first character of the item text will be overwritten. No error will result from removing a check mark that doesn't exist or from checking an item that is already checked.

Menu__Enable. This procedure will display the menu item in full-intensity and allow menu-selected events to occur when the item is chosen. It is not an error to enable items already enabled.

Menu__Disable. This procedure will display the menu item in half-intensity and prevent any menu-selected event from occurring when it is chosen. It is not an error to disable items already disabled.

Menu__Text. This procedure is used to change the text of a menu item. The text overwrites any text previously in the menu item and thus must be the same length. Part of planning a menu is to determine the maximum length string that may be placed in a menu item and ensuring enough space is reserved for it

when the item is initially defined. A caution is required here: *Menu_Text* may not work on your system, depending on the version of GEMSUBS.PAS you have. Some versions had an error in the *Menu_Text* declaration within GEMSUBS.PAS. However, it is easy to fix. Examine your GEMSUBS.PAS file. If the last parameter of *Menu_Text*, the parameter for the new text, is declared a VAR parameter, remove the word "VAR." Leave all else alone. *Menu_Text* should then work as advertised in the manual. My thanks goes to the ICD team for assisting in tracking down this error. It has been corrected in the latest version of PP, version 2.02.

Menu_Normal. After the action requested by the user has been performed, it will be necessary for us to return the menu title to normal video. When a menu is activated, GEM draws the title in reverse video. Since GEM has no means of determining when we are done performing the menu action, it leaves it up to the application to return the title to its original appearance.

Using these routines, we should have no trouble keeping our menus current and easing the task of the user when our application is used. After our application has performed its task and the user tells us he wants to get out, we must call *Erase_Menu* to remove the menu from the screen before exiting to the desktop. Actually, *Erase_Menu* could be used in conjunction with *Draw_Menu* to display several menus within an application. The spreadsheet *Masterplan* uses this technique effectively. Choosing one option brings up another menu with additional options, and so forth.

A final procedure available is *Delete_Menu*, which is used to release the memory consumed by our menu tree back to the operating system. This call is not normally needed since exiting the application releases all our memory back to the operating system. If you find a need for it, you must ensure the menu has been erased from the screen before calling *Delete_Menu* or some strange menus will be displayed. Try it sometime for a laugh, but be prepared to reset your computer.

I've included a short program, MENUS.PRG, which demonstrates the techniques presented in this article. It shows both the correct way to construct menus and some of the common errors made. The source code, MENUS.PAS, is liberally commented and should be easy to follow. ■

William Busby *is currently serving in the Air Force at Arnold AFB, TN. His spare time is spent studying for a degree in electronics engineering and playing with his one-year-old daughter, Brittany.*

MENUS WITH PASCAL
Listing 1:
PASCAL

```
program menus;

{$I d:\gemsubs}

var
  menu           : Menu_Ptr;          { pointer to menu bar tree          }
  quit,                               { flag used to exit program         }
  grid_status    : boolean;           { flag to show or disable grid      }
  unusual_text,                       { menu item text string             }
  new_text       : str255;            { global var for menu text change   }
  file_title,                         { integers for the menu titles and  }
  opt_title,                          { items                             }
  screwed_title,
  open_item,
  close_item,
  dummy_line,
  grid_mode,
  drw_mode,
  erase_mode,
  too_short,
  too_long,
  nocheck_space,
  non_standard,
  about_menus    : integer;

procedure build_menu;

{ Call New_Menu to reserve space for our menu tree, add all the titles to
  our tree and then add the items to each title in the order the titles
  were declared. The dummy line will be disabled, items that will lead to
  dialog boxes (such as open) are suffixed with '...' to indicate some-
  thing follows this selection. The text for grid status may be changed
  to Don't Show Grid and we have to be sure to make the string long
  enough to have room for the max text we may put in there. }

  begin
    menu := New_Menu( 15, '  About Menus...   ' );
    file_title     := Add_MTitle( menu, ' File ' );
    opt_title      := Add_MTitle( menu, ' Options ');
    screwed_title := Add_MTitle( menu, ' Screwed Up ' );
    open_item      := Add_MItem( menu, file_title, '  Open...    ' );
    dummy_line     := Add_MItem( menu, file_title, '-------------' );
    close_item     := Add_MItem( menu, file_title, '  Close      ' );
    grid_mode      := Add_MItem( menu, opt_title, '  Show Grid          ' );
    drw_mode       := Add_MItem( menu, opt_title, '  Draw Mode          ' );
    erase_mode     := Add_MItem( menu, opt_title, '  Erase Mode         ' );

    { Here we show some of the mistakes that can be made when designing
      menus. Run the program to see how these look on the screen.       }

    too_short      := Add_MItem( menu, screwed_title, '  Too Short' );
    too_long       := Add_MItem( menu, screwed_title, '  This text is too long
 to fit the screen width and wraps around. ' );
    nocheck_space := Add_MItem( menu, screwed_title, 'No Space for Check' );

    { This isn't a mistake, but it shows how you can use some of the non-
      standard characters within your menu.                              }

    unusual_text  := Concat( '  ', chr(14),chr(15),' ',chr(11),' ',chr(3),
                             ' ', chr(175),' ',chr(189),' ',chr(174) );
    non_standard  := Add_MItem( menu, screwed_title, unusual_text );
```

PROGRAM LISTINGS

```
  end;

procedure set_up_menu;

{ Proper planning dictates we should disable all options that make no
  sense or aren't available to the user to prevent confusion. }

  begin
    grid_status := true;
    Menu_Check( menu, drw_mode, true );
    Menu_Disable( menu, dummy_line );
    Menu_Disable( menu, close_item );
    Menu_Disable( menu, grid_mode );
    Menu_Disable( menu, drw_mode );
    Menu_Disable( menu, erase_mode );
    Menu_Check( menu, nocheck_space, true );
  end;

procedure do_close;

{ All we do is set the quit flag, a real program would probably have to do
  some cleaning before it could exit properly. You could erase the menu
  here but I left it in the main block to show program progression. }

  begin
    quit := true;
  end;

procedure do_open;

{ Here we would open the file by displaying the file selector box and load
  the chosen file, it is important to reconfigure the menu to enable any
  options now available to the user, also by disabling the open item in the
  menu we prevent loading another file on top of the one in memory. }

  var
    button : integer;

  begin
    Menu_Disable( menu, open_item );
    Menu_Enable( menu, close_item );
    Menu_Enable( menu, grid_mode );
    Menu_Enable( menu, drw_mode );
    Menu_Enable( menu, erase_mode );
    button := Do_Alert( '[1][Display the file selector here.][ Ok ]', 1 );
  end;

procedure do_grid;

{ Just a change to the menu text to change the commands available to the
  user. We don't show a grid anymore than we do anything else, but it's a
  command you may want to use one day. The grid_status flag is used to
  keep track of the current state of the grid. }

  begin
    if grid_status = true then                { Grid was previously on, but }
      begin                                   { user must want to turn it   }
        grid_status := false;                 { off since he called us here.}
        new_text := '  Don''t Show Grid  ';
      end
    else
      begin                                   { Grid was off, but user must }
        grid_status := true;                  { want to see it again.       }
        new_text := '  Show Grid        ';
      end;
    Menu_Text( menu, grid_mode, new_text );
  end;

procedure do_draw;

{ Show the current mode by checking the menu option chosen. }

  begin
    Menu_Check( menu, erase_mode, false );
    Menu_Check( menu, drw_mode, true );
  end;

procedure do_erase;

{ Show the current mode my checking the menu option chosen. }

  begin
    Menu_Check( menu, drw_mode, false );
    Menu_Check( menu, erase_mode, true );
  end;

procedure show_about_box;

  var
    button : integer;

  begin
    button := Do_Alert( '[1][A dialog box goes here!][ Ok ]', 1 );
  end;

procedure do_menu( title, item : integer );

{ First we check to see if Desk_Title has been chosen, if so display the
  'about menus' box. Since each item has a unique number we can just
  concentrate on checking the item number after that. After the item has
  been processed, we must return the menu title to normal video. }

  begin
    if title = Desk_Title then show_about_box
      else
```

```
    if item = close_item then do_close
      else
    if item = open_item then do_open
      else
    if item = grid_mode then do_grid
      else
    if item = drw_mode then do_draw
      else
    if item = erase_mode then do_erase;
    Menu_Normal( menu, title );
  end;

procedure event_loop;

{ Most of the parameters in the event call aren't needed to monitor menu
  bar events. We pass dummy values to fill the spaces. Since we only told
  GEM to notify us of messages, we don't need to test the event value for
  other events or messages, we just assume (probably not a good idea) GEM
  will pass only messages. We use the CASE statement to operate only on
  menu messages and ignore all others to keep it simple. In our case we
  need only the values in the 3rd and 4th subscripts of the message
  buffer. }

  var
    event,
    dummy  : integer;
    msg    : Message_Buffer;

  begin
    quit := false;
    repeat
      event := Get_Event( E_Message, 0, 0, 0, 0,
                          false, 0, 0, 0, 0,
                          false, 0, 0, 0, 0,
                          msg,
                          dummy, dummy, dummy,
                          dummy, dummy, dummy );
      case msg[ 0 ] of
        MN_Selected     : do_menu( msg[ 3 ],     { title index }
                                   msg[ 4 ] );   { item index  }
      end; { case }
    until quit = true;
  end;

begin                                 {    Main Program Block   }
  if Init_Gem >= 0 then               { Pretty self-explanatory }
    begin
      Set_Mouse( M_Point_Hand );
      build_menu;
      set_up_menu;
      Draw_Menu( menu );
      event_loop;
      Erase_Menu( menu );
      Exit_Gem;
    end;
end.
```

END

PROGRAM LISTINGS

BY ARTHUR LEYENBERGER

I just returned from the Spring COMDEX in Chicago. I won't provide a detailed show report because that task is being ably handled elsewhere in this issue by Frank Cohen. But I do want to share a few thoughts on what is considered to be the pre-eminent computer show.

First, the initials COMDEX refer to COMputer Dealers EXposition. The show is held twice a year, in Atlanta in the spring and in Las Vegas in the fall. This year, the Atlanta convention hall was being remodeled so the show was temporarily moved to the McCormick Center in Chicago, where the Consumer Electronics Show (CES) is held every June.

The contrasts couldn't be greater. Whereas CES draws about 100,000 people, COMDEX attendance was estimated to be roughly 60,000 (by the promoters) but it seemed like less. In addition, CES is usually an exciting event,with large and small manufacturers rubbing elbows with large and small distributors,who rub shoulders with large and small retailers, etc.

The rubbing takes place both at the show and at parties, press conferences and the like. You would think with all of that rubbing going on, the static electricity would blow out half the electronics products being exhibited. But it doesn't.

Electricity was the element that was lacking at COMDEX. It has become a place where three-piece-suit manufacturer types show off me-too type products to an uninspired audience. There were a couple of hot products, but, overall, aisles and aisles of PC clones have a tendency to put me to sleep.

The most excitement, aside from a couple of new Atari products, came from the fiasco of having a show within a show. You see, MACdex (get it, Macintosh Dealers Exposition) was supposed to be a hot item within the show, but a dozen or so booths does not a show make. Even Apple stayed away, apparently having better things to do with their time and money. There were more MAC products being demonstrated throughout the main show than at the MACDex exhibit.

The MACdex non-event only goes to show that the Interface Group (COMDEX's organizers) is thinking greed. They charged an extra $10 for the privilege of visiting MACdex, but the extra fee was dropped after the first day due to lack of interest. A dull COMDEX just proves that the industry is maturing, I guess, which is a good sign.

Atari is back

Speaking of good signs, Atari had a rather large booth located at the front of north hall. Ironically, IBM (heh-heh) had a (dare I even call it a booth) palace about five times the size right behind Atari. As usual, Atari fell under the shadow of the corporate standard-bearer.

Anyway, Atari had two products under glass (actually Plexiglas). One was Stacy, the ST laptop, and the other was a hand-held PC called Portfolio. According to Atari, both of these products will start shipping (in the U.S.) in June.

Stacy, which has apparently now been renamed the Transportable, was seen last November at the Fall COMDEX. However, it appeared then in two incarnations: a foam mockup of the final design and an extra-large working prototype with circuit boards, handmade interface cables and a separate LCD screen.

The working display model, a pre-production unit, was handsome with a sort of European-flavor design. Specs showed that it weighs 15.2 pounds, has a 640-by 400-pixel supertwist LCD screen (probably with backlighting), all of the ports of a regular ST or Mega ST, one megabyte of memory and a single 3 ½-inch floppy-disk drive (double sided). Battery powered, the Transportable can also accept another disk drive or hard disk.

Atari was kind enough to let me "play" with it for a couple of minutes. The keyboard was surprisingly good, similar to that on a Mega ST. Also, the built-in trakball seemed odd at first but I'm sure I could get used to it. There is a mouse port so you could use a normal ST mouse if you were unable to adapt to the trakball. All in all, the $1,500 Atari Transportable was quite impressive. Let's hope Atari can bring it out this summer and it is successful.

The other major hit both at the Atari booth and at COMDEX in general was the hand-held MS-DOS computer. Called the Portfolio, this under-$400 computer contains DOS 2.11 in ROM, 128K of RAM (expandable to 640K), an 8-line by 40-character display and a 63-key QWERTY keyboard. Since the Portfolio uses an 80C88 processor, it is really an IBM clone.

The unit is about the size of a videotape and weighs under a pound. Two standard AA batteries power the Portfolio for up to 48 hours of continuous use. It can

also accept either ROM cards for software or RAM cards for data storage. In addition, an interface jack is provided for exchanging data directly with a PC.

A text-editing program and an address/phone list ROM card are provided with the Portfolio. Not only does this product look good, but it appears to be functional as well. I was able to use the Portfolio for about three minutes (it was closely guarded during the entire show) and came away delighted at its design and overall feel. Although I doubt if I could type an entire article using the diminutive keyboard, it had a good feel and with practice could be used for short notes or memos.

Nineteen-eighty-nine could well be the year of Atari in the United States. After years of neglect, the American market will see more availability of STs, which means Atari has a reason to start advertising again. Advertising means more sales and that means more ST software. Although some major software players have left the Atari market for greener pastures, there is a good number of excellent ST programs currently available. More machines in the U.S. market can only help that situation.

More importantly, Atari's new Transportable and Portfolio signify Atari's potential at leading the industry with innovative products. The laptop is inexpensive (as laptops go) and should appeal to many people, including the burgeoning musician market. The Portfolio has the promise of appealing to everybody, especially the huge installed base of PC users.

For the last year or so, Atari has claimed that 1989 will see their re-emergence in the American market. Products like these will definitely help and make it exciting once again to be an ST user. I congratulate Atari on their return to these shores and wish them the best of luck. I know we all do.

New improved!

The Hewlett-Packard Deskjet Printer has been available for just over a year and is a remarkable product. It offers the power of a laser printer at the price of a high-end dot-matrix printer. It uses inkjet technology that Hewlett-Packard pioneered a few years ago with their ThinkJet printer.

The Deskjet is a much more improved product than its predecessor, which required specially coated paper, was slow and had output that wasn't as good as the best of the then currently available dot-matrix printers. Using liquid ink much like a printing press, the Deskjet sprays the ink onto the paper through a grid of tiny holes in a horizontally moving printer head while the paper moves past it.

Unlike conventional dot-matrix impact printers, only one pass of the print head is necessary per line. Darker or bold characters are created by spraying more ink onto the page. Light characters and fine lines are made with less ink.

The Deskjet can support up to 300 dpi (dots per inch) text and graphics output just like a good laser printer. In fact, the output is virtually indistinguishable from laser-printed output. With text, the printer will print either in draft or letter-quality modes at a speed of 120 and 240 characters per second, respectively. Graphics output is, of course, much slower.

Although the Deskjet has received more than its share of major accolades, there are still a few areas for improvement. First, the replaceable ink cartridges don't seem to last as long as HP claims they will. HP's claim that "your mileage may differ depending upon how you drive" is true; printing a lot of graphics and bold text will deplete the ink supply more quickly. Still, even normal text consumes more ink than you would normally expect.

Another problem is the ink itself. Because the ink requires several seconds to dry, the paper-handling mechanism does not drop one printed page on top of another. Instead, as each new page is printed, the previous page is lowered onto the stack of output and the new one is held separately.

A more serious problem involves chemistry. Once the ink has dried on the page, moisture will smear it. I get nervous using the Deskjet to print mailing labels for this very reason. If the Postal Service is true to their word about delivering mail in inclement weather (Neither rain, nor snow. . .), letters addressed with the Deskjet may never arrive.

Like everything else, you get what you pay for. The Deskjet print quality is laser-class, its speed is not. This is due to the differences in printing technology between the two printers. Since the Deskjet prints in real-time, each and every page requires the same amount of time and is limited to some extent to the speed that the computer can process the output. A laser printer buffers the output in its own memory so that multiple copies can be printed as fast as the mechanical mechanism will allow.

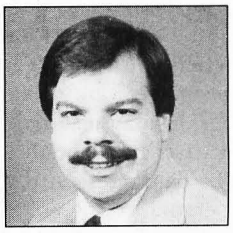
Aside from the differences between a laser and Deskjet printers, the Deskjet speed is respectable but not blazingly fast for this price of printer. The problem stems from the mechanical mechanism of the printer. A lot activity has to happen before the print head even starts to print a page.

To understand this better, you need to know that rated printer speeds are measured simply by how fast a line of text can be printed. Throughput is rarely mentioned by printer manufacturers because it must take into account paper handling, mechanical movement of the print head and the like. As a result the Deskjet is a much slower printer than the specifications indicate.

Hewlett-Packard has addressed some but not all of these problems with the introduction of the Deskjet Plus. The new version of the printer has all of the features of the Deskjet with the addition of a few more. Printing speed is still rated the same (120 and 240 characters per second for laser and draft quality printing) but throughput is said to be two to five times faster. This increase is due to a faster microprocessor, paper pick-up mechanism and motor, which moves the paper through the printer in half the time of the original.

You also get more fonts for your money. The Deskjet Plus now contains six portrait and four landscape fonts built-in. Landscape printing is now possible without the optional font cartridge. Larger fonts are also included, the largest being 30 points. The Deskjet Plus is now also capable of printing on legal size paper.

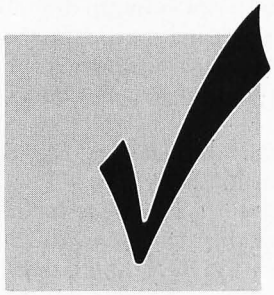
The Deskjet Plus has a list price of $995. The original Deskjet has been reduced in price to $795. Hewlett-Packard also reduced the prices by 30% on font cartridges and memory modules (for downloaded fonts) for the printer. Both the original Deskjet and the new Deskjet Plus should be given careful consideration if you are in the market for a new printer. ■



Arthur Leyenberger *is a computer analyst and freelance writer living in beautiful New Jersey. He can be reached on CompuServe at 71266,46 or on DELPHI as ARTL.*

ST-CHECK

A CHECKSUM PROGRAM FOR THE ST



BY CLAYTON WALNUM

ALL RESOLUTIONS

Typing in a BASIC program listing can be a frustrating and time-consuming task. Just one mistyped character will frequently render a program completely unusable. So to ensure that your program will run correctly, the entire listing must be checked character by character against the original. This can take many hours. To make matters worse, you can't trust your own eyes. Do you know how easy it is to overlook an O where a 0 is supposed to be?

Typing checkers like *ST-Check* take over the arduous task of proofreading your program files. Using this program can cut down your debugging time by a huge factor. When the checker's output matches that published with the listing, you can be sure your typing is accurate.

Introspection

When you run ST-Check against itself, you will get one of several results. The program may just give up and crash. In that case, go through the listing character by character until you find your typing error.

A second possibility is that the program will run okay, but will create all bad checksum data. This may indicate an error somewhere between Lines 80 and 420. Find the typo and correct it.

The last possibility is that the checksum data will have only a few bad values. In this case, use the normal method detailed below to locate your errors.

Warning: Until you get your checksum data for ST-Check to match the data following the listing, you can't trust it to proofread other programs.

Using ST-Check

When you finish typing a ST BASIC program listing from the magazine, save a copy to your disk, and then run ST-Check. The program will first ask for a filename. Type in the name for the program you wish checked (the one you just saved to the disk), and press RETURN. You'll then be asked for a "bug" name. Enter a filename for the checksum file (this can be any name not already on the disk), followed by RETURN.

ST-Check will now proofread the program. When the checking process is complete, you'll have a file on your disk (saved under your bug name) which contains the checksum data for the program checked.

Check the last value of each line. If it matches the value in the published checksum data, go on to the next. If it doesn't match, you've got a typo.

To find the error, look at the line number of the data statement in which the bad value occurred. This number is equivalent to the first program line the data evaluates. Let's call this "Line X." Count the entries in the data line until you get to the bad value. We'll call this count "Y." Now look at the program you typed in. Starting with and including Line X, count down Y lines. The line you end up on will be the one containing the typo.

Correct the error, and then rerun ST-Check. When you get all the checksum data to match that published in the magazine, your new program is ready to run.

Passing the buck

Okay, friends. Here's where the truth comes to the fore. I can take only minimal credit for ST-Check, as it's virtually a direct translation from D:CHECK2 (ANALOG #16) by Istvan Mohos and Tom Hudson. All accolades and tribute should be directed to those two fine gentlemen. I'm sure they'll divvy it up fairly, and perhaps pass a small share onto me. Thanks, guys!

You may now type in this month's ST BASIC program, secure in the knowledge that the searching eye of ST-Check is primed and ready.

ST-Check Listing 1 — ST Basic

```
10 'ST CHECK typing validator by Clayt
on Walnum
20 'based on a program by Istvan Mohos
 and Tom Hudson
30 if peek(systab)=1 then cl=17 else c
l=32
40 fullw 2:clearw 2:gotoxy cl,0:? "ST
CHECK":ex=0:sp=0:x=0
50 input "Enter filename: ",f$:input "
Enter BUG name: ",f1$
60 on error goto 590:open "O",#1,f1$:o
pen "I",#2,f$:close #2
70 open "I",#2,f$:on x goto 140,220
80 color 2:?:? "Counting lines":lineco
unt=0:color 1
90 on error goto 570
100 line input#2,i$:linecount=linecoun
t+1
110 ? ".";:goto 100
120 close #2:q=int(linecount/10):dim c
(linecount),r(q)
130 x=1:goto 70
140 range=0:lyne=0:color 2:?:?:? "Fill
ing array":color 1
150 ? ".";:count=0
160 line input#2,i$:count=count+1
170 lyne=val(i$):r(range)=lyne:range=r
ange+1
180 on error goto 580
190 line input#2,i$:count=count+1:if c
ount=10 then 150
200 goto 190
210 close #2:x=2:goto 70
220 color 2:?:?:? "Calculating checksu
ms":color 1
240 for i=1 to linecount:checksum=0:li
ne input #2,i$:l=len(i$)
245 if mid$(i$,l,1)=" " then l=l-1:got
o 245
250 for z=1 to l:number=asc(mid$(i$,z,
1))
260 if number=asc(" ") and ex=0 and sp
=1 then goto 320
270 if number<>asc(" ") then sp=0 else
 sp=1
280 if number<>34 then 300
290 if ex=1 then ex=0 else ex=1
300 if ex=0 and number>=asc("a") and n
umber<=asc("z") then number=number-32
310 product=x*number:checksum=checksum
+product:x=x+1:if x=4 then x=1
320 next z:? ".";
330 checksum=checksum-1000*int(checksu
m/1000):c(i)=checksum:x=2:next i
340 close #2:lyne=r(0):item=0
350 color 2:?:?:? "Creating BUG file":
color 1
360 count=10:total=0:if linecount<10 t
hen count=linecount
370 i$=str$(lyne):i$=i$+" data "
380 for i=1 to count:datum=c(10*item+i
)
390 i$=i$+str$(datum):i$=i$+",":total=
total+datum:next i
400 i$=i$+str$(total):print #1,i$:? ".
";
410 item=item+1:linecount=linecount-10
:if linecount<1 then 430
420 lyne=r(item):goto 360
430 close #1:clearw 2:?:gotoxy 0,1
440 ? "To check BUG data against the c
hecksum data found in the magazine,"
450 ? "return to the GEM desktop and d
ouble click your BUG file.  You may"
460 ? "then SHOW the data on your scre
en or PRINT the data to your printer."
:?
470 ? "The line number of each data st
atement coincides with the first line"
480 ? "of the user program the data st
atement evaluates.  Numbers within"
490 ? "each data statement represent c
onsecutive lines of the user program."
500 ? "The last number is the total.":
?
510 ? "Check the last number of each s
tatement against the version in the"
520 ? "magazine.  Only when there's a
discrepancy need you check each number
"
530 ? "in the data statement.":?
540 ? "Take note of the lines containi
ng typos, then make corrections.  When
"
550 ? "all corrections have been made,
 rerun this program to double check."
560 ? "Press <RETURN>":input i$:close
#1:close #2:end
570 if err=62 then resume 120
580 if err=62 then resume 210
590 if err=53 then ? chr$(7);"FILE NOT
 FOUND!":close:resume 50
600 ? "ERROR #";err;" at LINE ";erl:en
d
```

ST-Check Checksums

```
 10 data  447, 129, 203, 518, 661, 160
, 942, 482, 640, 556, 4738
 110 data  25, 905, 797, 52, 79, 349,
852, 644, 9, 402, 4114
 210 data  883, 479, 834, 822, 42, 498
, 255, 165, 826, 410, 5214
 310 data  337, 1, 166, 578, 136, 801,
 898, 937, 271, 769, 4894
 410 data  363, 99, 155, 889, 243, 764
, 168, 192, 906, 156, 3935
 510 data  757, 251, 146, 509, 146, 91
6, 539, 541, 733, 845, 5383
```

ST CHECK

END

IAN'S QUEST

BY
IAN
CHADWICK

Whew. Talk about opening up a can of worms! You know the old saw that "everything will be fine as long as you don't mention sex, religion or politics"? Well, include computer languages in that list too.

I've received letters and phone calls about the virtues of almost every computer language ever invented. People want to run me out of town, metaphorically at least, because I took a few lighthearted swings against their particular sacred cow. A comment I made about GFA BASIC was questioned by a reader and answered in the letters column—not by me, but by someone else in the magazine who wanted to run roughshod over my ideas. I *didn't* write that comment that went "What Ian really meant was. . ."

Heck, what I said was what I really meant. I don't need any translators. I am quite capable of planting my own foot in my mouth, thank you.

What's this all about? I expect the language police to come knocking at my door one night: "Open up! We know you're using BASIC in there. Come out, the house is surrounded!"

At least that's how it will be when some of these people get their way, judging by the tone of their comments.

Several people have a penchant for C that borders on obsession. C is currently the number one hit on the computer-language hit parade and may be on its way to becoming the first computer religion. The C crowd told me off in several letters and conversations, warning me to steer clear of their favorites. These guys were the worst, sort of language hit-men. I think it's because C is a rigid, structured, no-nonsense language. It broaches no bungling, half-hearted attempts to learn it, no lallygagging, dawdling, tinkering. C demands harsh discipline and dedicated effort to master it. Nietzsche would have loved C.

Me, I don't care much for C, partly because it generates frighteningly large computer books on how to use it. Any language that requires you to buy a 1,000-page tome from a third party as your reference guide (and not even a tutorial!) is suspect. Who's going to read all that? It has no plot, no dialogue, no pictures of dinosaurs in it.

C doesn't let me ramble. It's a force of linearity, absolute obedience, a Triumph of the Will diskwise. Besides, C users have no sense of humor. They discuss their language in reverential tones, brook no snide comments or witty asides. Me, I still have a "4th Love If Honk Then" sticker on my car. Not that I really like Forth all that much, but I love seeing the reactions it generates when people in the car behind me try to figure it out!

And maybe I'm a tad prejudiced because C lurks behind Unix, the most difficult, obstinate, obtuse, obscure operating system ever devised, and 15-plus years ago at that! I figure anything *that* closely tied into Unix can't be good. If, by some massive stroke of ill fortune, Unix comes to the Atari, I'll probably pack up my computers and take up full-time zymurgy instead.

I guess which language is best depends on 1) who you are and 2) what you want to do with the language.

I like BASIC because I can write doodle-bug code. What's wrong with that? I tinker a lot with it, horse around. It suits me; it's like the way I cook, by guess and by golly. If you came over to my house, you'd understand. I have a workroom cluttered with books, software, computers, rats, guitars—and I bounce around inside it from one thing to another. My mind wanders, so I like a language that reflects that. Of course, I can't think of any single major application ever written in BASIC, but that's fine by me.

Nowadays, however, BASICs are getting away from the dab-and-dollop approach by mutating into structured BASICs. Nothing very new, really; I had S-BASIC on my old Kaypro 4. They're sort of like C hiding in BASIC clothing. GFA BASIC is like that, and that's the one thing that I *don't* like about the program. It's like the authors are trying to gently lead us over to C. Not me; I'll go back to that old fossil, ST-BASIC, before I give up my GOTOs and GOSUBs.

HiSoft BASIC is a bit more flexible. It allows you to roam all over the place as you write, but it jumps back to C's embrace when it comes to handling GEM, BIOS and the other important calls. It disappointed me because I wanted a smoother transition from Microsoft QuickBASIC on the PC, but I wanted the power and chrome of GFA. Maybe one day they'll get married and give birth to SoftGFA or something like that.

What do you want to do with a language, anyway? It's not as if programming is exciting. Excitement is being a fighter pilot, fireman, policeman or mud wrestler. Programmers are generally a dull lot who only talk about computers, can't spell properly and write opaque manuals for their products that no one else understands [I resent that!—Ed.], even when they can untangle the bad grammar.

When you're a programmer, you don't get together over a brew with the boys to watch the Big Game and brag about "generating one mean subroutine" during the week. It just isn't a back-slapping, shoulder-punching lifestyle.

Erroll, a friend, just bought an ST and came to ask me what language he should learn to program. I asked him "Why bother?" Of course, no one accepts that answer, and he waited for my (yawn) usual routine:

If you *really* want to tear into the machine and get control right down to the individual bits, then assembly language is the tool to choose. Only problem is, it takes 10,000 lines of code to do anything even vaguely important in AL. The serious AL programmers I've known are troglodytes anyway, people with bizarre eating habits (pizza and Coke at 7 a.m.) and worse social habits, mostly caused by their insistence in leading nocturnal lives. [Sheesh! Ian, if you thought you got a lot of mail before. . .—Ed.]

AL is relatively easy to learn and write, given that each line is really only one single command, and there aren't *that* many commands. But it's nearly impossible to decode it later without copious comments. It's also difficult to debug, because some of the simple commands are downright arcane. Besides, if you can't do hexadecimal math in your head, then you'll fail as an AL programmer.

Pascal and its somewhat richer cousin, Modula 2, are "sorta" languages. They're sorta like structured BASIC, sorta like C, sorta a bit of this, that and the other thing. I always disliked Pascal because I figured it was invented by a bunch of guys sitting in a hot tub somewhere in California. Nowadays, I'm better informed and know it was created by some guy in Switzerland. Probably in a hot tub.

To my delight, I discovered Alice, a rare Pascal interpreter, which gives the language a much-needed shot in the arm in my eyes. Sadly, it lacks its own compiler. However, if you're teaching Pascal or want to experiment with it, then this is the version to play with. A nice implementation too. Very classy act.

Still, I can't get away from the deep-seated fear that Pascal programmers all drive BMWs, say "smashing" a lot and wear tennis whites to putter around the house. Probably drink gin and t's too.

Anyway, I still think that GFA BASIC may be the best language around for anyone who isn't willing to stay up nights to program in AL, isn't up to the drill-sergeant correctness of C, or isn't one of the upwardly mobile Pascal hot-tub crowd. Look: BASIC is easy, fun and rewarding. Most people can't say that about much else in their lives.

Thoroughly discouraged, Erroll decided he ought to learn to use the applications first, before dipping into the murky swamp of programming on his own.

I shouldn't knock hot tubs, though. I spent the whole last week of April assembling one in my backyard, as well as helping building the deck around it. Lots of hammering, sawing, tapping tender digits with 20 oz. framing hammers. One day, when I finally get it working and the power upgraded, I'll probably enjoy it. Right now, my hydro service is so overloaded that when I turn on my computer, the street lights dim. So the tub is sitting there, full of pure Canadian ice water. Nice. Maybe I should drop a few trout fingerlings into it.

Did you know that a 600-gallon tub weighs 6,000 pounds? So what, you ask? Well, I know a lot about hot tubs now, not to mention decks. During the week, I made about 800 trips to the lumber and hardware stores, trying to get enough wood, screws, nails, blades—whatever. I always bought one too few of everything, it seemed. I had the deck all planned out and printed, but I really needed a good estimating package, one that I could use to tally up everything, make one trip, and be done with it. I also needed to figure costs in advance to know for how much to sink into debt.

Sure, this could easily be done on a spreadsheet, but I want something more than just a one-shot deal (I may go into the deck-building business). A general-purpose template would basically require writing an entire hardware store and lumber yard inventory into rows and columns. Rather large, cumbersome and inelegant.

I need a better solution, something that prints out an estimate, not just rows and columns. A multi-window spreadsheet with cut and paste would work, but would still be clumsy.

What I *need* is a super-smart CAD program that can do the estimates based on the drawing. I'd be able to select an area and define it as 2x6 pressure-treated, 3/16-inch spacing and the program would figure out the area, then how much lumber it translates into, along with approximate number of nails (by the pound). Somewhere, maybe in another window, it would keep track of all the running totals: joists, trim, lattice, that sort of thing. Then all I'd have to do is throw in a few prices for the wood and a guess at the number of hours it would take, and *voila!*, an estimate.

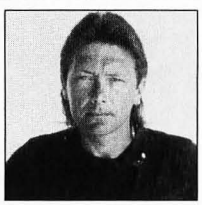
That's a ways off, maybe not before Halley's comet comes around again. Maybe. In the meantime, I'm looking for something functional in a single application, something to write up shopping lists, figure costs and mark-ups and prepare estimates. I'm doing other renovations this year and need to do this often. Anyone have any suggestions? I haven't seen anything like it yet.

More importantly, such a program would be a godsend for the laptop ST (if, and when, it arrives). I know several contractors who want desperately to be able to do on-the-site estimates. This combination could push them into Atari's hands, saleswise.

While you're at it, I need something else too: an animal maintenance program. That's right, I want to track the daily food intake of my menagerie. I have two dogs, six cats and 13 rats (Natasha had babies). I also need to record weights, height, length, vet visits, illness, pedigrees, etc. Really. Everything, both the general stuff and the daily. Maybe graphs too; you know, weight changes, size, health quotas.

I know, I know. Why don't I just write these applications myself? Well, I would, if I had the time. And I may have to anyway. But I thought I'd ask first.

Last column, I mentioned a new game: *Tetris*, from Spectrum Holobyte. Since the game is slated for a full scale review, I won't go into the details here. But to make a long, long story short, it's *great*. One of the most addictive games I've ever played. Very hot stuff. Simple but demanding. And best of all, absolutely non-violent. Maybe the Russians will become the master programmers of the non-violent game. Wouldn't that roast a few socks in the White House? Never mind, just go buy one and see what I mean. I've reached level eight, with a score around 8,400, so far. A real tough haul to get further. That's all for now. Keep the letters coming, but please, no more bombs.



***Ian Chadwick** is a Toronto-based freelance writer with massive blisters and sore muscles from a recent encounter with construction. He lives beyond his means in a small house with his wife, Susan, and too many animals and spends his time trying to keep the neighbors' kids out of his yard.*

CALORIE COUNTER
Listing 1:
GFA BASIC
2.0

```
'IMPORTANT NOTE: THE "~" CHARACTER AT THE END OF SOME OF THE LINES
'  IN THIS LISTING INDICATES THAT THE PROGRAM LINE WRAPS AROUND TO THE
'  NEXT LINE OF THE LISTING.  THE TWO LINES SHOULD BE TYPED AS ONE,
'  WITHOUT THE "~".
'
' ##############################################################################
' ##############################################################################
' ##########################  CALORIE COUNTER  #################################
' ######################  BY RON & KATHY SCHAEFER MDs  #########################
' ###################################  Copyright 1989  #########################
' ###################################    by ST-Log     #########################
' ##############################################################################
' ##############################################################################
If Xbios(4)=0 Then
  Alert 3,"SORRY WORKS IN HIGH & MEDIUM|RESOLUTION ONLY",1,"BYE",Dummy
  Edit
Endif
Rez%=Xbios(4)              ! Check resolution 0=low 1=med 2=high
If Xbios(4)=2 Then
  Tf%=7                    ! Correction faction factor for text size
Endif
Dim Spalette%(16,3)
@Save_pal                  ! Save current pallet
@Setcolors                 ! Set program screen colors
@Introscreen               ! Do title screen
Do
  At$="Calorie Counter|By Ron & Kathy Schaefer M.D.s|(C) 1988 Schaefer~
 SuperGraphics|"
  At$=At$+"Published by ST Log"
  Alert 0,At$,1,"Count|Help|Quit",Dummy
  Exit If Dummy=3
  If Dummy=1 Then
    @Calcount
    Cls
  Endif
  If Dummy=2 Then
    @Help
  Endif
Loop
@Restorepal            ! Restore original palet at end of prgram
Edit
' ------------------------ HELP SUBROUTINE -------------------------------
Procedure Help
  Titlew 1," /|\ HELP /|\ "
  Deftext 1,0,0,6+Tf%
  Openw 1              ! if using 3.0 use this instead OPENW 1,0,19
  Fullw 1
  Clearw 1
  Print At(1,2);
  Print "  To use the Calorie Counter just load in the expandable data base"
  Print "  of food items called CALORIES.DAT, this is done automatically if"
  Print "  the file is in the same directory as the program CAL_CNT.PRG"
  Print
  Print "  Now enter in the number of calories that you want to plan your"
  Print "  meal or day for.  This will serve as a Goal diet."
  Print
  Print "  Once the food items have been loaded just click on the items"
  Print "  that you want added up.  Click with the left button to add,"
  Print "  and with the right button to subtract an item.  As you plan"
  Print "  your menu, try and approximate the Goal or Ideal diet."
  Print
  Print "  You can print out a list of the selected foods to take with"
  Print "  you to store by clicking on LIST at the bottom of the screen."
  Print
  Print "          Ideal American Diet:      % of total calories"
  Print "          ============================================"
  Print "                   Protein            12%"
  Print "                   Fats               30%"
  Print "                   Carbohydrates      58%"
  Print "                                            hit return to continue";
  Void=Inp(2)
  Closew 1
Return
' ---------------------- MAIN CALORIE COUNT SUBROUTINE ----------------
Procedure Calcount
  K=0
  If Not Openfile!         ! If the file has not been opend do so
    Path$=Dir$(0)
    Filename$=Path$+"\CALORIE.DAT"
    If Not Exist(Filename$) Then
      Fileselect "*.DAT","CALORIE.DAT",Filename$
    Endif
  Endif
  If Filename$<>"" Then
    Titlew 1," Calorie Counter and Menu Planner "
```

PROGRAM LISTINGS

CALORIE COUNTER

```
Openw 1              ! if using 3.0 use this instead OPENW 1,0,19
Fullw 1
Graphmode 1
Clearw 1
Deftext 1,0
If Not Openfile! Then
  Print
  Print
  Print "                    Opening the file ";Filename$
  Print "                    Reading in calorie data on item:";
  Color 1
  Box 139,13*Rez%,467,35*Rez%
  Box 136,11*Rez%,470,37*Rez%
  Open "I",#1,Filename$
  Openfile!=True
  Input #1,T%        ! Read in number of food items and DIM arrays
  Dim N$(T%),Cals(T%),Fats(T%),Carbos(T%),Prots(T%),Quant(T%)
  Do
    Inc Nt%
    Print At(54,4);Nt%
    Input #1,N$(Nt%),Cals(Nt%),Prots(Nt%),Fats(Nt%),Carbos(Nt%)
    Exit If Eof(#1)
  Loop
  Close #1
Else
  For N%=1 To T%
    Quant(N%)=0
  Next N%
  Caltotal=0
  Prottotal=0
  Carbototal=0
  Fattotal=0
Endif
Print At(21,10);"Enter the number of calories to be"
Print At(21,11);"your goal: ";
Color 1
Box 154,68*Rez%,445,92*Rez%
Box 151,66*Rez%,448,94*Rez%
Input "",Gcaltotal
Clearw 1
Defmouse 6
Deftext 2
@Initmenuplaner
' ******* main loop *********
Do
  If Mx>475 And My>157*Rez% And K=1 Then
    @Do_sound_1(5,4)
  Endif
  Exit If Mx>475 And My>157*Rez% And K=1
  Showm
  Mouse Mx,My,K
  If K>0 Then
    If My>35*Rez% And My<153*Rez% Then
      @Do_sound_2(9,7)
    Endif
    If My>159*Rez% Then
      @Do_sound_1(3,4)
    Endif
    If My>35*Rez% And My<153*Rez% Then    ! Find which item mouse is over
      If Rez%=1 Then
        L%=Int((My-35)/8)+1
      Else
        L%=Int(((My-35)/8)/Rez%)-1
      Endif
      If K=1 And L%+F%<=T% Then
        Inc Quant(L%+F%)                   ! Add food item
        Add Caltotal,Cals(L%+F%)
        Add Fattotal,Fats(L%+F%)
        Add Prottotal,Prots(L%+F%)
        Add Carbototal,Carbos(L%+F%)
      Endif
      If K=2 And L%+F%<=T% Then
        Dec Quant(L%+F%)                   ! Subtract food item
        If Quant(L%+F%)<0 Then
          Quant(L%+F%)=0
        Else
          Sub Caltotal,Cals(L%+F%)
          Sub Fattotal,Fats(L%+F%)
          Sub Prottotal,Prots(L%+F%)
          Sub Carbototal,Carbos(L%+F%)
        Endif
      Endif
      If L%+F%<=T% Then
        If Quant(L%+F%)=0 Then
          Deftext 1,0
```

CALORIE COUNTER

```
      Else
        Deftext 2,1   ! If the quantity is >0 highlight that item
      Endif
      Print At(2,4+L%);N$(L%+F%);"  "
      Print At(30,4+L%);Cals(L%+F%);"  "
      Print At(40,4+L%);Fats(L%+F%);"  "
      Print At(50,4+L%);Carbos(L%+F%);"  "
      Print At(60,4+L%);Prots(L%+F%);"  "
      Print At(70,4+L%);Quant(L%+F%);"  "
    Endif
    Deftext 3,0
    Print At(2,3);"TOTAL";
    Print At(30,3);Caltotal;"  "
    Print At(40,3);Int(Fattotal);"  "
    Print At(50,3);Int(Carbototal);"  "
    Print At(60,3);Int(Prottotal);"  "
    Print At(66,3);"Quantity"
    Deftext 1
  Endif
  If Mx<154 And My>157*Rez% Then
    Add F%,15
    If F%>T% Then
      Sub F%,15
    Endif
    For N%=1 To 15
      If (N%+F%)<=T% Then
        If Quant(N%+F%)=0 Then
          Deftext 1,0
        Else
          Deftext 2,1   ! If the quantity is >0 highlight that item
        Endif
        Print At(2,4+N%);N$(N%+F%);"  "
        Print At(30,4+N%);Cals(N%+F%);"  "
        Print At(40,4+N%);Fats(N%+F%);"  "
        Print At(50,4+N%);Carbos(N%+F%);"  "
        Print At(60,4+N%);Prots(N%+F%);"  "
        Print At(70,4+N%);Quant(N%+F%);"  "
      Else
        Print Space$(72)
      Endif
    Next N%
    Deftext ,0
  Endif
  If Mx>154 And Mx<321 And My>157*Rez% Then
    Add F%,-15
    If F%<0 Then
      F%=0
    Endif
    For N%=1 To 15
      If Quant(N%+F%)>0 Then
        Deftext 2,1   ! If the quantity is >0 highlight that item
      Else
        Deftext 1,0
      Endif
      Print At(2,4+N%);N$(N%+F%);"  "
      Print At(30,4+N%);Cals(N%+F%);"  "
      Print At(40,4+N%);Fats(N%+F%);"  "
      Print At(50,4+N%);Carbos(N%+F%);"  "
      Print At(60,4+N%);Prots(N%+F%);"  "
      Print At(70,4+N%);Quant(N%+F%);"  "
    Next N%
    Deftext ,0
  Endif
  If Mx>321 And Mx<475 And My>157*Rez% Then          ! LIST routine
    Clearw 1
    At$="Where do you want the menu|list to be printed?"
    Alert 2,At$,1," Screen | Printer ",Pr
    @Do_sound_2(4,4)
    Defmouse 6
    If Pr=1 Then
      Deftext 1
      Tx$="  Menu Listing           Calories   Fat      Carbo    "
      Print At(2,2);Tx$+"Protein   Quantity"
      Print At(1,3);String$(72,"=")
      Ln%=0
      For N%=1 To T%
        If Quant(N%)>0 Then
          Print At(2,4+Ln%);N$(N%);"  "
          Print At(30,4+Ln%);Cals(N%);"  "
          Print At(40,4+Ln%);Fats(N%);"  "
          Print At(50,4+Ln%);Carbos(N%);"  "
          Print At(60,4+Ln%);Prots(N%);"  "
          Print At(70,4+Ln%);Quant(N%);"  "
          ' Print At(2,3+Ln%);N$(N%),Cals(N%),Quant(N%)
          Inc Ln%
```


CALORIE COUNTER

```
                If Ln%>15 Then
                  Ln%=0
                  Print At(55,21);"Click to continue."
                  Do
                    K=Mousek
                    Exit If K>0
                  Loop
                  Clearw 1
                  Tx$="  Menu Listing           Calories   Fat      Carbo    "
                  Print At(2,2);Tx$;"Protein   Quantity"
                  Print At(1,3);String$(72,"=")
                Endif
              Endif
            Next N%
            Print String$(72,"=")
            Deftext 3
            Print At(2,5+Ln%);"TOTAL";
            Print At(30,5+Ln%);Caltotal;"  "
            Print At(40,5+Ln%);Int(Fattotal);"  "
            Print At(50,5+Ln%);Int(Carbototal);"  "
            Print At(60,5+Ln%);Int(Prottotal);"  "
            Deftext 2
            Print At(55,21);"Click to continue."
            Deftext 1
            Do
              K=Mousek
              Exit If K>0
            Loop
            Clearw 1
          Else        ! Print out list of items on the printer
            Sd=10
            Tx$="  Menu Listing           Calories   Grams     Grams     "
            Lprint Tx$;"Grams    Quantity"
            Lprint Space$(35);"Fat       Carbo     Protein"
            Lprint String$(72,"=")
            For N%=1 To T%
              If Quant(N%)>0 Then
                Lprint N$(N%);Space$(27-Len(N$(N%)));
                Lprint Cals(N%);Space$(Sd-Len(Str$(Cals(N%))));
                Lprint Fats(N%);Space$(Sd-Len(Str$(Fats(N%))));
                Lprint Carbos(N%);Space$(Sd-Len(Str$(Carbos(N%))));
                Lprint Prots(N%);Space$(Sd-Len(Str$(Prots(N%))));
                Lprint Quant(N%)
              Endif
            Next N%
            Lprint String$(72,"=")
            Lprint "  TOTAL";Space$(20);
            Lprint Caltotal;Space$(Sd-Len(Str$(Caltotal)));
            Lprint Fattotal;Space$(Sd-Len(Str$(Fattotal)));
            Lprint Carbototal;Space$(Sd-Len(Str$(Carbototal)));
            Lprint Prottotal;Space$(Sd-Len(Str$(Prottotal)))
          Endif
          Ln%=0
          F%=0
          @Initmenuplaner
        Endif
      Endif
    Loop
    Clearw 1
    Closew 1
    Deftext 1
  Endif
Return
' ################### set up and draw first screen for menu planner ##########
Procedure Initmenuplaner
  Deftext 2
  Print "                                Calories   Fat      Carbo    Protein"
  Deftext 3
  Print At(2,2);"GOAL";
  Print At(30,2);Gcaltotal
  Print At(40,2);Int(Gcaltotal*0.3/9)
  Print At(50,2);Int(Gcaltotal*0.58/4)
  Print At(60,2);Int(Gcaltotal*0.12/4)
  Print At(2,3);"TOTAL";
  Print At(30,3);Caltotal
  Print At(40,3);Fattotal
  Print At(50,3);Carbototal
  Print At(60,3);Prottotal
  Print At(66,3);"Quantity"
  Deftext 1
  Print String$(72,"=")
  Print At(1,20);String$(72,"=")
  Deftext 3
  Print At(7,21);"NEXT PAGE"
  Print At(27,21);"LAST PAGE              LIST              QUIT"
```

CALORIE COUNTER

```
  For N%=1 To 15
    If Quant(N%+F%)>0 Then
      Deftext 2,1
    Else
      Deftext 1,0
    Endif
    Print At(2,4+N%);N$(N%+F%);"  "
    Print At(30,4+N%);Cals(N%+F%);"  "
    Print At(40,4+N%);Fats(N%+F%);"  "
    Print At(50,4+N%);Carbos(N%+F%);"  "
    Print At(60,4+N%);Prots(N%+F%);"  "
    Print At(70,4+N%);Quant(N%+F%);"  "
  Next N%
  Color 0
Return
' ---------------------- CLICKING SOUND SUBROUINTES ------------
Procedure Do_sound_1(Snd,Snd1)
  Sound 1,12,Snd,Snd1
  Wave 1,1,9,6000
Return
Procedure Do_sound_2(Snd,Snd1)
  Sound 1,12,Snd,Snd1
  Wave 1,1,8,512,5
  Wave 0,0
Return
Procedure Do_sound_3(Snd,Snd1,Per,Dur)
  Sound 1,2,Snd,Snd1
  Wave 1,1,9,Per,Dur
Return
' ---------------------------- SET SCREEN COLORS -------------------
Procedure Setcolors
  Setcolor 2,0,7,7
  Setcolor 0,0,0,0
  Setcolor 3,7,7,7
  Setcolor 1,7,0,2
Return
' ------------- SAVE ORIGINAL COLOR PALETTE ----------------------
Procedure Save_pal
  For Z%=0 To 15
    Dpoke Contrl,26
    Dpoke Contrl+2,0
    Dpoke Contrl+6,2
    Dpoke Intin,Z%
    Dpoke Intin+2,0
    Vdisys
    Spalette%(Z%,0)=Dpeek(Intout+2)
    Spalette%(Z%,1)=Dpeek(Intout+4)
    Spalette%(Z%,2)=Dpeek(Intout+6)
  Next Z%
Return
Procedure Restorepal
  ' --------------------- RESTORES PALLET -------------------
  For Z%=0 To 15
    Dpoke Contrl,14
    Dpoke Contrl+2,0
    Dpoke Contrl+6,4
    Dpoke Intin,Z%
    Dpoke Intin+2,Spalette%(Z%,0)
    Dpoke Intin+4,Spalette%(Z%,1)
    Dpoke Intin+6,Spalette%(Z%,2)
    Vdisys
  Next Z%
Return
' ------------------------ DO INTRO TITLE SCREEN ---------------------
Procedure Introscreen
  For Zz=1 To 12
    Deftext 3,0,0,Zz
    @Do_sound_1(1,Zz/2)
    Text 160,30*Rez%,"Calorie Counter"
    Pause 3
  Next Zz
  For Zz=1 To 12
    Deftext 2,0,0,Zz
    @Do_sound_1(1,Zz/2)
    Text 80,48*Rez%,"by Ron & Kathy Schaefer M.D.s"
    Pause 3
  Next Zz
  For Zz=1 To 12
    Deftext 1,0,0,Zz
    @Do_sound_1(1,Zz/2)
    Text 125,66*Rez%,"Brought to You by ST Log"
    Pause 3
  Next Zz
  Deftext 1,0,0,6+Tf%
  Pause 10
  Return
```

END

PROGRAM · LISTINGS

CALORIE COUNTER

12 Issues $28

$19 OFF THE COVER PRICE

12 Issues with Disk $79

NEW LOWER PRICE

BREAK AWAY FROM THE PACK

The world of ATARI-ST continues to grow by leaps and bounds, and ST-LOG is there every step of the way! We stand apart from the competition by offering more color, comprehensive reviews and in-depth features. SUBSCRIBE NOW!

☐ **12 Issues $28**
MCTWW

☐ **12 Issues with Disk $79**
DCTWW

ST LOG

MAKE CHECK PAYABLE TO L.F.P., INC., P.O. Box 16928, N. Hollywood, CA 91615.

☐ PAYMENT ENCLOSED ☐ BILL ME
CHARGE MY: ☐ VISA ☐ MASTERCARD

CARD # EXP SIGNATURE

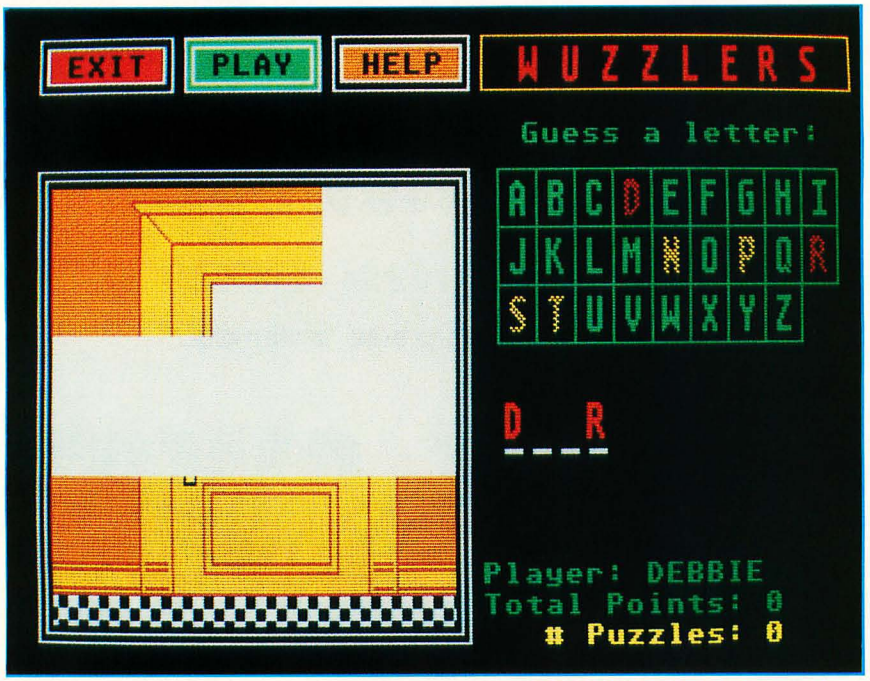
NAME

ADDRESS

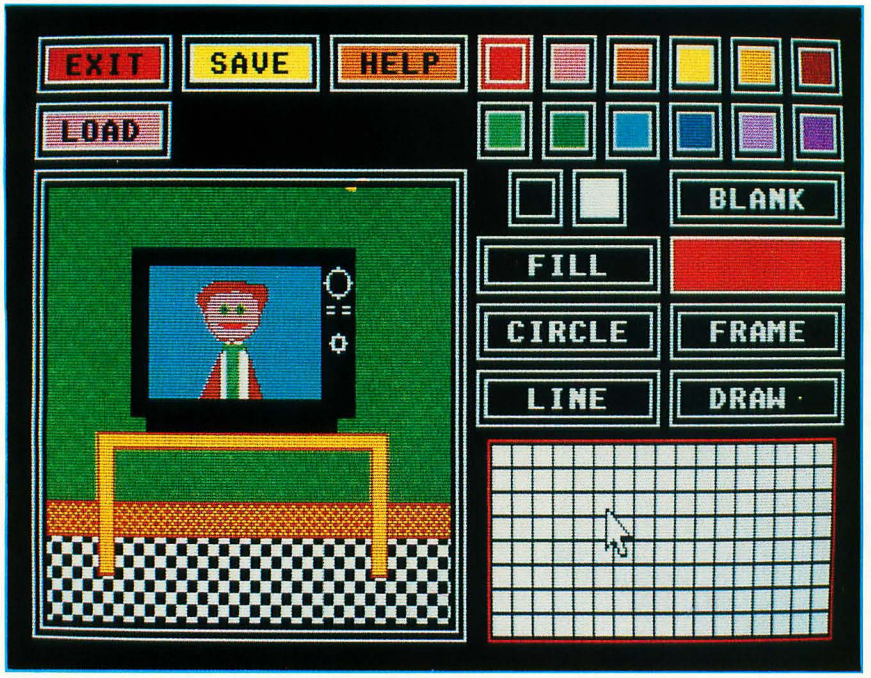
CITY STATE ZIP

The Compukid

BY D.A. BRUMLEVE



WUZZLERS



My eight-year-old son and I learned to program at the same time. I would say we learned to do it together, but in reality it was more like side by side. I went my own way, writing mouse-driven children's programs, and he went his, writing games of all kinds. One of his first efforts was Hangman, the age-old word-guessing game. He displayed dashes corresponding to the letters in the word and a hangman's gallows. Guesses were made primarily on the basis of trial and error, with a little help from word length and letter position as correct answers were given. If the player guessed a letter wrong, body parts of a stick were progressively added; when the stick figure was complete, the player lost the round. His game mimicked precisely the version I had played on paper as a child, and I was impressed. I wanted to do my own version of Hangman.

Wuzzlers is a version of Hangman, but with a twist: A piece of a picture puzzle is displayed for each word. If the user guesses a letter that is not in the word, another piece of the puzzle is displayed. The developing picture reminds the player that his guesses are running out and provides a hint as to what the mystery word could be. Thus, the "hangman"—in this case, the puzzle—is not simply a foreteller of doom but rather an aid that increases the likelihood of success.

I originally designed the program as a practice aid for beginning readers, but it could also be used for vocabulary building, spelling lists, rebus puzzles, foreign noun acquisition or just plain fun. The program consists of a game screen and a file-creation screen, both of which are accessible from a shell screen. As programming for children is my specialty, I wanted to make the program simple enough for the youngest potential Hangman players. Some concessions to these young users were required, and I will discuss them in detail later. But first, let's take a look at the program.

Connection: WUZZLERS

LOW RESOLUTION

Disk preparation

The minimum hardware requirements for use of the program are a 520ST with a single-sided drive and a color monitor. Follow the ST-LOG disk instructions to decompress WUZZLERS.ARC, and transfer the files WUZZLERS.PRG and WUZZLERS.DAT to the root directory of a freshly formatted disk (that is, do not put them in a folder). An additional file in the ARCed file, WUZZLERS.LST, contains the GFA BASIC source code for the program. Only WUZZLERS.DAT and WUZZLERS.PRG are needed on your play disk. Make sure your screen is in low resolution. Insert the disk in Drive A and double-click on WUZZLERS.PRG. A shell screen will appear, offering the choices Play the Game, Make a Puzzle and EXIT.

Playing the game

Click the mouse on "Play the Game." The game screen is shown with the following mouse-controlled boxes: EXIT, PLAY, HELP and the alphabet boxes. When the game screen first appears, a green alert box is on the left-hand side of the screen. Choose the number of players and type the players' names as requested. Instructions will be displayed near the top of the screen. Each player works on a separate puzzle and initiates his turn by clicking the PLAY box when directed to do so.

When a puzzle is loaded, a broken line is drawn under the alphabet boxes to indicate how many letters are in the puzzle words and one piece of the puzzle is displayed in the large box on the left-hand side of the screen. Use the mouse to guess the letters in the puzzle word. A perfect score of nine points is possible on a puzzle. Each time you guess a letter that is not in the puzzle word, one point is deducted from the possible nine and another piece of the puzzle is displayed.

When the last piece of the puzzle has been placed in the box, you may keep guessing, as long as each letter you guess is in the word. If one more guess is wrong, your turn is forfeited (with a score of 0 for that puzzle) and the mystery word is revealed.

Statistics on scores are revised at the end of each turn and displayed in the bottom right-hand corner of the screen. "# Puzzles" (printed in yellow text) indicates the number of puzzles a player has successfully solved; a running total of the number of points a player has accumulated is displayed in green text.

This game is open-ended: You can end the game whenever you please or continue as long as you like. If you play long enough, a purple alert box will eventually announce that all the puzzles on the disk have been used. If you want to continue at that point, you'll be seeing the same puzzle files again. You won't have another disk of puzzles available this first time, but in the future, when you see this message, it can serve as a cue to put another puzzle disk in the same drive from which the program was loaded. Click on EXIT, and you'll be looking at the shell screen again.

Creating/adding a puzzle file

Click "Make a Puzzle." The file-creation screen will appear. This subprogram allows you to develop and alter puzzle files for use with the game. Options on this screen include EXIT, SAVE, HELP, LOAD and a variety of drawing options. Use the mouse to select colors, drawing options, fill patterns and magnify mode. A white "easel" is on the left half of the screen. Drawing and filling are achieved by simply pressing either mouse button while the mouse is on the easel.

Two different size "nibs" are available for drawing; click the DRAW option once for a tiny dot; twice for a larger one. You can slowly "drag" the mouse on the easel, holding either button down, to draw a continuous line. The fill pattern box (to the right of the FILL option) displays the currently selected fill pattern (the default pattern is solid). Click repeatedly on the box to scroll through the many patterns offered. To fill a shape on the easel, make sure FILL and the desired fill pattern are selected, place the mouse within the shape and click either button.

Other drawing options (LINE, FRAME and CIRCLE) require you to press the left button once to begin the operation and again to set the final shape/line in place. For example, to draw a line, click the left mouse button on the easel where you want the line to start, let go of the button, move the mouse to wherever you want the line to end and click the left button again. *DEGAS* users are accustomed to "dragging" the mouse with the left button down to perform these operations. My choice of two clicks of the left button is one of my concessions to young users, who sometimes have difficulty dragging the mouse.

Once you have begun a drawing operation, you can end it without drawing the shape/line in either of two ways. One is simply to click the right mouse button. The other is to move the mouse away from the easel and click either button.

To erase a small area of the easel, use one of the DRAW nibs and the color

(to page 62)

STEP

BY MAURICE MOLYNEAUX

The Danger of Power Tools

Power tends to corrupt and absolute power corrupts absolutely." —*Lord Acton*

How many times have you heard all or part of that famous quote? It's been repeated so often that most people welcome it about as warmly as they would a visit from a collection agency. Despite the fact that it's almost a walking, talking cliché, it does seem to hold a sizable (but perhaps not ironclad) and bitter grain of truth. I can't possibly imagine how many times an argument has raged about what that quote is really saying, and whether or not it is absolutely correct.

No, I'm not going to engage in a philosophical discussion about what Lord Acton meant. However, I want you to keep that quote in the back of your mind as you read this article, because it's relevant to the theme here.

Power Tools

There's no question in my mind that computers can be a great aid in life. Of course, the computer alone can and will do absolutely *nothing* for you; conversely, it's what you do with the computer that matters. I'm fast approaching my fourth year as an ST user, and my fifth year as a serious computer user, and I've been doing a lot of reflection on how these machines—my STs in particular—have shaped and changed my life. I went from struggling to find a way to channel my interests into a job to making a career out of doing what I want.

I've taken to referring to my ST as my all-purpose tool because I use it for just about everything. Want to write a story? Whip out a word processor. Want to send an article to ST-LOG? Turn on the modem, boot *Flash!* and the editor has my article in minutes. Want to design a spaceship? Use the *Cyber Studio*, and I can model it in 3-D and look at it from any angle. Want to animate a cartoon? Dig out *Film Director* and make Megabit Mouse tap dance. Want to make it more interesting? Genlock Megabit so he's dancing on my shoulder.

But playing around with what you *want* to do isn't everything. A computer can also help you get what you *need*. CD ROMs (just wish Atari would ship theirs!) are making encyclopedias, dictionaries and other references quicker and easier to use by eliminating the time you'd spend digging through massive volumes for the information. On-line databases and information services provide powerful research facilities where you can access more information than the average public library, capture it in a form you can edit to your needs (keep this part, throw away the excess) and print out to keep a nice copy for future reference.

Stewart Brand's excellent book, *The Media Lab, Inventing the Future at M.I.T.* (I recommend this so highly I'd almost urge you to hunt down a copy this very minute) gives a tantalizing glimpse of what kinds of fantastic capabilities technology may give us in the near future. Personalized electronic newspapers, high-tech telephone answering systems that can actually screen calls, take messages and remind you to call your mother on her birthday; fiber-optic lines into every household, where home computers/video systems could tap into massive databases and download everything from the latest stock quotes to complete movies. There are scientists, educators and businesspeople all over the world struggling with the problems of making technology improve people's lives, giving them the power to do whatever they please, be it music, animated cartoons, learning an obscure African dialect or watching every single movie made that features someone speaking the word "kumquat".

But there's trouble brewing in paradise. No, not some new symptom unique to the information age, but some old, unwelcome and dubious "friends": dishonesty and greed.

Something for (next to) nothing

"Information wants to be free because it has become so cheap to distribute, copy and recombine—too cheap to meter. It wants to be expensive because it can be immeasurably valuable to the recipient."

—Stewart Brand, The Media Lab

Information, and not just of the "news" variety, has indeed become cheap. It's almost *too* cheap. All one has to do is look how far we've come in the past decade to get an idea of how technology has changed our lives. In 1979, Atari was just getting to the task of launching the 400 and 800 computers, each starting with 8 to 16K of RAM. Throw in a disk drive and a color monitor, and you were looking at $2,000 easily. Fast-forward to today—a Mega ST4 (with 256 times as much memory as that first 400!) with a color monitor and a hard disk runs about $3,000 retail. A lot more bang for your buck, especially when you compensate for inflation.

But that example just covers the hardware end of it. And while the hardware makes everything possible, it's the information created and used by the hardware that's of key importance. Let's look back exactly 11 years to the August 1978 issue of the late *Fantastic Films* magazine. *Star Wars* was all the rage in such publications, and on page 61 of that issue we find an ad selling "Star Wars Home Movies" (remember, this is before the VCR really took off as a mass consumer item). What were they offering?

"Now you can see almost 8 minutes of exciting scenes from *Star Wars* in a specially condensed Super-8 version . . ." And the price? $8.95 for a black-and-white silent version! $17.95 for color silent version, and $29.95 for a color and sound version! Thirty dollars for eight edited minutes! I've seen boxed sets of all three Star Wars movies in video stores for about $80! That translates to roughly 20 cents per minute for the videos and $3.75 per minute for the 8mm film. That's almost 19 times cheaper!

Now, how many people do you suppose bought a film like that? Not a lot. How many people have tapes of the *Star Wars* films? Lots. Next question, how many people *bought* those tapes at a video store, and how many people copied someone else's tape? (We won't discuss taping it off a TV broadcast because it's irrelevant at this point in the discussion.) A blank videotape can be had for as little as $5, renting a VCR, another few bucks. Now how about records, cassettes and CDs? A new one costs between $8 and $16. A blank cassette costs a buck or two. People make tapes for their friends all the time. You see a couple of kids buy a tape in a record store, and there's little doubt that one keeps the original and gives his buddy a copy.

Now, without making any proclamations about the rightness or wrongness of that, let's take a quick peek at computer software. An average ST program costs about $40. A disk costs about a buck. If you have one computer and one disk drive, you can probably copy the disk.

See? It's so cheap to reproduce these items that it's not even funny. It would seem that with VCRs, digital audio, computers, fax machines and photocopiers, every medium would be riddled with illegitimate copies running hither and yon.

But it ain't so. Take a look at book publishing. I can pick up a 250-page paperback for between four and five dollars. How much does 250 sheets of typing paper cost? Getting close, eh? How about photocopying the book? Let's see, if we lay it flat and copy two pages per sheet (single-sided copy), it's going to take 125 sheets of paper to make the copy. If a cheap photocopier charges four cents a page, it's going to cost $5 to copy the book, more than the cost of buying a legitimate copy. Could you imagine photocopying a newspaper? It costs maybe 35 to 50 cents and would cost you probably $10 to photocopy.

So, what we begin to see is that the industries that don't have problems with people making copies are those where the real goods are less expensive than an illegitimate duplicate.

If I have it, is it mine?

Probably one of the stickiest questions around today is exactly what are you buying when you buy a book, videotape or computer program. This is where the copyright laws come in.

When you buy a toaster-oven, you own it. As far as the laws of this country are concerned, you own every molecule and fleck of dust in that appliance. You can do anything you want with it. Jump on it, throw it in the swimming pool or make toast. Your choice.

Now, when you buy a novel, what are you buying? Okay, yes, you bought a book, but what part of that book do you own? You own the paper it's printed on, the ink that makes the letters, even the cover; but, according to the law, what you don't own is the intellectual property behind that book. You don't own the exact sequence of words or the thoughts therein. You bought paper and ink and paid for the privilege to read the contents of the book. The same goes with software. What you've purchased when you buy *Alien Ninja Vixens from Hell* (in 3-D) is a cardboard box, some paper with ink on it and a floppy disk. In doing so, you have paid for the privilege of playing the game. Like the toaster-oven, you can stomp on the package, throw it in the pool or try to make toast with it. But the one thing you can *not* do is duplicate the intellectual property, which includes the text of the manual, the art on the box and the data on the disk.

Now here is where the question arises as to what you own. Whenever a discussion of this topic crops up, you have people who point out that when they buy that toaster-oven, they own it lock, stock and barrel, but when they buy a computer program, they are restricted, because they don't really own it. They own the disk and box and paper, but not the program proper or the instructions. This is a double standard, they say.

Or is it?

Let's go back and take a look at that toaster-oven. No intellectual property here, is there? Let's make sure. As we poke around the rear panel of this appliance we find some text, most of which warns us we can get zapped good if we mess around back here while the thing is plugged in. Nothing relevant there. Below that is a small sticker states very plainly, "Manufactured under the following U.S. Patents:"

Whoa! Intellectual property just reared its ugly head! The design of the components of this toaster-oven are patented. Legally, we can't duplicate them without permission of the patent holders. So, what we find is that this toaster-oven isn't much different from a book or a piece of software. You own the metal, the dial, the heating elements, but you don't own the design of the components. If you built a duplicate of the-oven, you'd be violating the laws by infringing on registered patents.

Interesting. We own every molecule of that toaster-oven, but not the exact sequence in which those molecules are arranged to make its parts. That is the intellectual property. The same holds true for cars, toothpaste tubes, VCR's, those toy Smurfs I burn to cheer me up on depressing days. In every case, we own the physical property but not the intellectual property. You own the car but you don't own the design of the patented rack-and-pinion steering. You own the toothpaste tube but not the design of the tube itself. (Heck, even the company logo has a little registered trademark symbol on it!) You own the plastic Smurf, but the look of their little blue buns are copyrighted.

So, in this light, a book, videotape, CD or piece of computer software is no different from a car or a toaster-oven. We own the material but not the intellectual property.

Apply here

According to the law, copying certain kinds of intellectual property is a major no-no. Okay, this is nothing new. I think I've more or less established that, on the ground level, software (all types, books, videotapes, etc.) is no different from hardware. You can't copy an Atari ST bit for bit without stepping on Atari's patents. You can't copy the data for *Ninja Vixens* without stepping on the software producer's copyright.

Now, let's say some unscrupulous person manages to get his hands on a photocopier and finds he can churn out photocopies of the latest hardcover Stephen King novel for one cent a sheet. Each sheet holds two pages of the book, which is 600 pages long. One complete photocopied manuscript is 300 sheets and costs him $3. He finds some customers willing to buy these copies from him for $4 a pop, while the hardcover goes for $18 in the bookstores. He has one photocopier, and it takes a while to grind out 10 complete manuscripts. Also, he's limited to the locals as his market. Thus, he does little harm all in all (not to say there's nothing wrong with what he's doing, it's not right at all).

Compare this to someone who decides to copy *Alien Ninja Vixens* for a few friends. His friends give away copies to other friends. Unlike a photocopy or a videotape, the copies are perfect, digitally identical to the original, with no degrading of quality. The disk copies in minutes, the data can be smashed into a single big file and sent across modems. Modems can reach just about anywhere. A copy of the game can go from California to New York in minutes. One way or another, the disk ends up in the hands of someone who runs a BBS (Bulletin Board System) on his ST, and who has no qualms about posting copyrighted software where others can download it. The software spreads from board to board, user to user, until there are literally thousands of illegitimate copies floating around the country.

The effect is a bit more devastating than the man selling a few dozen photocopies. Consider a Stephen King book may sell millions of copies in hardcover and paperback. A successful ST program is measured in the teens (yes, teens) of thousands. There are a few dozen bootleg King novels and a few thousand bootleg copies of *Ninja Vixens*. Who stands to remain in business longer, King or the publisher of *Vixens*?

This is where the power principle comes in. Modern technology is too easy to abuse. How many of you never gave a second thought to copying an audio tape or a piece of computer software? Would

you have second thoughts about stealing the design for Atari's Megafile 44 removable hard disk and using it? Probably.

The rub

There is a serious possibility for abuse in our high-tech world. A computer, while a marvelous tool that allows us to do things otherwise impossible, expensive or difficult, can also be a dreadful weapon—one many people wield without realizing just what they are doing: computer viruses, phone phreakers passing around and using other people's credit card numbers, hackers who break into the computers of government agencies and hospitals to play with their records, pirates passing around copyrighted software as if they had every right to.

One thing it seems our modern society has to learn—and I'm talking from the youngest child to the eldest statesman—is that with power there comes responsibility. We have to learn to think beyond our own wants and see the greater consequences of our actions. From TV to videos to music to computer software, the prevailing idea is to get it for nothing. Don't buy it if you don't have to. That doesn't say too much for what kinds of values we're passing down from generation to generation, does it?

And this is where we can say, "Power corrupts." Why? Because when we have the power to do these things, we begin to think we *should* be able to do these things. We no longer question if we're right or wrong, we no longer look at what we're doing as stealing (whether it be an idea, money, whatever), we just say, "Hey, it's my right to do what I want," and leave it at that.

So, I'd have to say that I must generally agree with the first part of Lord Acton's statement, as I see it happening before my eyes. As to ". . .and absolute power corrupts absolutely," I'll reserve judgement until I meet someone with absolute power.

The real world

The potential for high-tech abuse is rampant. In an excellent "Footnotes" column a few months back, Michael Banks discussed the all-too-real dangers and temptations for misusing modern information technology (for plagiarism and so on), so I won't retread that ground here. But on the specific topic of software piracy, here I'll concentrate.

The piracy end of this issue would seem to have been beaten to death, but it hasn't. There are all kinds of rationalizations and explanations bandied about to try to legitimize its practice in some circumstances. Some say they aren't really pirating, they:

—Just get a copy to see if they want to buy it.

—Want to make sure it does what they want and need.

Really? And how many do buy it once they have it? And how many who decide not to buy it ever erase those bootleg copies? Very few.

Then there are those who rationalize it, saying they aren't hurting anyone or the ST community because:

—They wouldn't have bought it anyway.

—One copy won't make a difference.

To address the first excuse, just because you wouldn't have bought it anyway doesn't make up for the fact that you're taking something that you have absolutely no right to possess—unless you pay for a legal copy of it. Secondly, sure, one copy probably doesn't make any difference, but if everyone thinks that, then it's not just one copy anymore.

For several years now, I've seen ST users complaining that software publishers blame their poor sales on piracy, and who say that the Atari community is so pirate-riddled that it's unprofitable. I can't speak from the point of view of those publishers, but I can speak from my own experience. I have been paying attention to what I see and hear. The Atari World Show in Anaheim last April was a real eye-opener. Aside from wandering around the show, I was helping out at the CodeHead Software booth, where not only did we find people were stealing diskless manuals (presumably to have the instructions when they got a bootleg copy of the software), but I also heard a lot of interesting comments from users. Let me quote a few of them:

"My brother buys all the software, and he just sends stuff to me over the modem."

"*We* want to buy a copy of —."

"My friend gave me a copy of *MultiDesk*."

And they say this in front of the guys who *wrote* the stuff! I got the impression that a lot of them had no idea that what they were doing was wrong. Furthermore, a lot (and I mean *a lot*) of average users were telling us that they used TOS 1.4. Odd, because as of this date TOS 1.4 is not released on ROM, and the only versions out there are disk-based Beta versions, which are supposed to be only for licensed developers. Yet, it seemed half the people who talked to us had a copy. Sure, the disk-based versions of TOS 1.4 aren't being sold, so technically it's not hurting Atari moneywise, but the very idea that so much of the ST community has bootleg copies of the under-development operating system makes the mind reel.

I've had other users try to talk me into trading copies of software. I've been at user-group meetings where one guy will sit down at his computer and grind out copies of copyrighted software for anyone who asks, and I've been appalled to find that just about every computer user I've ever met has at least a few bootleg programs in his/her software library.

I've heard some ST users state that piracy on the ST is, percentagewise, not as bad as on other brands of computers, that ST users are, all in all, better educated. My experience doesn't tend to make me agree with them, but if they are right, then I'm going to start to feel that only a tiny fraction of computer users are honest. . .and that isn't the kind of thing I want to think about.

To close, if and when ever you find yourself tempted to provide other users (or yourself) with illegitimate copies of commercial software, I'd like you to try to recall the following quote by Richard Brinsley Sheridan, which I think sums this subject up quite nicely:

"Be just before you're generous."

As the disk turns

If you find this topic of discussion a little out of place here, I'm not surprised. Step 1 is not an editorial platform, but it is supposed to be here to help novice users understand their STs better. A number of people I talked to said they felt the question of software ownership was an important one and that I should cover it for those who are new to computing or unaware of these aspects of it. I agreed, particularly in light of what I have seen and believe to be a widespread problem.

Next issue, the topic will be a little more upbeat, as I'll be covering in depth the myriad ways a personal computer can help people reach goals and make it easy for them to do what they want. The downside of that for those of you who enjoy this column is that the next Step 1 will be the last. Yup, it all wraps up next time.

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. ■

SOFTWARE

BY KARL E. WIEGERS

Over the past several months, we've talked a lot about concepts that distinguish the discipline of software engineering from the art of computer programming. The underlying theme has been that a disciplined, structured approach to software development will result in a final product of higher quality and greater durability than will a more casual approach. This durability becomes particularly important when your software system enters the maintenance phase, which is everything that you do to the system after you thought you were done with it. Another of our goals is to design and write program modules so that they can be reused in future projects, thereby saving us the pain of continually reinventing all sorts of wheels.

We've talked a lot about how to decompose even a very complex software system into a collection of well-defined, bite-sized processes. If we do this properly, we can go off and write program modules (subroutines, functions, procedures) in our favorite language with confidence that we'll be able to plug the pieces together successfully at the end. Last time, we looked at some ideas for documenting our final product, in the form of both internal comments describing each specific program module and external documentation showing how the system as a whole fits together and operates.

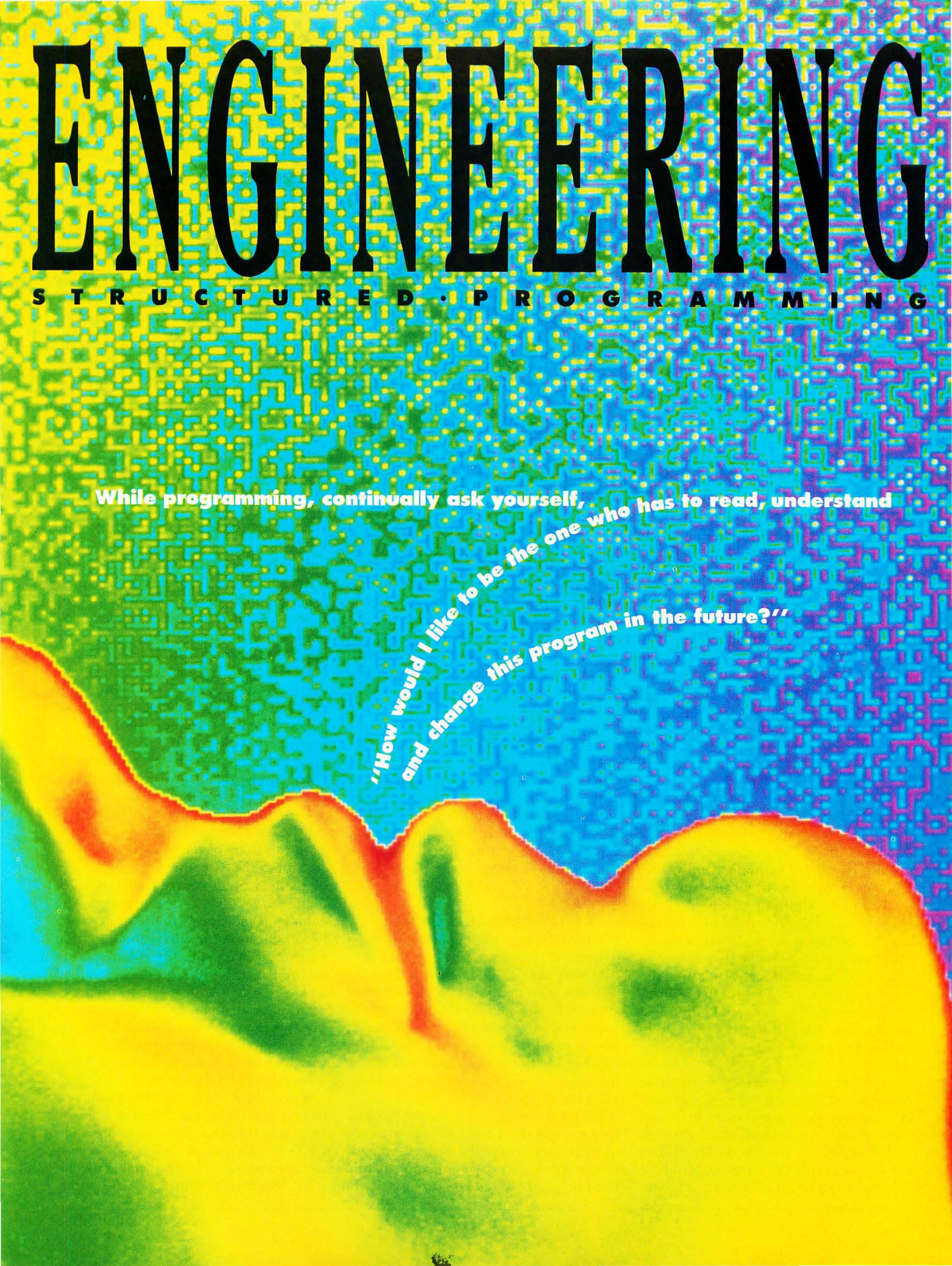
The one aspect of software development we really haven't addressed is the actual writing of the computer program itself. I've been making the assumption that most of you are seasoned programmers, with some experience in modern structured languages such as C, Pascal or GFA BASIC. But perhaps the time is ripe to re-hash a topic that helped begin the software engineering revolution: structured programming. Some other thoughts that aren't specifically part of structured programming might be described better as smart programming. I touched on some of these basic ideas back in the November 1988 issue; now let's take a closer look.

PHOTO: DIGITAL ART

ENGINEERING

STRUCTURED · PROGRAMMING

While programming, continually ask yourself,
"How would I like to be the one who has to read, understand
and change this program in the future?"



What is structured programming?

It's difficult to provide a concise answer to this question. In some ways, it fits into the "I know it when I see it" category, although we will discuss some specific structured programming principles shortly. But on the other hand, structured programming is consistent with our general philosophy of software engineering, which emphasizes enhanced communication between people. Computer hardware has become powerful enough that programmers should no longer emphasize compactness of code and cleverness of algorithm on their road to the nirvana of "efficiency." Contemporary software development stresses clarity and structure of code, not cuteness.

While programming, continually ask yourself, "How would I like to be the one who has to read, understand and change this program in the future?" Structured programming can be a powerful aid to effective software maintenance. And in some ways, structured programming is an attitude, as well as a behavior. The emphasis is on person-to-person efficiency, not computer efficiency.

The golden rule

In my opinion, the single most important idea of structured programming is that the code you write should represent a clear, simple and straightforward solution to the problem at hand. Keep this guiding principle in mind as you read the rest of this article.

Start at the top

One of the basic precepts of structured programming has been with us clear through our software engineering sojourn: a structured program consists of a hierarchical collection of individual modules, which appear more abstract at the top levels and more detailed at the lower levels. This fits with our overall strategy of hierarchical decomposition, which we've followed from structured system specification through structured design and now down to structured programming. The process of building a program in this fashion is called "top-down programming," or step-wise refinement.

Another point I stressed during our discussion of program design is that each of these modules (or processes, in design terms) communicates with others through well-defined data interfaces. These data interfaces typically are subroutine parameter lists or function argument lists. Each part of the program appears to other parts as a black box that simply performs its assigned function in some unknown way.

You've encountered this idea every time you called a built-in function in some programming language. Think about 8-bit Atari BASIC. Do you recall the STICK function? It told you something about the position of the joystick. Do you know how to communicate with the STICK function? Yes; all you had to do was pass it the number of the joystick you were interested in, like this: STICK(1), and it returned a numeric answer. Do you know how it worked? No; could be magic, for all you know. Do you care? No. This is the beauty of a "black box" approach to software development. There's no reason why the modules you write should be any different in this regard than the modules supplied by the guys who wrote the language you're using.

In practice, you apply the notion of step-wise refinement by writing your initial description of each fundamental process in a very high-level "language" that we called pseudocode. This is a first attempt at a picture of how each process will accomplish its assigned task. As you continue down the path from design toward code, you add detail to this description until eventually you reach something that conforms to the exact syntax of the language you're using: source code.

I suppose we could consider that one additional step takes place even after this, which is the compiling of your source code into something the computer can deal with: object code. Fortunately, we humans can halt our step-wise refinement at the source code stage and let the machine take over from there. (Someday, we'll be able to stop at the pseudocode stage.)

The dreaded GOTO

The first thing most people learn about structured programming is that you shouldn't use GOTO statements. This notion stems from an article published in 1968 by E. W. Dijkstra, which helped the structured programming push get underway in earnest. The succinct title said it all: "Go To Statement Considered Harmful" (*Communications of the ACM*, 11 (3), 147-148).

We don't always have a choice. In older forms of BASIC, GOTO statements were needed everywhere, because there just wasn't the richness of commands that we need to avoid GOTO. The careless use of GOTO inevitably leads to the notorious "spaghetti code" that makes a program nearly impossible to comprehend and debug. The worst case is a GOTO that branches back to a previous statement in the program listing.

Modern programming languages provide logic and control commands that allow us to almost completely avoid using GOTO statements (we'll discuss these shortly). However, there are still a few situations in which a GOTO actually can result in cleaner, more understandable code. Error-handling situations sometimes benefit from GOTOs. Premature exits from loops, or breaking out of deeply nested IF structures, may be more easily handled with a GOTO than by some other method. Nonetheless, the general guideline that GOTOs should not be used for routine transfer of control within a program is still valid, so try to break any lingering bad habits from your earlier experiences with BASIC interpreters.

A question of style

The programming style you use can greatly influence the readability of your code. While not strictly part of structured programming, there are some matters of style to keep in mind. A pretty good book on this topic is *The Elements of Programming Style*, Second Edition, by Kernighan and Plauger (McGraw-Hill, 1978). It's somewhat dated, and the code examples are all in FORTRAN or PL/I, but the principles remain the same. The following represent some of the highlights of programming style in my mind; some don't apply to all programming languages.

Use indentation to visually block logically related sections of your code, such as the sections of IF/ELSE IF/ELSE/END IF and SELECT/CASE constructs. Use blank lines in the source code to further delineate sections of the program. Use comments judiciously in the source code for clarification (we talked about this last time), and make sure the comments are accurate. Don't bother to document bad code—rewrite it, instead. Never put more than one statement on the same source line. We always did this in Atari BASIC because it saved six bytes per statement, but you don't need such tricks when you have a megabyte at your disposal.

Select meaningful variable and procedure names. Don't use different names to represent the same piece of information (refer to your data dictionary). Explicitly declare the type of each variable used, if your language permits this. Use paren-

theses to resolve any ambiguities in mathematical expressions, even if they aren't required for the operation to be executed correctly. Make sure conditional tests (IF some condition THEN do something) read clearly. Generally speaking, the first condition tested for should be the desired condition, with an error condition handled second.

Program defensively: Try to anticipate all possible errors in input data or mathematical operations, and write code to handle such situations. This includes validating input data before trying to use it and testing for such mathematical problems as division by 0 or taking the logarithm of a negative number. Make sure that input data does not exceed the bounds of what the routine can handle. Think of the user when designing your programs; make input easy to prepare correctly, and make output self-explanatory.

Initialize variables before using them. Who knows what was in those bytes before they were reserved for a variable's use? Avoid multiple entry points, and exits from loops and subroutines (more about this later).

The first priority is to get the program running correctly. You can worry about optimization later. And when you do, make sure the program still runs correctly. Don't try to optimize every little step; the compiler will do a lot of this for you. Usually, a program spends most of its time in a small section of the code, so concentrate your optimization efforts here (if you can find it).

Use the best algorithms you can find for calculations, but remember that both the algorithm and the structure of your data will influence how the algorithm will be implemented in code. Insert "instrumentation" checkpoints in your programs to write out intermediate results someplace; so that you can verify accuracy, track down errors and assess efficiency. These outputs can be sent to a trace file on the disk, which you can then examine at your leisure.

Building blocks

Another basic premise of structured programming states that any program logic, no matter how complex, can be expressed in terms of just three kinds of logical operations: sequence, selection and iteration. A program then is made up of a series of blocks of code to perform these operations. Let's define these three kinds of operations.

Sequence—a series of program statements are executed one after the other, in the order in which they appear in the source code. Obviously, this rules out statements like GOTO and IF, restricting us just to statements that perform some specific action. (A CALL or GOSUB to another procedure would qualify as an action in this sense.) Hence, a block of statements that are executed sequentially is called an "action block."

Selection—one set of statements, from a choice of two or more, is selected for sequential execution, based on some criterion. One way to accomplish this is to use an IF/THEN/ELSE construct. Some of the languages will also permit a SELECT/CASE/OTHERWISE/END - type structure, perhaps with different but analogous keywords. The set of statements that gets executed in each case is itself an action block. Sometimes selection constructs are called "branch blocks."

Iteration—a series of statements is executed repeatedly until some termination condition is met. These are also called "loop blocks." Virtually all languages contain simple FOR/NEXT or DO/END-type loops. More modern languages include variations such as DO UNTIL/END and DO WHILE/END.

These three kinds of "control blocks" have some features in common. First, the code in each is executed from top to bottom, which is the same way that it appears in the source file. This makes the program much easier to read and understand than does the convoluted branching you find in so many BASIC programs. Of course, in a selection block, not every statement is executed, and in a loop block they may be executed more than once, but they still are always executed from top to bottom.

In addition, each control block has just one logical entry point: the first statement. And if they're well structured, they have just one logical exit point: the last statement. A complete program is written by assembling and nesting blocks of these three kinds to perform the required processing.

Believe it or not, it's possible to write more or less structured programs in BASIC by following these rules. Some simulation of certain missing language features is required and some GOTOs inevitably creep in. But by keeping the notion of just three flavors of control blocks in mind, a surprisingly good job of structured programming can be done. If you still use BASIC, I encourage you to read a series of articles on structured programming in BASIC by Arthur Luehrmann in the May, June and July 1984 issues of *Creative Computing*. Unfortunately, *Creative Computing* is no longer published, but perhaps you or a friend has these back issues in the dusty magazine archives.

More iteration

I imagine you're pretty comfortable with the ideas behind action blocks and selection blocks, but let's take a close look at the iteration, or repetition, constructs. I'm sure you're familiar with the simplest type of loop, which looks like this in BASIC:

```
FOR I = 1 TO 10
  calculate something
NEXT I
```

In other languages, such a loop is commonly called a DO loop and is terminated by an END or END DO statement:

```
DO I = 1 TO 10
  calculate something
END
```

In either case, some test is used to determine the number of times the loop is executed. In these simple examples, a variable called *I* (the index variable) is incremented after each iteration and compared to the value 10. If *I* is less than or equal to 10, the statements in the loop are executed again; otherwise, execution of the loop terminates. Here we're assuming that the value of *I* will go up by one on each iteration. Of course, you can set a different step interval with a statement like: DO I = 1 TO 10 BY 0.5.

Here's the key question: Is the comparison done *before* the loop is executed or *after*? There's a big difference. Suppose *I* has a value of 20 at the time this loop is encountered in the course of executing the program. Will the loop be executed (since the value of the index variable is already greater than 10) or not?

In Atari BASIC, the comparison is done after the contents of the loop are executed, so the loop is always executed at least once. I think this is generally true of simple FOR/NEXT and DO/END loops. Many modern languages resolve any ambiguity by providing two explicit statement choices: DO WHILE and DO UNTIL. In a DO UNTIL loop, the termination condition is tested at the end of the loop, so the loop is always executed at least once. The simple FOR/NEXT loop in BASIC is thus a DO UNTIL type loop. In a DO WHILE construct, the termination condition is tested at the beginning of the loop. If the termination condition is already true, the contents of the loop aren't executed at all.

One important point is that DO

WHILE and DO UNTIL loops need not rely on a changing index variable in the termination test. Any logical expression can be used, such as: DO UNTIL STATUS = 'DONE'. It also may be possible to have complex combinations of termination conditions, either of which could cause loop execution to cease. Consider this example, which will terminate either when *I* is greater than 100 or when *J* becomes less than or equal to 30:

```
DO I = 1 TO 100 BY 10 WHILE J > 30
  calculate something
END
```

Maybe it will help to see a visual representation of these two looping structures. I'm sure you're familiar with flowcharts, in which action statements are represented with rectangles and decisions with diamonds. Figure 1a uses a fragment of a flowchart to illustrate that, in a DO WHILE loop, the conditional test is done before the action statements are executed. In Figure 1b, you see that the conditional test for a DO UNTIL is performed after the action statements are executed. I'm always getting these two confused, so maybe these simple diagrams will help.

Once upon a time we talked about using another diagramming technique to depict the structure of a program module: action diagrams. You may recall that, in an action diagram, a loop is denoted by a double horizontal line, which encloses the action statements in the loop within brackets. You can distinguish between DO WHILE and DO UNTIL logic by placing the double horizontal line at the top or the bottom of the loop bracket, respectively. Figure 2 shows what you get.

There are times when you want execution of a loop to end prematurely, for a variety of reasons. Bad input data may have been encountered, some error condition may have cropped up or whatever. Remember that we want to write our code so that each control block has only one logical exit point, so let's resist the impulse to GOTO out of the loop. A DO WHILE is a good approach, because you can put the anticipated error or exit conditions into the WHILE clause:

```
DO I = 1 TO 10 WHILE EOF = 'FALSE' AND INPUT = 'OK'
```

But things can get tricky if you don't have a DO WHILE available; you may just have to violate the one-exit rule. Some languages provide a command like LEAVE for premature loop exits. In BASIC, you can set the index variable to its termination condition, so that the loop isn't executed anymore; FORTRAN won't let you do this. A more graceful solution is to use an IF block that tests for the abnormal termination condition right inside your DO block:

```
DO I = 1 TO 10
  IF EOF = 'FALSE' AND INPUT = 'OK' THEN
    DO
      calculate something
    END
  END IF
END
```

This is basically a way to simulate the DO WHILE command with your own explicit test. Note the innermost DO/END block without any loop criteria on the DO statement. This is really defining an action block to be executed within the selection block (IF/END IF), not a loop block. If you prefer, you can think of this as a loop block that gets executed exactly one time.

FIGURE 1A

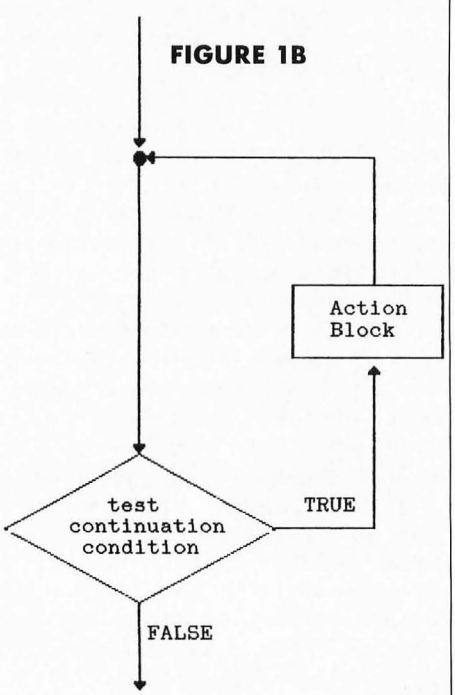


FIGURE 1B

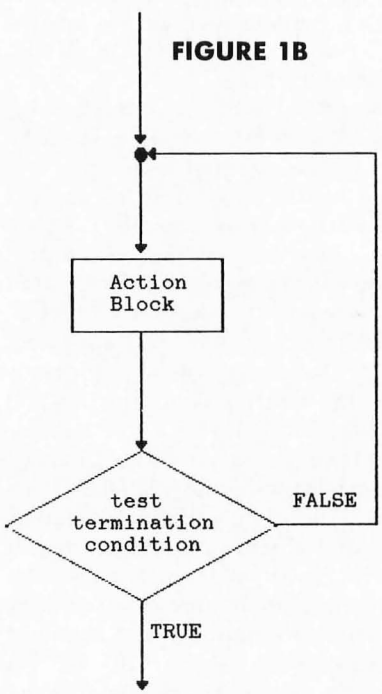
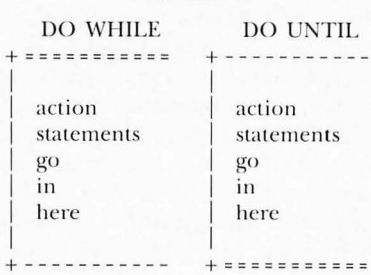


FIGURE 2



Let me out!

The one-entry, one-exit concept applies to individual modules just as it does to loops. Suppose you're calling a subroutine from a main program. You should always enter the subroutine at its top and exit at its bottom. This seems straightforward enough, but there are several ways things can go awry.

Some languages permit multiple entry points to a called procedure, but this is a bad practice. For example, in BASIC you may have seen the use of GOSUB in this form:

```
10 IF A=1 THEN GOSUB 1000
20 IF A=2 THEN GOSUB 2000
30 END
1000 calculate something
2000 calculate something else
2010 RETURN
```

Essentially what we have here is one subroutine (Lines 1000-2010) with two entry points (Line 1000 or Line 2000). If A = 1, then control is passed to Statement 1000, it drops through to Statement 2000, then returns at Statement 2010. But if A = 2, only the Lines 2000 and 2010 are executed. Concise, but very confusing; don't do it. If it's necessary to selectively execute

certain statements in the subroutine under only certain conditions, put a test for those conditions at the top of the subroutine, or (even better) write two separate subroutines.

Similarly, each subroutine should have only one exit point or RETURN statement. But now something of a dilemma crops up. Suppose that we're being good little software engineers and are programming very defensively, writing lots of code at the top of the subroutine to validate ("audit") the input data. If errors in the data are encountered, we want to abort the operations in the subroutine and return to the calling program with some indication of what went wrong. How can we handle this if we only want to have one exit point in the subroutine? I can think of three alternatives.

1) Forget about the one-exit rule and have multiple RETURN statements in the subroutine, one for each failed audit plus the proper one at the very end.

2) Forget about the no GOTO rule and have a GOTO after each failed audit, which transfers control to the single RETURN statement at the end of the subroutine.

3) Nest our audit tests (which are probably IF statements) as deep as necessary so that only conditions which pass all the tests finally work their way to the inner sanctum where the actual subroutine code resides.

It seems that we have a real quandary here. All of these approaches seem to violate one of the structured programming rules we're trying to abide by. Actually, the third scenario is okay if we have only a couple of audit conditions, because nesting IF statements two or three levels deep is perfectly fine. More levels of nesting than that becomes a problem both in terms of readability (do you want to match up six or seven END IF statements with the corresponding IFs on previous screens?) and because, if you're indenting nicely at each IF level, you may wind up with awfully short lines available for the actual subroutine code in the middle of the IFs. Not good.

My preference in handling the validation question is to think of the blocks of audit code as a "filter" through which the data must pass in order to wend its way to the heart of the module. I tend to put each audit block at the same level of indentation and have a RETURN statement in each block. The main processing section of the procedure has only a single RETURN statement.

Using the filter analogy, any time a piece of bad data is encountered, the RETURN statement acts as a block that prevents the process from continuing. I find that this approach keeps the code easy to read and modify, although I am violating the one exit rule. The next best solution would be to GOTO the single RETURN statement each time an audit fails, but I find this approach to be a little harder to read.

It might be useful to let the calling program know why processing was halted in a subroutine. The calling program will take different actions depending on the cause of the error, be it bad input data, mathematical impossibilities in the computations, file I/O problems or whatever. One way to do this is to pass a return code back to the calling program. The numeric value of the return code might represent the sequence number for an error message to display to the user. The interpretation of the error numbers could rely on a table or external file of messages or actions to match the return codes.

Alternatively, the return code might actually be the name of a procedure to call for handling the error or otherwise transferring execution control. By convention, a return code of 0 usually indicates that no error was encountered.

Wrapping up

The structured programming concepts we've talked about here are all geared toward making the programs you write more understandable to human beings. The computer doesn't care if your program makes sense, so long as it compiles properly. But anyone who must understand how your program works needs all the help he or she can get. You can provide that help not only by using good structured programming practices, but also by using common sense. Keeping your code clear and simple, rather than cute, condensed or clever, will go a long way toward writing programs that are easy to read, comprehend and alter. ▪

After receiving a Ph.D. in organic chemistry, ***Karl Wiegers*** *decided it was more fun to practice programming without a license. He is now a software engineer in the Eastman Kodak Photography Research Laboratories. He lives in Rochester, New York, with his wife, Chris, and the two cats required of all* ST-LOG *authors.*

FROM OVER THE BIG WATER

By Marshal M. Rosenthal

It's around this time of year when, thanks to the heat, we all dream of frigid December nights. Yes, there's still plenty of hot times ahead—especially with this latest crop of European software. So position one fan at your head—and another at your ST—and let's get on with it.

But first a correction. I noted two new programs from Psygnosis in the last issue, *Super Menace* (tentative title) and *Blood Money*. Well, I goofed. *Blood Money* is the name for what I called *Super Menace*—the other one being nameless at the time. Just goes to show what happens to brain cells when you stare at monitor screens too long!

Now that that's out of the way, I have some good news and some bad news. The bad news is that all of Earth's populace is dead, the planet a barren shell after colliding with a massive asteroid. The good news? You're now the big boss.

Not exactly a birthday card for the commander of Moonbase, which now holds the last remnants of humanity. Set in the year 2200, *Millennium 2.2* requires a lot of concentration and consideration. The stakes are huge, because it's now your task to repopulate the solar system with humans. This means a lot of hard choices and hard work. Technology must be created so that the spacecraft can be launched for colonization. People to fill the craft must be genetically manufactured. And then there're the "everyday" problems of coping with the Moon's hostile environment.

An icon bar at the top activates the various areas of interest and need, from computers to the research labs and mining facilities. Great detail has been applied to this game, from the atmospheric graphics to the galactic map that moves the way stars really do. Additionally, there's a wealth of accurately researched data to gather and explore in your quest to restore humanity to the universe.

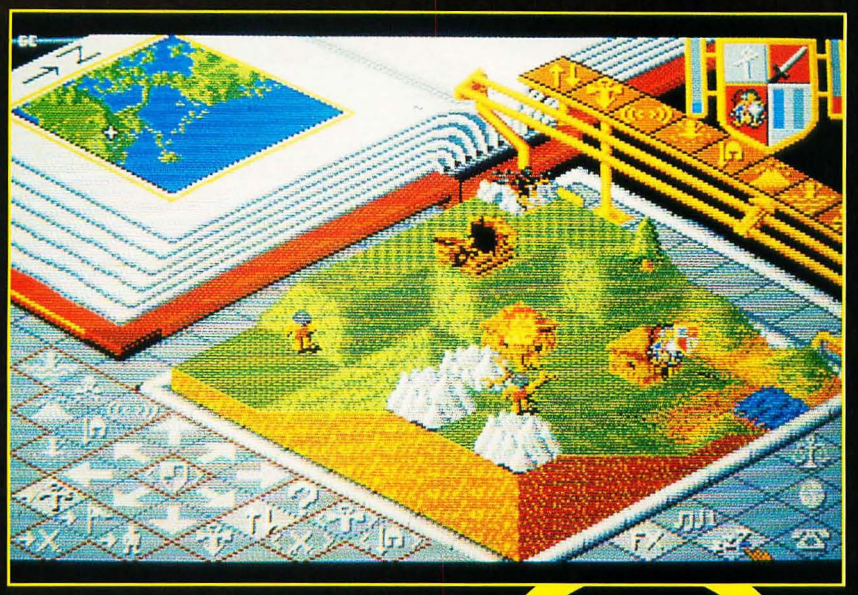
A lot of games are created from other ones; think about how many *Defenders* and *Pac-Mans* you've seen in different clothes. So here's a refreshing, if a bit odd, change from the usual: *Populous* from Electronic Arts U.K.

You finally made it to the top; you get to play God. Commanding one warlike nation is your chore, while the evil one schemes to overcome and destroy the guys that look to you for guidance. Expect to spend some time expanding your people's territories and helping populate them with the good guys. Give them a nudge at creating new technologies so that the other nation can be wiped off the face of the planet.

The first step is to clear some land so that cities can be built. Technology increases as the cities grow, and your divine power rises in proportion to the number of people and the level of their sophistication. Plan on creating knights to fight and destroy the enemy, while also harnessing the forces of nature. (Unfortunately, there are some giant birds and sea serpents that randomly show up to shake things about.) Digitized sound effects enhance such "natural" disasters as earthquakes, floods and volcanoes.

There are two modes of play. "Conquest" features multitudes of worlds based on varying terrains, the arctic being one, while "Custom" lets you pick the specifications for the world. There's a 3D map of the current world, which can be zoomed in on for minute investigation. The graphics are nifty and colorful. Play against the computer or connect via modem to another ST or an Amiga.

Also from EA is *Fusion* (created by Bullfrog Productions). The mission sounds simple: Just collect nine pieces of a bomb that's been scattered over 14 alien levels and return them to the first level. Of course, there are a few difficulties. First you have to pilot an Assault crawler in search of the Mother Ship (faster and with more firepower). Thank goodness there's joystick control; trying to maneuver around with a mouse would be insane here. Then expect to be baffled by a number of puzzles; you have to do some thinking here. Icons scattered around the alien



POPULOUS

FUSION

MILLENNIUM 2.2

SHINOBI

landscape are helpful because they add weapons and reenergize shields.

The sound effects are digitized and are quite good overall. Unfortunately, the opening music (also digitized) just doesn't make it with the ST, being full of distortion and noise. A tip: Don't keep waiting for the program to load just because you see the drive light on. Most European programs keep the drive "lit" even when it's not running, so once you tire of the tune, hit the spacebar to start up the game.

An on-screen information panel indicates the ship's structural integrity, shield level, active switches and bomb display units. A combination of arcade and strategy, *Fusion* is plenty of fun, what with the eight-direction scrolling and the digitized sound effects (including the use of a professional drum machine). Graphics are nicely detailed; nothing looks chintzy here.

Next up is Virgin Mastertronic's arcade transfer of *Shinobi—Master Ninja*. So new in fact, there's only a demo to look at right now. The Ring of Five, evil terrorists all, have kidnapped children of the world's leaders. The kids are held prisoner in different locations and guarded by vicious karate thugs. Your mission is to find and rescue these children. So it's time to tape up your fingers and grab the old joystick. You are well prepared, being a master at stealth and the martial arts, your body as lethal a weapon as any of those you carry, including sword, shuriken and nunchaku.

Each level will pit you against henchmen and a member of the evil ring—plus many surprises as you battle through five missions (19 levels). Graphics on the demo look good (there's that big helicopter waiting to pulverize or be pulverized), and karate fans will find this game worth sinking their shurikens into.

Aaar, mateys, it's a cosmic pirate's life for me. But things aren't as simple as they were in the days of Bluebeard. First you have to prove yourself worthy of the skull and crossbones, and that means working your way up the ranks. *Cosmic Pirate* (Palace Software) requires you to practice through a number of vector-graphic scenarios in order to learn the ins and outs of your craft. Points are awarded, which can be used towards purchasing a spaceship for use in a real mission. Then it's out into the galaxy in search of plunder and one of those fat Spacetrucks.

Plunder can be acquired by blasting. Follow the hyperspace beacon and check out the sector map. Then head for a new planet and see what you can find. Of

course, you have to keep out any competition (i.e. blow up other pirates), and any aliens you might meet in space should be neutralized (i.e. again, blast them to smithereens). There are far too many of these dorks, and they've a few bad surprises, like intelligent homing missiles.

Your ship comes equipped with guns and power shields; customizing it is a priority and can be done, once you've made a few bucks, after you've returned to base. All action is controlled by joystick, with menu features activated by the function keys. There's a snappy tune to listen to, and visual effects are neat (watching your ship blow up is actually fun). You've always wanted to be the bad guy, right? Well, now's your chance!

Next time we'll see what's cooking over at U.S. Gold in Britain. This is one company putting out a lot of stuff for the ST.

Marshal M. Rosenthal *is a New York-based photographer and writer specializing in children with product, video graphic enhancements and high-tech entertainment. His written/photographic projects have appeared in major publications in England, France, Germany, Sweden and the U.S.*

Product Information:

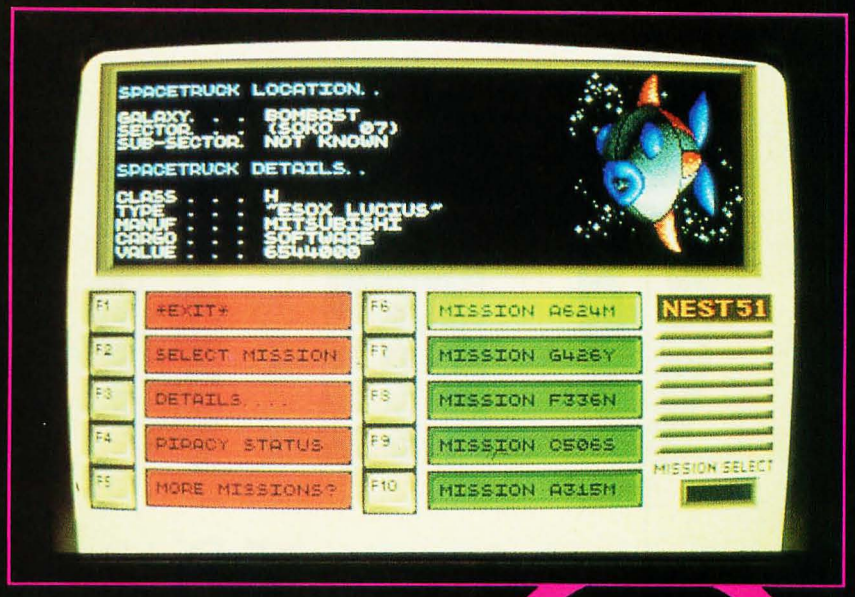
COSMIC PIRATE
The Old Forge Business Center
7 Caledonian Road
London, England W1V 3FF

FUSION
Electronic Arts Ltd.
Langley Business Centre
11-49 Station Road
Langley, Berkshire, England Sl3 8YN

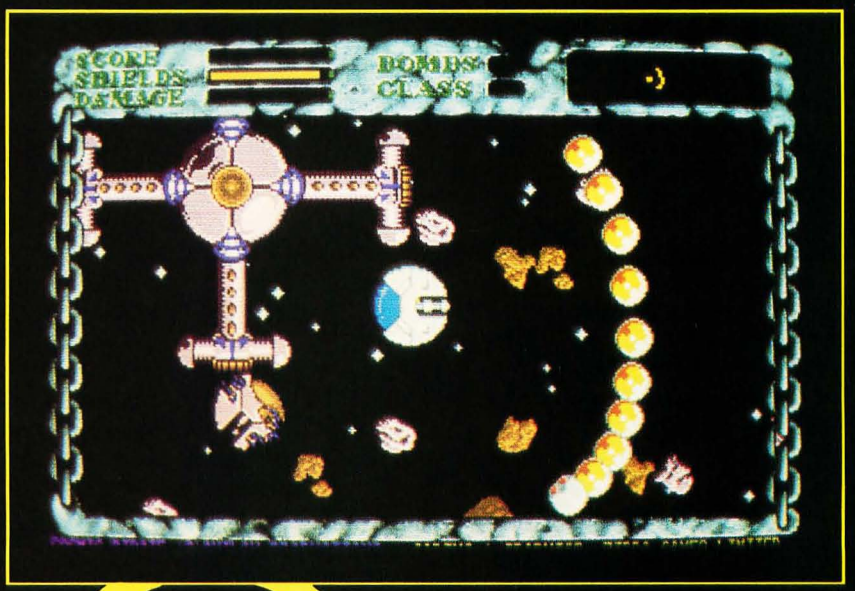
POPULOUS
Electronic Arts Ltd.

MILLENNIUM 2.2
Activision U.K. Ltd.
Blake House
Manor Farm Road
Reading, Berkshire, England
RG2 OJN

SHINOBI—MASTER NINJA
Virgin Mastertronic Ltd.
100 Campden Hill Road
London, England W8 7AR



COSMIC PIRATE



COSMIC PIRATE

COMDEX

SPRING '89

FRANK COHEN

In March, Atari Corp. made a significant showing at Cebit, the largest European computer trade show, held annually in Hannover, West Germany. On display were the new Atari Transputer Workstation (ATW); Atari 68030 machine (TT); a new hard-disk drive with removable disks; ST Laptop (Stacy); hand-held MS-DOS computer; and several other products that have yet to be shown in the United States.

On return to its home in Sunnyvale, California, Atari Corp. stopped in Chicago to display some of its wares at the Spring COMDEX computer trade show. COMDEX—the Computer Dealers Exposition—has in recent years drawn ever fewer actual dealers. Instead, the press, the distributors and the curious come to view what's new and what might become a future market leader. Atari sees COMDEX as an important public-

relations tool and, to a lesser extent, a good place to poll its audience.

The Atari booth, prominently positioned at the entrance to one of the main halls, was filled with the usual Atari Corp. personalities: Sam Tramiel, president; Sig Hartmann, corporate marketing; and Mel Stevens, trade show promotion director. The turnover of Atari employees has been just as fierce as previous trade shows. New on the scene is the executive trio of Atari Computers USA: Mike Dendo, V.P. sales; Joe Mendolia, V.P. marketing; and Tony Solarno, software marketing.

COMDEX is promoted as a business trade show, so the usual bag of Atari products surprisingly did not include MIDI or entertainment products. Atari's image as a game-machine company does not have to be promoted at COMDEX. However, its lead in the MIDI market is vulnerable to attack from PCs, the Macintosh and the Amiga. Among the various conferences during the show, Atari was not present at the MIDI conference, while several Amiga enthusiasts did attend, promoting the Amiga as the better MIDI machine.

Atari is aiming its higher-end ST machines at the desktop-publishing (DTP) market; yet, the Atari booth did little to promote the DTP solution it so zealously wants to pitch. Previous Atari trade-show exhibits have been effective in comparing the operating speeds of the Macintosh, Amiga and ST. Desktop publishing products on the level of *Calamus*, *PageStream* and *Desk Set II* fare very well in head-to-head challenge with *PageStream PC* or *Illustrator Mac*.

Atari's promotional failings are overshadowed by the new hardware the company will soon begin shipping to U.S. dealers. The commitment to new products seems never-ending, with new technology and improvements to existing platforms at the center of Atari's product-development efforts. Atari exhibited four new hardware products and two new software products, with promises of quick releases to dealers.

ST laptop

Perhaps the most exciting news from Atari is the introduction of Stacy, a laptop version of the 1040ST computer. Weighing in at under 15 pounds, including batteries, Stacy costs under $1,500 and comes with one megabyte of memory, an LCD Supertwist display, an exceptionally nice keyboard and one floppy-disk drive.

Atari expects a number of Stacy accessories to be available late this year. In the works are additional floppy-disk drives, an internal 40-megabyte hard-disk drive and two-/four-megabyte memory upgrade kits. Extended memory upgrades are important to MIDI users.

Although Atari expects MIDI revenue to account for the bulk of Stacy's sales, Wayne Smith, Southern California sales manager for Atari, foresees the Stacy being sold to government and municipal accounts. "Zenith is the only laptop company that has gone after police departments," Smith said. Smith sees the Los Angeles Police Department using Stacy to take police reports. Insurance companies, another type of organization that works outside the office, will also be a candidate for Atari's new laptop.

Macintosh users have also begun looking at Stacy. With the Spectre 128 cartridge, Stacy becomes an inexpensive Macintosh portable. Apple's plans for a Mac laptop seem to be stalled for the moment, providing Stacy with a boon of publicity.

A firm release date for Stacy was not available by press time; however, Tramiel promised the first shipments to begin in June/July 1989. Tramiel further pledged the first production runs of Stacy to the U.S. market.

The Stacy is the portable equivalent of a 1040ST computer with monochrome monitor. Since the TOS operating system has not changed, Stacy should be able to run all existing ST software and utilities. Minor problems might be encountered with the screen display, which is driven by a new custom chip called Shadow. The LCD screen operates fairly quickly, with a very subtle lag noticeable when the cursor is moved around the display. The screen may be tilted to several positions.

Built into the keyboard is a small trackball—an upside-down version of a mouse. Rolling the trackball affects the screen position of the cursor, just as a normal mouse would. Directly above the trackball are left and right mouse-button equivalents. GEM operations that expect a mouse button to be held down while the mouse is moved are difficult with the Stacy trackball. Stacy also has the two standard joystick/mouse connects, so a regular Atari mouse may be used.

Stacy's keyboard is the laptop's most impressive feature—a complete departure from Atari's existing line of computers. Each key has a complete mechanism, making fast touch-typing possible for the first time on Atari equipment. Secretaries accustomed to an IBM Selectric typewriter will be happy to use Stacy.

To the right of the LCD display are

▶ PORTFOLIO

A handheld IBM-compatible PC, Atari Computer's Portfolio, is about the same size as a VCR tape. Priced under $400, the system includes word processing software, a spreadsheet, a personal calendar and a phone directory

volume control knobs for the built-in speaker and a contrast knob that changes the display's angle of visibility. Above the knobs is a paper-holder clip that doubles as a trackball holder when the case is closed.

Several bus expansion plugs have been designed into Stacy, and removable plastic doors protect the machine from the elements. The rear panel protects the connector plugs for printers, modems, external hard disks, external floppy disks and RGB or monochrome monitor hookup. The rear panel doubles as a rear stand, positioning Stacy toward the user.

MS-DOS pocket portable

Amid rumors that an Atari pocket computer was being developed, Tramiel officially introduced the Portfolio hand-held MS-DOS computer at COMDEX. Portfolio is not the first MS-DOS hand-held, but Atari might certainly have the least expensive system: Portfolio carries a low $399.95 list price.

Inside its clamshell-style enclosure, Portfolio sports a four-line, 40-character LCD display. The text is large enough for most users; however, the display is big enough for simple tasks but not enough for most MS-DOS applications.

A small audio speaker produces key clicks to the left of the screen. The speaker can also be used to vocalize phone company touch-tone sounds as part of a phone-directory application that comes with the operating-system software.

The keyboard follows the QWERTY layout, but is much too small for touch typing. The key tops most resemble Hewlett Packard's scientific calculator keys, with foam pads being used for each key mechanism. Considering its size and weight, the keyboard is better than one might expect.

A slot on the lower left side allows data cards to be added to Portfolio. The cards add extra memory, programs and non-volatile memory to the system. Memory cards have an additional expansion bus so other cards can be "piggy-backed."

Built into the Portfolio ROM operating system are several applications: personal diary, phone book, worksheet and text editor. The programs are multilingual, with an easy software switch from English to French to German. Other applications can be loaded into Portfolio, but only through a communications data card, which won't be available with the release of the computer.

Three AA batteries power the Portfolio for approximately six hours of use. Designed by DIP, Ltd., of England, the software has an English tone, with English pound signs often found where American's might expect "$" characters.

Although Tramiel has promised a quick release to U.S. dealers, there are many parts missing from the complete Portfolio system. A ten-minute video presentation prepared by DIP, Ltd., shows the Portfolio used by salesmen taking spreadsheet figures to a jobsite away from his/her office. Data from *Lotus 123* on an IBM PC is downloaded into a data card, which is then plugged into the Portfolio. Data is transferred to the data card through a device that looks like a half-height floppy-disk drive that can read and write to a data card. Atari did not know if the data-card transfer box would ever be available for the Portfolio.

Atari also lacked a clear understanding of the expansion cartridges. Attaching a serial or parallel printer, modem or other external device requires a data cartridge. Atari has in the past been equally vague about products that were given a minimal marketing effort.

The ST removable

For two years, Atari has displayed prototypes of its CD-ROM device. More than 300 megabytes of data can be stored on a compact disc (CD); the CD-ROM unit allows an ST computer access to the CD data.

"The CD-ROM is now shipping in West Germany," said Joe Mendolia, V.P. marketing U.S., "But I am not certain if it will make it into American distribution. Our plate is very full right now." Mendolia expects his view to change once plans for existing product introductions are complete.

Currently on Atari's plate is a new hard-disk drive that features a removable platter. Manufactured by Syquest, the removable disk drive has been put into a Mega ST-type box. Platters are removed through the front of the cabinet, just like floppies. Each platter stores 44 megabytes of data, with an average access time of only 25 milliseconds.

The device plugs into the DMA port. DMA Out is included, so the removable may be daisy-chained with other devices (e.g., laser printer, tape streamer, hard-disk drive, etc.). The platters will have a list price of $150, with the drive unit costing $1,199. Mendolia expects dealer shipments to begin in August 1989.

The PC99?

Atari's latest attempt at the MS-DOS personal-computer market, the Atari PC4, was unveiled at COMDEX. Atari introduced its PC1, a low-end IBM PC XT clone at the 1987 Fall COMDEX trade show, as reported in ST-LOG. Atari's hopes were to compete with the low-end IBM PC-XT-clone manufacturers, and as an afterthought, they expected PC dealers to pick up the ST line. Since then, Atari has shown the PC2 (with IBM-style expansion slots) and the PC3 (offering the more advanced 80286 central processing unit compatibility).

The PC4 offers a 1.2-megabyte 5¼-inch floppy drive and a 60-megabyte hard disk. The system is equipped with one megabyte of RAM, with expansion slots up to eight megabytes. Video output is user-selectable for monochrome, Hercules or color displays. The system is equipped with CGA, EGA and VGA output; VGA is currently the most sophisticated video display adapter available to IBM PC users.

The PC4 has a suggested retail price of $1,995, which is competitive with other clone manufacturers, though not a bargain. Atari sees the PC4 sold to small-business and vertical-market system users looking for an IBM PC-AT clone.

Previous Atari PC machines came with bundled software from Digital Research, the makers of the GEM operating system; however, the PC4 is bundled with Microsoft Windows, a competing visually driven operating system to GEM. Also included with the Windows operating system is MS-DOS Version 3.3, GW-BASIC Version 3.2, *Windows Paint* (a drawing program similar to *Macpaint* for the Macintosh) and *Windows Write* (modeled after *Microsoft Word*).

A major drawback of the PC4's predecessors has been resolved with the inclusion of one PC/XT-compatible expansion slot and five PC/AT slots. Obtaining the true power of a PC requires plugging in expansion boards that add more memory, ports and other accessories. Previous Atari PC machines have lacked this ability. The PC4 also comes with a standard serial, parallel and mouse I/O port, as well as a socket for an 80287 math coprocessor.

While the Atari PC has sold fairly well in West Germany, it has fallen on its face in England, Canada and France and was never released in the U.S. market. Atari's price point, lack of local service centers and product specification deficiencies have kept the Atari PC out of the U.S. market.

The ST invisible

Nowhere to be seen on the COMDEX floor was the rumored Atari 68030 ST computer, commonly known as the Atari TT. As the Atari ST celebrates its fourth birthday, you might expect Lorne Greene to say, "That's one hundred human years

in the lifetime of a computer." Atari enthusiasts eagerly anticipated the public display of the next generation of Atari computers. Fearing a backlash of negative publicity due to the company's previous announced-but-never-shipped products, the TT was not shown.

"The TT is in the prototype stage," Tramiel told a group of Atari software developers attending a cocktail party after the show. Two hundred and fifty TT units are under construction in Atari's Taiwan plant, to be distributed to developers over the next few months.

Tramiel told developers the TT will come with TOS 1.4, making it compatible with existing ST software. Atari has tested *DynaCADD*, the powerful Computer Aided Design (CAD) system from ISD Marketing, with the prototype TT computer.

The TT will also come with the UNIX 5.3 operating system, popular with educational and scientific institutions. In the upcoming battle between microcomputers and workstations, Atari could become a winner by offering GEM, UNIX, Macintosh (using Spectre 128) and MS-DOS (using PC-DITTO II) compatibility.

Internal software

Atari created a new position, director of software marketing, 22 days prior to COMDEX. Antonio Salerno is responsible for bringing new Atari software products to market: desktop publishing, word processing, PostScript emulation and several utility programs.

Desk Set 2 is a sophisticated desktop-publishing system from G.O. Graphics. Atari announced *Desk Set 2* at last year's COMDEX in Las Vegas and promised shipment by July 1989. At $299, *Desk Set 2* is not for every ST user: It requires a Mega ST4 and SLM804 laser printer.

G.O. Graphics has licensed a large library of fonts from Compugraphic, the world's leading manufacturer of typesetting equipment. The fonts are created by describing the outline of each character. When a large 48-point font is needed, the font description is enlarged mathematically to create the necessary screen or printer font. The results are smooth curves and slick strokes.

The overwhelmingly negative feature of *Desk Set 2* is the program's slow screen redrawing speed. When changes are made to an area of text, the program redraws the entire page, including those objects that are not visible within the GEM window displaying the document. Once the drawing is complete, the user may speedily scroll around the document. Most desktop-publishing programs redraw only the portion of the document that fits in the GEM window. To the experienced end-user, the *Desk Set* method of redrawing the screen will appear too slow to be practical.

Fairly wordy

As a replacement for the failed *Microsoft Write* word processor, Atari unveiled *Word Flair* ($149 List), a word processor with graphics ability and a built-in database.

"*Word Flair* is a compound document processor," said Lauren Flanegan-Sellers, chairman of Blue Chip Software, the software development firm that created the new Atari-licensed product. *Word Flair* works much like a desktop-publishing system: Objects and text areas are defined, then the actual text of a document is added later.

Flanegan-Sellers approached Atari Corp. in 1987 to produce an object-oriented word processor that incorporated the functions demanded by small-businesspeople in Blue Chip's computer training courses. Three years and 250,000 lines of Mark Williams C code later, *Word Flair* is poised to be a significant contender in the ST word-processing market.

Word Flair is a GDOS application, making it immediately compatible with the large library of fonts available from Atari, as well as the printer drivers available from companies such as MichTron and Neocept. GDOS is Atari's enhancement to the GEM operating system that allows programs to display various fonts and graphics on the ST screen and then print the results to several dot-matrix printers or the Atari SLM804 laser printer. The added graphics ability usually retards ST programs—not true in *Word Flair*'s case. The program speeds along, displaying stylized fonts and graphics while keeping up with the end-user.

Scanned images and clip-art may be pasted into a *Word Flair* document. The program recognizes the .IMG graphic file format, used by most image scanners and PC-based graphics programs. *Word Flair* may use the clip-art images sold by Migraph, a leading ST graphics firm, and a conversion utility is available for *DEGAS Elite* files. Images may be scaled up or down and moved anywhere within the document. Cropping an image is unavailable.

The program functions as a desktop-publishing system: A palette of objects is used to create regions of text and graphics within the open document. Any number of columns of text may be established with the quick click of a mouse button. Text may either be imported from an ASCII disk file or typed directly into the document. Macintosh-style text selection using the mouse is supported with Cut, Copy and Paste appearing as standard drop-down menu functions.

Word Flair displays a document in two views: Actual and Whole Page. Actual shows a close-up view of the text, making detailed editing easy. Whole Page shows the layout of the entire page, allowing quick changes to large portions of the document. Text is entered and edited directly on the document window, unlike *Desk Set 2*, which requires the use of a dialog box. The program's text layout does not support leading (varying the space between lines) and kearning (varying the space between letters).

A database program is built in, allowing mailing lists and other data to be stored and summarized. Data can be used as part of a *Word Flair* document, and financial calculations and bar graphs are easily created with the built-in database commands. Within five minutes, Flannegan-Sellers created and printed a form letter that graphically showed the sales performance of several real estate contractors.

Help screens are context-sensitive, and every dialog box includes a help button. Help screens appear in a GEM window next to the *Word Flair* document, so the help information can be used while working on a document.

At first glance, *Word Flair* is a winner. The few flaws all have to do with GDOS. *Word Flair* relies on GDOS to print a document, and GDOS treats all documents as graphic images. Most businesspeople do not have the five to seven minutes required to print a GDOS document on a dot-matrix printer. With the Atari laser printer, this time lag is reduced to less than 30 seconds; however, the laser printer is an expensive accessory for most ST owners. *Microsoft Write* overcame this problem by offering a "print text only" option, which *Word Flair* does not offer.

Ethernet/Cheapernet

For years, Atari has promised a local-area network (LAN) for the ST. LANs allow many computers to share disk drives, printers and other devices. To the end-user, files stored on one ST's hard disk will appear on the GEM desktop as though the hard disk was connected to the local ST. Lack of an ST LAN has hindered the ST in the American small-business market.

"Our approach was to make a LAN compatible with all computers," said Tan

Sickmann, lead programmer for BioNet GMBH, based in Burbach, West Germany. Sickmann demonstrated BioNet 100, an Ethernet LAN for the ST and MS-DOS computers. Ethernet is a popular LAN that works well for high-speed data transmission: Sickmann's demonstration made it appear that the hard disk was on the local ST, when it was really attached to another system.

Although the $800 price tag is too high for most American ST users, BioNet is expensive when compared to equivalent PC-based LAN systems. The cost is based on hardware and software development expenses. Special patch software is placed in an ST computer's AUTO folder, making BioNet compatible with the most primitive levels of the ST operating system, and a small box providing Ethernet coaxial cables connects the ST to the rest of the LAN.

BioNet gives ST users access to the Digital Equipment Corporation DEC-NET international network, and corporations looking for inexpensive workstations can immediately connect the ST to the 800,000 DEC-NET nodes worldwide. An IBM PC/AT clone may also be connected as a file server using a BioNet 100 PC-card.

"We have tested BioNet with all the released versions of TOS," Sickmann said, "including TOS 1.4 and Blitter TOS." BioNet software is very sophisticated. Problems with viruses have been solved by placing BioNet software below the TOS BIOS, the part of the operating system that a virus attacks. Built-in network safety systems include password protection, system administrator functions and intelligent file-locking mechanisms that also detect an ST crashing off the network.

Desktop publishing

Atari announced a special package of desktop-publishing software and hardware, bringing the price tag of a complete system to under $4,000. The package includes a Mega ST4, a laser printer, a 30-megabyte hard disk, the *Timeworks Publisher ST* software and a library of 50 stylized fonts. Compared to separate purchases of all the items, the DTP package adds up to a $1,760 savings.

Atari appears to have shelved GDOS for Ultrascript, a PostScript clone that works with the Atari laser printers. Atari now promotes the SLM804 laser printer as "PostScript compatible." PostScript is a page-layout language, made popular by Apple with the graphically oriented Macintosh laser printer. Previously, Atari evangelistically relied on GDOS to print graphic images and stylized fonts. UltraScript was written by Imagen, a subsidiary of QMS, which manufactures, among other things, PostScript laser printers.

Atari's decision to bundle *Timeworks Publisher ST* aims the DTP package largely at the PC market. *Timeworks Publisher ST* is modeled after *Xerox Ventura Publisher*, the leading DTP package for MS-DOS computers. The other DTP systems for the ST are modeled after Macintosh DTP systems.

"We're going after *Pagemaker* on the Mac," said Nathan Protechin, marketing director of ISD Marketing. ISD introduced *Calamus Outline*, a rendering program that offers many of the features of *Freehand*, a leading Macintosh program. *Outline* supports splines, curved lines that flow through a set of points defined by the artist. The baseline of text may curve to a defined spline. Other features include color layers, access to a large font library and pattern fills. Graphics development with *Outline* may be imported into *Calamus*, ISD's high-end DTP system.

Protechin demonstrated a new version of *Calamus* that includes *Microsoft Write* and *WordPerfect* file import, text shadowing and a new outline font system. The new outline fonts allow *Calamus* users to individually specify the point size of any font. The program then mathematically calculates and draws the font to a display screen or printer. ISD licensed 84 fonts from FC Font Technologies, a division of Agfa/Compugraphic Corp., one of the largest U.S. manufacturers of typesetting equipment. Graphic artists will rejoice over the new fonts, which include the true versions of Garamond, Souvenir, Times, Avant Garde, Futura and Univers. Each typeface carries a $99.95 list price and works with *Calamus* and *Calamus Outline*.

For designers of type styles, the new *Calamus Font Editor* ($99.95 List) allows artists to create their own outline fonts. The program uses Bezier Curves—a special spline—to define the shape of each letter. The *Font Editor* disk includes 24 additional fonts, which may easily be modified by an ST user. The program is fairly sophisticated, supporting free rotation, italics, mirror imaging and font condensing functions. ISD also expects to have a *Calamus* image scanner available later this year.

The third method of getting professional graphics onto your ST screen was demonstrated in the latest version of *PageStream*, formerly *Publishing Partner Professional*. *PageStream* was developed to use a PostScript laser printer, then was later changed to support most of PostScript's command internally to the ST. The result is a sophisticated program that supports text rotation, kearning and leading.

Remote control

R/C Aerochopper ($199.95 List) is a sophisticated flight-simulation system that turns your ST computer into a remote-controlled glider, helicopter, jet fighter and trainer. The package includes the same transmitter box and controls as the Futaba Conquest series of remote controls used by the real hobbyist. Two joysticks control the power and direction of the remote-controlled aircraft. With the control in your hands, the simulation accurately displays the view of your aircraft, making maneuvers with adjustable wind conditions, adjustable control response and realistic sound effects.

Playing with *R/C Aerochopper* is like eating peanuts: One game is not enough. The disk-storage features save the 131 user-selectable flight characteristics. To practice certain maneuvers or to make things more challenging, the program has various camera control modes, trim adjustment and even random control glitches. When your aircraft crashes into the ground, rather than losing a $1,000 remote-control glider, you can hit Undo and keep playing.

Scanners

Migraph introduced several new scanners for the ST. Scanners make it easy to incorporate photos, drawings, logos and clippings into computer documents, and scanned images can be loaded into desktop-publishing systems, such as the Supercharged edition of Migraph's *Easy Draw*. The new scanners include a hand-held model, a full-page sheet-fed model and a flatbed model.

The Hand Scanner, expected at under $400 list, scans images up to four inches wide and has different scanning resolutions between 100 and 400 dots per inch with adjustable contrast. A special GEM desk accessory edits the scanned image, saving images in a variety of file formats: .IMG, .PCX, .PIC, *DEGAS Elite*, *MacPaint* and *PrintMaster*.

The sheet-fed and flatbed scanners operate at several resolutions up to 300 dots per inch. Scanned images are more accurate overall than the Hand Scanner. Prices are expected to run $1,000 for the sheet-fed scanner and $1,500 for the flatbed unit.

In support of the scanners, Migraph has introduced *Touch-Up* ($179.95), a

graphics application that works on bit-images not limited by screen size or resolution. Images that have been scanned almost always need corrections and touch-ups. The program combines a wide range of drawing tools with special effects, outline fonts and screen-editing functions. A special version of *Touch-Up* will be bundled with the Hand Scanner for only $400, and current *Touch-Up* owners will receive a special upgrade offer.

"We've had a problem with users not understanding the difference between *Touch-Up* and *Easy Draw*," said Liz Mitchell, V.P. operations for Migraph. *Touch-Up* was designed to create and edit bit-images that may be incorporated into a desktop-publishing program, such as *Supercharged Easy Draw, Calamus, PageStream* and others.

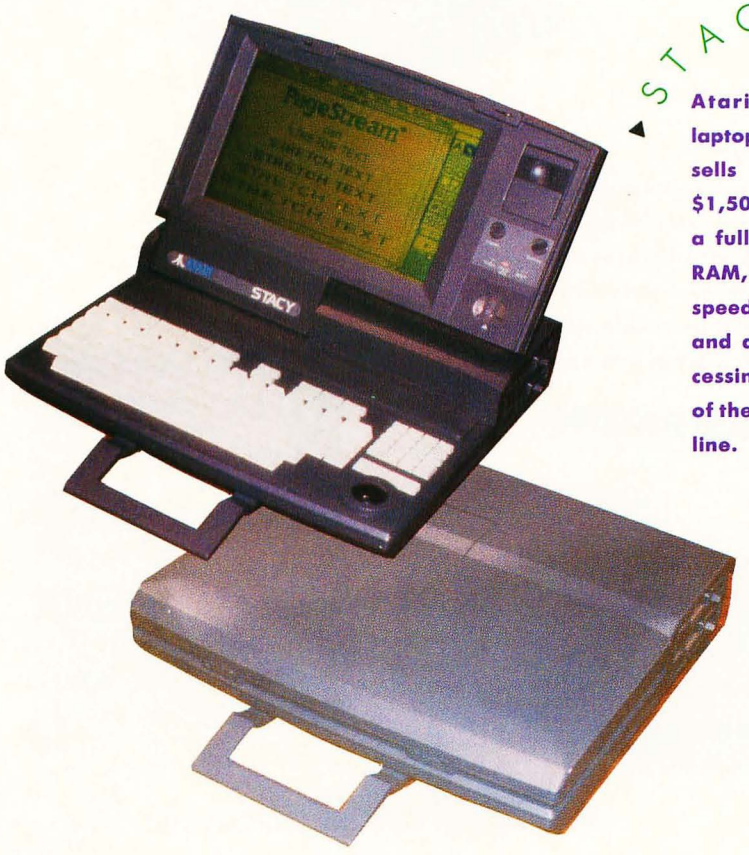
MichTron

Bouncing back from the troubles experienced with GFA Systemtechnik, the West German company that created GFA BASIC, MichTron introduced a new line of programming tools and utilities for the ST. HiSoft BASIC is MichTron's replacement for GFA BASIC 3.0. HiSoft BASIC is modeled after Microsoft Quick BASIC for the IBM-PC and Macintosh. Supporting HiSoft BASIC are Devpac ST (68000 Macro Assembler), *WERCS* (Resource Construction Set) and *SAVED* (Utilities Desk Accessory). The programs work together to provide a complete programming environment. HiSoft BASIC is also available for the Amiga; however, no plans have been made for MS-DOS or Macintosh compatibility. MichTron continues to sell the *GFA BASIC Training Reboot Camp* and the *GFA BASIC Programmers Reference Guide*, though they no longer market GFA BASIC.

Fast FAX is MichTron's announced facsimile-transmission system for the ST. Packaged in a silver external equipment box, *Fast FAX* connects to the ST's serial port and acts either as a 9600-baud FAX modem or a 2400-baud Hayes-compatible modem. Expect a $700 to $800 list price when the product begins shipping late this year.

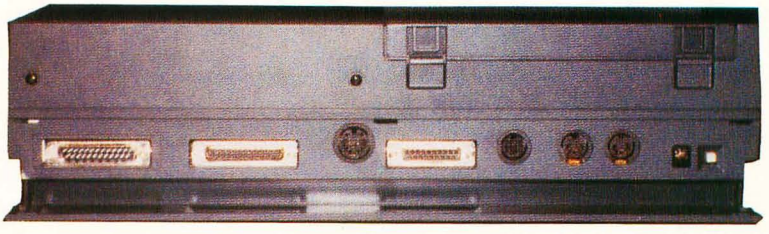
MichTron introduced *TailSpin* ($49.95 List), an icon-driven adventure creator. Functioning much like *Hypercard* for the Macintosh, live areas of an ST screen are delimited that cause actions and special effects. Color art images and sound effects may be incorporated into finished programs. A public-domain run-time module is included, making it possible to publish your own *TailSpin* programs.

Fleet Street Publisher 2 ($149.95 List) was demonstrated by George Miller,



STACY

Atari Computer's laptop PC, the Stacy, sells for less than $1,500 and includes a full megabyte of RAM, 8MHz system speed, LCD display and all of the processing capabilities of the company's ST line.



▶ MEGAFILE 44

Atari Computer has combined the portability of a floppy drive with the capacities and access times of a hard drive in the Megafile 44 hard drive with removable cartridge. The system provides ST users with virtually unlimited hard-disk storage in a single drive.

▶ COMPLETE, POSTSCRIPT-COMPATIBLE DESKTOP PUBLISHING SYSTEM

Atari Computer offers a complete DTP system, including the powerful Atari MEGA 4 personal computer, the PostScript-compatible SLM804-PCV laser printer, a Megafile 30 hard drive and Timeworks Desktop Publisher software for under $4,000.

MichTron's new director of Technical Support. *Fleet Street 2* offers true what-you-see-is-what-you-get displays on monochrome monitors (72 dots per inch.) The new version supports PostScript output files, multiple pages and a font library, and works with the stock 520ST color or monochrome computers.

Back-N-Up

ICD Technologies demonstrated the new Fast Tape Backup, a hardware accessory that archives data from a hard disk onto a specially designed audio tape. The unit moves data onto the tape at 6.5 megabytes per minute, about two minutes for an average 20-megabyte hard-disk drive.

"The tapes are thicker and have higher-quality coating than regular audio tape," said Craig Thom, ICD Technical Support. The extra coating makes the tapes, which cost $34.99, run faster and withstand high temperatures. Up to 155 megabytes can be recorded on each tape.

The backup unit has the same footprint as the Megafile 30. The unit plugs directly into an ST's Hard Disk/DMA port. DMA Out allows the backup unit to be daisy-chained to other devices: hard disk, laser printers, etc. ICD also sells hard-disk-equipped units. A battery-powered clock is built into the controller board, and individual file recovery is possible using the GEM-based backup software. TOS, MS-DOS, Spectre 128 and Magic Sac partitions are also supported in image-backup mode.

Also on the backup scene, *Mirror Image* ($27.95 List) is a new hard-disk backup from Computer Avenue. Developed by Ron Carter, *Mirror Image* makes a bit-image copy of a hard disk, storing the data onto a floppy. The program, written for beginners, estimates how many floppy diskettes will be needed before commencing the backup. Optional printed status reports are also available.

Mouse accessories

Often it becomes annoying to keep reaching from the mouse to keyboard while working with a program. *Fastkeys* ($39.95 List) is an ingenious new product that brings the ST keyboard to the mouse. *Fastkeys* is a plastic strip that sticks to the side of a mouse. A flat plastic runner attaches to the mouse cable, eventually running into the ST's cabinet. Under the ST keyboard, the runner terminates into a set of metal contacts that are attached to the switch contacts of several keys. When a button is pressed on the side of the mouse, the keyboard switch makes contact, fooling the ST into thinking that a key has been pressed. Since any keyboard

switch may be contacted, *Fastkeys* can simulate any keyboard function. *Fastkeys* comes with several commonly used function button labels: Return, Delete, Backspace, Undo, Shift, Tab, etc.

Several ST games use the calculator keypad to control the direction of a player, just like a joystick. *Keestick* ($6.95 List) is a new product that turns the calculator keypad into a joystick. *Keestick* places a small circular piece of rubber on the "5" key. A plastic wand attaches to the "5" key, pressing the "4," "8," "6" and "2" keys, depending on the direction of the wand. Although it might seem foolish to use *Keestick*, a noticeable difference can be felt when playing games that use the keyboard for movement.

Finally, for those of us who just use a plain mouse, there are full-color high-tech mouse pads. Precision Line has jumped into the four-color mouse pad market with a series of designer mouse pads. Graphics range from Japanese illustrations to photos of fish swimming around a coral reef. The company also offers mouse pads that feature a clear plastic lift-off cover. These pads allow you to put your own artwork under your mouse.

16 Mhz ST

Datafree Industries, a Toronto-based Canadian firm, has introduced an upgrade board for ST computers that pushes the operating speed of the CPU to 16 Megahertz—twice the usual speed. The 16 Megahertz Upgrade kit costs $199.95 and replaces the 68000 chip of any ST machine. Installation requires technical experience with electronics, as the 68000 chip must be removed from the ST motherboard (not an easy process.) The kit makes most ST software run 30% faster overall. Additional speed improvements will be made when a software blitter becomes available from Datafree later this year.

Ashes of Batteries Included

Atari old-timers have fond memories of Batteries Included (B.I.), a Canadian software house that provided most Atari 800 owners with their first word processor, *Paper Clip*. B.I. was eventually bought out by an oil company, then sold to Electronic Arts, another software-publishing company.

Martin Herzog, B.I.'s former marketing director, appeared at the Atari booth to "see what was still happening in Atariland." Out of the ashes of the old B.I., most of the employees are now back together at Delrina. Mark Skapinker,

▶ PORTFOLIO

Inside its clam-shell-style enclosure, Portfolio sports a four-line, 40-character LCD display. The text is large enough for most users; however, the display is big enough for simple tasks, but not enough for most MS-DOS applications.

former president of B.I., heads the new software-publishing company that specializes in MS-DOS applications.

Enhancements and previews

TOS, the ST operating system, is being upgraded. TOS 1.4 fixes many major bugs, gives ST users a new GEM file selector, adds extra options to the GEM Desktop and operates faster overall. TOS 1.4 was officially announced at COMDEX. Mega ST computers were the first this year to receive the chips.

Logical Design Works (LDW) demonstrated the enhanced version of what has become the most popular spreadsheet for the ST. *LDW Power*, Version 1.1, fixes several minor bugs, allows graphs to be saved in *DEGAS Elite* format and supports up to 32,000 points in a graph—the first version was limited to 100 points.

Mark Williams Company demonstrated the new version of its popular C compiler: Mark Williams C, Version 3.0. At $179.95 list, MWC 3.0 comes with a new resource editor that creates icons, menus, dialog boxes, forms and alerts. A new source-level debugger has also been released at a $69.95 list price. The debugger allows programmers to view the actual source code as expressions are evaluated, functions are traced and variables are displayed.

Williams has also begun shipping the Atari ST Library Source Code. By looking into the C Library, C programmers can learn the techniques used by Williams' professional programmers to develop programs. The Library Source Code has a list price of $149.95.

Abacus Software introduced *BeckerCAD ST* ($295 List), a professional computer-aided design system that supports PostScript output for finely detailed printed results. *BeckerCAD*'s standard features include polygons with any number of sides, freehand drawing, automatic line measurement, definable grids, multiple line widths, trim, divide, zoom and undo operations. Abacus' demonstration of *BeckerCAD* shows a sophisticated piece of software that runs on any ST with a minimum of one megabyte of memory—although two megabytes are recommended for complex drawings.

Abacus also introduced *Computer Viruses* ($2.95), a pocket booklet by Ralph Burger that covers the ins and outs of how viruses work and how to protect yourself from them. The booklet is available directly from Abacus.

▶ ATARI SCHOLASTIC SERIES

The Atari Scholastic Series brings the classroom to the ST. Available programs include: Biology, Chemistry, Physics and Algebra I and II.

▶ PC-4

With 1 MB of RAM (expandable to 8MB) and a 60-MB hard drive, the PC 4, Atari's 286 AT-compatible, offers the expanded memory required for today's CAD, graphics and desktop publishing applications.

Conclusion

COMDEX isn't drawing in new Atari dealers, as was the original intent of the trade show. Rather, Atari is making good use of COMDEX to mark milestones of progress in providing a sophisticated and broad platform to the American computer-buying public. Media and industry spokespeople come to the Atari booth every six months to see what Atari is up to, and they are often surprised to see new hardware being developed.

Atari tried to show its business image with flat black-metal panels highlighted with modern lamps casting a brilliant white light upwards to an Atari logo. The booth design was reminiscent of IBM's earlier trade shows, when the high-tech image was important.

The purge of Atari employees is never-ending. Mike Katz, head of Atari's entertainment division, left the company in February amidst a series of negative public remarks both from Atari and Katz. Shiraz Shivji, Atari's lead electronics engineer, was fired by Leonard Tramiel, head of development, because of a strained working relationship. Cindy Clavern, who administered just about all developer programs and user group functions, left Atari in March without explanation. Atari has fewer than 50 marketing and sales people in its Sunnyvale headquarters, not nearly enough to be able to administer the promised marketing program announced by Sam Tramiel at COMDEX.

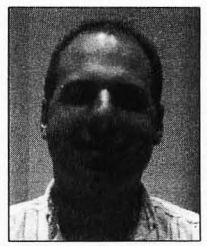
COMDEX focuses Atari's problem: They have a difficult time getting the message to the public because of disorganization brought on by frequent employee turnover. An example in point is the question of the newly announced three-pronged product-specific dealer channel.

"The three prongs are mass distribution," said Joe Mendolia, V.P. marketing, "That's really the Sears type of distribution. Next is the computer specialty dealer. Finally, there is the business computer center. That's a three-pronged approach." Mendolia describes specialty dealers as the bulk of current ST retail outlets, business computer centers handle the Mega ST and offer their own service and support for Atari equipment.

"I, very frankly, have to see the press release to see how it's worded," replied Mike Dendo, V.P. sales, when asked about the three-pronged dealer channel.

Andy Marken, the new Atari press relations representative, spelled out the three-pronged dealer channel as a push to rebuilding the existing dealer channel with more dealer-cooperative advertisements, beginning a series of end-user advertisements later this year and opening new markets to existing products.

Three different answers describing a new approach to the American market could only be the result of a group of new employees not knowing what each other is doing. The group of people, individually, get high marks for appearing open, honest and competent. Atari's future in the American market rests squarely on the amount of time these key employees stay with the company. ■



***Frank Cohen** has been developing Atari programs since his first commercial product,* Clowns & Balloons. *He later developed* Regent Base, *an SQL 4GL database, and is currently producing a series of videotapes teaching ST users how to make the most of their ST software. You may contact Frank directly on CompuServe (76004,1573) and GEnie (FRANK.COHEN), or by writing to P.O. Box 14628, Long Beach, CA 90803-1208.*

Companies Mentioned:

COMPUTER AVENUE
669 South Pearl Front
Denver, CO 80209
(303) 733-7727

SOUNDSIGHT CORPORATION
2105 Alcyona Drive
Los Angeles, CA 90068
(213) 463-9464

ATARI CORP.
1196 Borregas Avenue
Sunnyvale, CA 94086
(408) 745-2000

ISD MARKETING, INC.
2561 John Street, Unit #3
Markham, Ontario L3R 2W5
(416) 479-1880

BLUE CHIP INTERNATIONAL, INC.
501 Second Street #414
San Francisco, CA 94107
(415) 243-0188

DATAFREE INDUSTRIES, INC.
391 Ormont Drive
Weston, Ontario M9L 1N8
(416) 741-9825

MIGRAPH, INC.
200 S. 333rd, Suite 220
Federal Way, WA 98003
(800) 223-3729

MICHTRON
576 S. Telegraph
Pontiac, MI 48053
(313) 334-5700

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

LOGICAL DESIGN WORKS
780 Montague Expwy., Suite 403
San Jose, CA 95131
(408) 435-1445

PRECISION LINE INC.
14100 23rd Ave N.
Plymouth, MN 55447
(800) 328-0077

TIMEWORKS, INC.
444 Lake Cook Road
Deerfield, IL 60015-4919
(312) 948-9200

ABACUS
5370 52nd Street SE
Grand Rapids, MI 49512
(800) 451-4319

MARK WILLIAMS COMPANY
1430 West Wrightwood Ave.
Chicago, IL 60614
(800) 692-1700

ROTAL LTD.
5850 Cypress Gardens Blvd.
Winter Haven, FL 33884
(813) 324-0519

AMBROSIA MICROCOMPUTER PRODUCTS, INC.
98 W. 63rd. Street, Suite 371
Willowbrook, IL 60514
(312) 655-0610

COMDEX SPRING '89

END

(from page 37)

white (or your picture's background color). If you want to erase the entire easel, click on BLANK. To use magnify mode, click the magnification grid in the lower right-hand corner of the screen. Then move the mouse to the easel. The mouse will be replaced by a small rectangle. Move the rectangle to surround the area you want to magnify. Press the left mouse button. A red box will surround the identified section of the easel and a magnified representation of that section will be displayed in the magnification grid. Each square in the grid represents one pixel within the red box on the easel.

Now you can draw on the magnification grid directly (with the left button), and corresponding changes will automatically take place on the easel. Colors may be reselected while in the magnify mode. To exit the magnify mode, click the right button or click the left button anywhere on the screen except on the grid or the color choice area.

Pressing the UNDO key will erase the last change you made on the easel, but you must press UNDO before clicking the mouse button again. If you press UNDO just after exiting the magnify mode, all the changes you made while in that mode will be erased and your picture will be restored to its previous state. When you are done with a picture, click SAVE to type in your puzzle word and save the puzzle to disk. When typing the word, be sure to spell it correctly. You may use only the letters from "A" to "Z" in the twelve-letter puzzle word. Hyphenated words, apostrophes and multiple-word entries are not allowed. The Backspace key may be used to edit.

If there is no file called WUZZLERS.DAT on the root directory of the disk in the drive from which the program was loaded, the program will create one when it saves your puzzle. Two opportunities to change your mind are offered during the saving process; when you first click on SAVE and after you have typed in the puzzle word.

The LOAD option allows you to view all the puzzles in WUZZLERS.DAT in sequence. However, it also allows you to (in effect) delete puzzles by transferring pictures between WUZZLERS.DAT files on different disks. The method is a bit roundabout, as follows: Suppose you want to create a file of puzzles for your first-grader. Suppose, too, that you have 20 puzzles on another disk, and a few of these are appropriate to the child's skills. Use the LOAD option to load the desired puzzles from the one disk, then save them on a separate disk (placed in the same drive) for your child.

It's a good idea to develop a library of puzzle disks, with each disk containing puzzles for only one skill level. An appropriate selection of puzzle disks representing various skill levels and interests will enhance the game's value to each potential player in your home. Then, whenever you play the game, simply insert the desired puzzle disk in the same drive from which the program was loaded.

Some suggestions for creating puzzles

It goes without saying that the people who create the puzzles won't find the game very challenging. But fine art does have its rewards, and players will be grateful for carefully prepared puzzles. Samples of puzzles for players of various skill levels are included on your ST-LOG disk. In actual use, however, it is best if all the puzzles on a disk are designed for the same skill level.

For the benefit of early readers, puzzle pieces will provide more substantial hints if the object representing the puzzle word occupies most of the easel. The puzzle is divided into a grid of 3×3 blocks. It is important that each block have something in it other than the default white background; otherwise, players won't notice that a hint has been offered when an all-white piece is displayed in response to an incorrect guess.

For young readers, I would suggest pictures in which a single object is prominently displayed against a simple background. My five-year-old became angry when trying to solve a puzzle showing a girl wearing a hat. She kept thinking that the mystery word was "hed" (sic). When she discovered that the puzzle word was "hat," she threw up her hands in frustration and said, "Why didn't you just draw a hat, then?!" Puzzles for young children should be, with very few exceptions, obvious representations of nouns.

More difficult puzzles are likely to have considerable distracting detail. They may express parts of speech other than nouns, rather than being strictly representational (for example, a picture of a sailboat used to hint at the word "sailing," or a picture of a rain cloud for "stormy"). Rebus constructions can be especially challenging. Sophisticated puzzles will prove too difficult for early readers but will make for a challenging game for older children and their parents.

Whether the puzzles are simple or complex, it is important that puzzle words be spelled correctly when newly created puzzles are saved. The program does not include a spell-checker, so you will have to take this responsibility upon yourself. When planning your puzzles, remember that puzzle words may contain only the letters "A" through "Z" and are limited to twelve characters.

Little warnings

When you load WUZZLERS.PRG, whatever drive you use will serve as the only active drive for the program that session. Any puzzle disk you use will need to be inserted in that same drive in order for the puzzles to be loaded. The program will not run out of a folder. The program will operate from the root directory of a hard-drive partition, but this is not recommended. Although a hard disk might accommodate a huge puzzle file, which would be an advantage in a lengthy game, specialized puzzle disks will not be readily available and the LOAD/SAVE method of transferring desired puzzles will be complicated by extra steps.

During a game, the program avoids showing duplicate puzzles from a disk by assigning a number to each puzzle in WUZZLERS.DAT. If you switch puzzle disks before the purple alert box announces, "There are no more puzzles on this disk!" the program will not load those puzzles from the second disk, which are assigned the same number as previously shown puzzles from the first disk.

With its fine color graphics, pla
easy-to-use mouse, the ST is

A word about kid-friendliness

Early on in the development of this program, I set a goal: The game must be accessible to even the youngest beginning readers and the file-creation program should be usable for children only slightly older. This goal meant that the program needed a great deal of user-friendliness, much more than would be necessary for an adults-only audience. I have written many programs for children, and testing with my own children and in schools has demonstrated that some of these programs have a more successful user-interface than others. Sometimes these adaptations for ease of use limit the programmer's (and even the user's)

choices in frustrating ways. A certain amount of sacrifice is in order.

As I approached this program, for example, I knew I would not be able to use file selector boxes, created by a simple one-line command in GFA BASIC, because young children find them confusing. I would also need to limit choices to avoid too much confusion and screen congestion. Large text, color-coding of selection boxes and instructions, minimal keyboard use, large targets for the mouse and careful planning for users' mistakes all contribute to an easy-to-use interface for children.

It was not difficult to provide a kid-friendly interface for the game itself, but the file-creation program was another matter. I had planned to provide some sample pictures in my release of the game. I could draw them with *DEGAS*, but could a typical 8-year-old use *DEGAS* and follow instructions for creating a puzzle from his art? (Yes, perhaps your child could, but I'm talking about a typical one.) Not likely. I would therefore need to provide a drawing program of my own for the creation of puzzle files.

The drawing program would have to be capable of producing the kind of pictures I could draw with *DEGAS* but nevertheless be somehow less complicated. What *DEGAS* features could I live without? After careful consideration and some experimentation, I determined that I couldn't possibly draw to my own satisfaction without at least the following functions: DRAW, LINE, FILL, FRAME and CIRCLE.

stic-encased, durable disks and an ideal machine for children.

But my list didn't really stop there. If I was going to have FILL, I also wanted fill patterns. I wanted to be able to erase the entire drawing screen and I wanted to be able to erase sections of the drawing screen relatively quickly. I wanted a fair selection of colors and I wanted magnify mode. Most importantly, because I frequently change my mind, I wanted a way to undo my mistakes quickly and easily. And because I was not only a programmer but also a potential user of the program, it seemed safe to assume that most users would also want these functions.

Thus, all of the above operations have been provided in the drawing program, but others were omitted in order to keep it as uncomplicated as possible. This was not done without some regret, as I am reminded whenever I look for the now non-existent Copy option. The drawing program is still not quite easy, but it also isn't too difficult for a typical eight-year-old to master.

In my favorite word processor, *1st Word Plus*, I can select boldface type in three different ways: by pressing the F1 key, by clicking an icon at the bottom of the screen or by selecting Bold from the Style menu. Multiple methods of performing a single function can be confusing to children, and normally, I avoid them. However, in this program I provided two methods to end a drawing procedure without completing it. You can click the right button, or you can click either button (or both, by the way) while the mouse is away from the easel.

Likewise, to exit magnify mode, you can click the right mouse button or you can click the left button, as long as the mouse is not on the magnification grid or in the color-choice area. In this particular case, I selfishly sacrificed my one-choice policy in order to permit me to use the right mouse button, which I much prefer. Very young children—and some will use this program—sometimes have trouble telling left from right. They would not be able to use the right-button method successfully. In cases where the users cannot tell left from right, I would encourage them to use the left button routinely and exit a drawing operation by clicking the left button while the mouse is in the blank area below the SAVE and HELP options. If they can't remember which button is the left one, mark it or put a sticker on it to make identification easier. Users who can distinguish right from left, however, will almost undoubtedly prefer the right-button method.

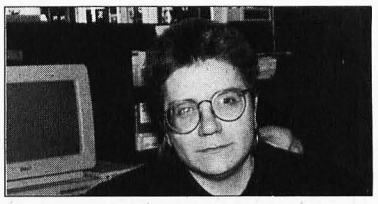
Further contributions to user-friendliness can be seen in the alert boxes on the Blank, Save and Exit options, providing the user with an opportunity to change his mind. These features were not included solely to benefit young users but should be a part of any program, no matter what the user's age.

The current functioning of the Load option and my suggestions for file transfers were also the result of an effort to provide a friendly interface. This device was added to satisfy an obvious need for transferring and viewing files, while not adding too much to the complexity of program use or to the congestion on the screen. There is still some potential for confusion, but then again, at least young users will not have to deal with a file selector box.

Children tend to flay about wildly with the mouse. While intentional "dragging" of the mouse may be difficult for them, children often unintentionally move the mouse while the button is pressed. This can result in unexpected selections; jerking the mouse upward from the easel, for example, could result in the inadvertent selection of HELP, LOAD, SAVE or EXIT. To prevent this, each time the mouse leaves a target area, the program checks to make sure the button has been released before permitting further selections.

When you select SAVE, the program checks to make sure that there is enough room to save another puzzle. When you select PLAY or LOAD, the program makes sure that there is a puzzle file (WUZZLERS.DAT) in the root directory of the disk. Yet another trap prevents saving a picture without an accompanying puzzle word. In each of these cases, a purple alert box will bring the nature of the error to your attention, and then the program will resume. Some mistakes are not announced by the program. If a player fails to type his name when beginning a game and simply presses RETURN at the prompt, for example, the program will adapt to this lack of information.

In spite of (and sometimes because of) the special considerations in program design required, programming for children is very rewarding. With its fine color graphics, plastic-encased, durable disks, and easy-to-use mouse, the ST is an ideal machine for children. My own children have received many hours of enjoyment from my programs (and from those of countless other programmers as well), and I hope that you and your child will thoroughly enjoy *Wuzzlers*! ■



***D.A. Brumleve**, M.A., is involved with children and computers in a variety of ways. The mother of five children, ages 2 to 10, 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.*

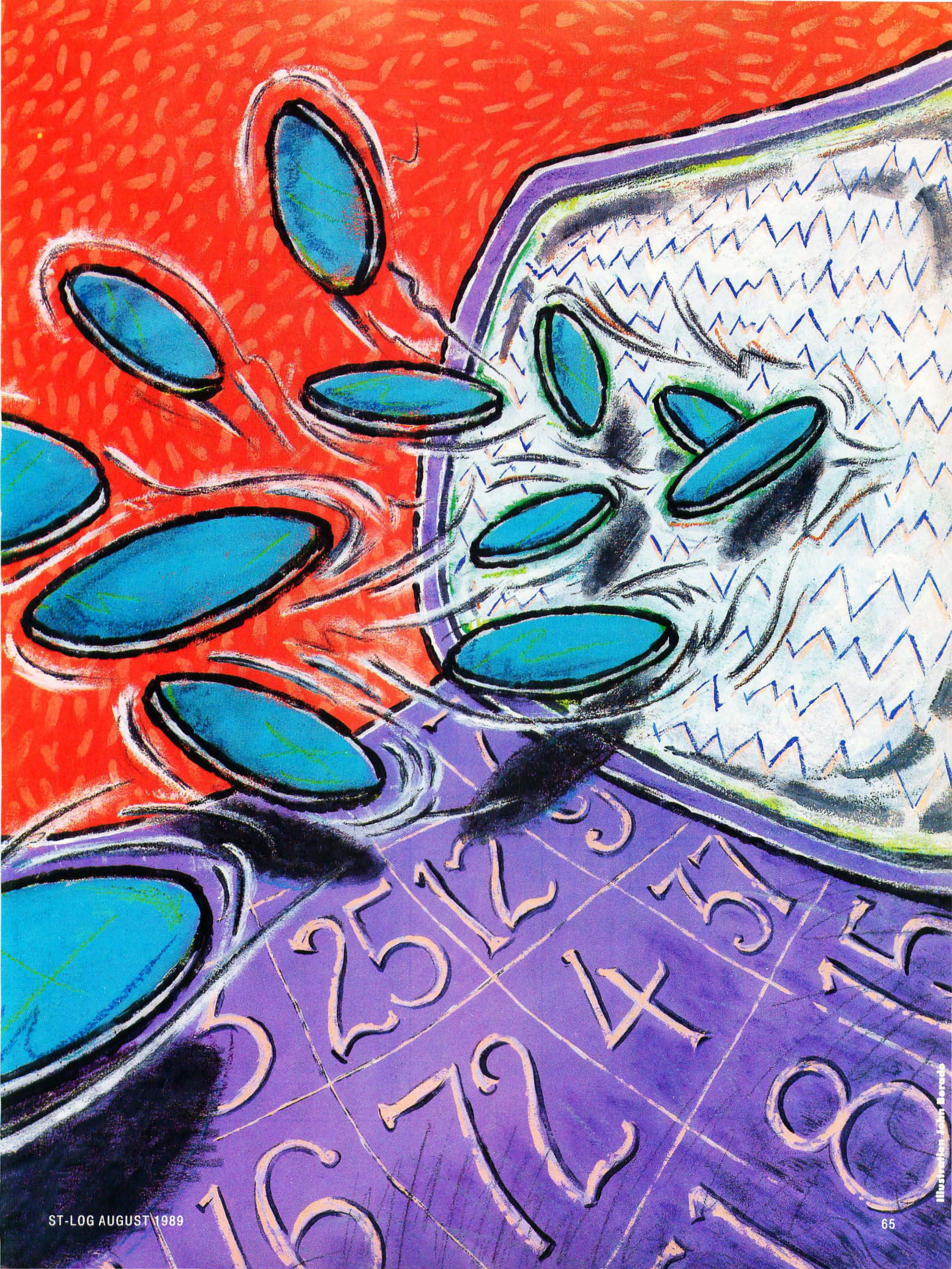
ST BINGO

BY ALBERT BAGGETTA

There is a game of chance that has fascinated people for over 100 years. In Italy it is known as Tumbule. In Latin America they call it Bolito. In Great Britain it is known as House. Americans know and love the game as Bingo. Every Saturday night masses of people gather in church halls all over the country with dreams of being the big card winner.

The problem with ordinary Bingo is that you have to play with a group of people in order to have any real fun. The element of competition is what makes the game interesting. Now, with *ST Bingo*, enthusiasts of the game can enjoy playing in the comfort of their own homes—even if there aren't any other players available. But if you happen to have a few friends around, they can play too.

For those not knowledgeable about the game, let me state the rules: Each player has a card or set of cards. (How many can you handle?) The cards have five rows and five columns of numbers, each column capped with a letter. Some of the number slots are marked FREE; these can substitute for any number. Letters and the numbers 1 to 75 are called out randomly, and when a player gets five horizontally, vertically or diagonally, he screams, "Bingo!" and is proclaimed the winner. And so it goes with *ST Bingo*, which can be played by any number of players—one or more players against the computer (it plays two cards), and each other; or for those who just like to watch, one computer card can be played against the other, on screen.



Running the program

To run the program, you will have to take care of a couple of details. Your master disk must have the program called CONTROL.ACC. This came free on your system disk when you bought the computer. Boot up your ST so that you can see the usual desktop. Move the mouse arrow to the top of the screen and click on Desk. On the pull-down menu click on Install Printer.

In the dialog box that is produced you will see a box for Pixels Per Inch. Click on 960 and then on OK, and the box will disappear. This will set up your printer for the proper placement of graphics. Now save the desktop by clicking on the Save Desktop selection of the Options drop-down menu. After you have done this, you will no longer have to change the printer default before running the main program.

In addition to CONTROL.ACC, your master disk must contain the following programs: BINGO.PRG and WINNER.FIL. The first is the compiled GFA BASIC program, and the latter is a file that will hold the name of the most recent winner.

If you don't have this month's disk version, you can create a copy of *ST Bingo* by typing in Listing 1 using GFA BASIC 2.0. You will also have to create the file WINNER.FIL. Do this with any word processor that can save ASCII files. Simply type in the string "COMPUTER", including the quotation marks, and save the file as WINNER.FIL.

To get started, click on BINGO.PRG. The main menu will now pop on the screen. Using the mouse to click on the circle, choose one of the following: Play Bingo, Make Cards or Quit.

Making cards

Before we play Bingo, let's click on the circle next to Make Cards, so that we can print some cards to play with. Be warned that in order to print the cards, you must have done the printer configuration explained above. Also, you must have an Epson or Epson-compatible printer (I use the Star SG-10) attached to your computer and powered up, ready to go.

You should now see an Alert Box offering three options:

a) *Randomly*—the computer will design the cards, randomly placing the numbers in the columns and rows (with up to three free boxes).

b) *Personal*—You can design your own cards, placing the numbers and free boxes where you want.

c) *Quit*—Return to the main menu.

Click on Randomly first. The computer will instantly draw a card that fills the majority of the screen and plunk down a combination of numbers and free boxes.

Take a good look at the card. If you do not like the way it came out, use the mouse to click on the letter N on the far right of the screen. The card will quickly redraw with a new set of numbers. You may do this as many times as you like until you're satisfied. If you are happy with the card (check to see that your printer is turned on and the paper at the top of the page), click on the letter Y at the far right of the screen. The prompt area will clear and the screen will dump a handsome Bingo card to your printer. The process takes a little over two minutes per card.

When the printing is completed the screen will clear, a new card will be screened, and you will be given the option to change the card, print the card or quit. If you choose to quit, you will be returned to the main-menu screen.

Click on Quit then click on Make Cards again. Let's try designing our own card now. Click on Personal. You should now see a screen reminiscent of the other make screen, except the card does not have any numbers.

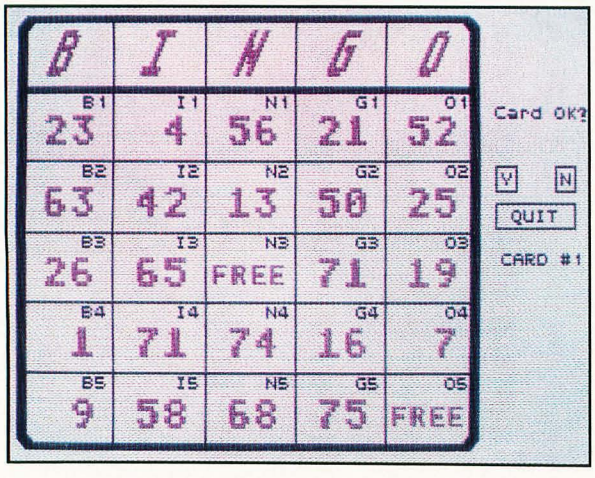
You can enter any number from 1 to 75 in each box. Point to one of the boxes and press the left mouse button to increase the numbers. Press the right button to decrease the numbers. Notice that between 1 and 75 you are offered a free box (FR). Go ahead, and fill in all of the boxes.

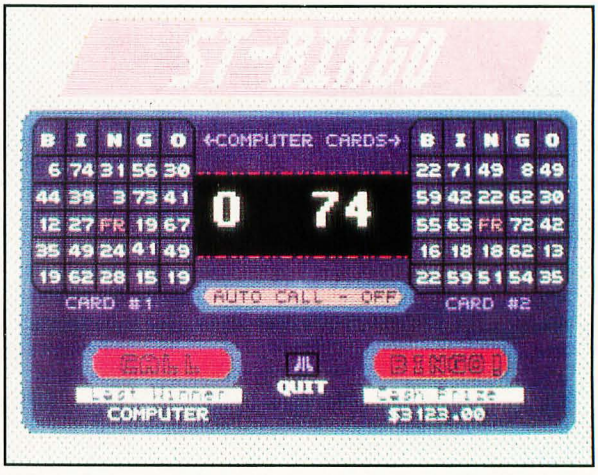
All done? Okay, set up your printer again (actually, you are probably already at the top of the page after printing the last card), then click on the box that reads Done, and your personal card will be sent to the printer. Of course you may quit at any time without printing by clicking on QUIT.

That's all there is to making cards. You will find that if you make a few cards before each game session, you will soon have quite a collection to use with friends later on.

Playing Bingo

Let's take our cards and go back to the main menu. (Click on QUIT.) To get to the game, click on Play Bingo. Your screen will clear and then fill with lots of graphic goodies. On either side of the screen, under the title, you will see the two computer cards. These will be played by the ST. If the computer gets a number, the corresponding square will fill with a dark gold color.





In the upper center of the screen is a black window that will display the current letter and number being called. At the start, the window will read Let's Play Bingo! At the bottom of the play screen, you will see two large orange and green bars. The one marked Call is pressed when you want to see the next letter/number combination.

If by some chance any of the players (except the computer) has made Bingo on a card, press the other orange bar, marked Bingo! This will allow the winner to verify his winning card. More on this later.

Underneath the black window, there is a gold and green bar marked Auto Call-On/Off (the "/" indicates a toggle-type switch). If you press this bar, the computer will take over, calling the numbers automatically, allowing a reasonable amount of time for the players to study their cards. You may shut off the auto calling by pointing to the bar during a call and holding down the left mouse button, until the indicator reads Off.

Take a look at the very bottom of the screen, left and right. See the two white message bars? One reads Last Winner, and the other reads Cash Prize. Under these banners you will find the name of the last game champion and the prize money for the current game, respectively. (I've worked on the game for quite a while, and I'm still not sure who is going to pay the prize money.)

Finally, if for some reason you have to leave the game, just click on the Quit icon, the little Atari symbol in the bottom center of the screen. This will bring you back to the main menu, where you can quit properly.

Verifying a winner

Okay, let's see where that Bingo bar brings you. Do not press this unless you really have a winning card. First, you will go to a black verification screen, where you will then be asked your name (eight characters only). Press Return when you are done. You may use your real name, an abbreviation or some code word.

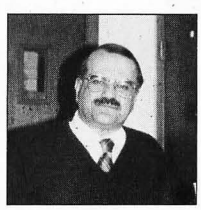
Now you will be given five prompts for your winning entries, one at a time. For each, you must enter a letter and a number. For example, you might enter B29, G72, I15. . .etc. You will only be allowed three characters for each entry. Press Return after each entry. If you have a free block enter FRE instead of a letter/number combination. You must enter five legitimate combinations to leave this area. The computer will also check with you to be sure you have entered the correct combinations. If not, you may reenter all of the information.

When all five of your numbers have been accepted, you will receive either some bad news or some good news. The bad news is that your card is no good (don't try to pull a fast one on old ST). In this case, the game will be resumed. The good news is that you are a winner. The screen will light up with your name on a star field, announcing you as the winner of whatever sum of money is displayed. (Sure wish I knew who was going to pay all this cash.)

After a few fleeting moments of glory, your name is recorded as the latest winner (remember this will be displayed in subsequent games until you are unseated), and you will be returned to the main menu to play another game or quit.

If the ST gets a Bingo, this verification is not necessary. An immodest flashing Bingo sign will appear in the black window, after which the computer will reveal its winning card. After the computer's name is registered as victor, the main menu will appear to offer a new game or to allow you to quit.

Bingo is a fun game, and *ST Bingo* makes it even more fun, since you won't have to fight the mobs in local church halls. Now if I can only figure out how to make the computer pay out the prize money, it will be the greatest game ever. ■

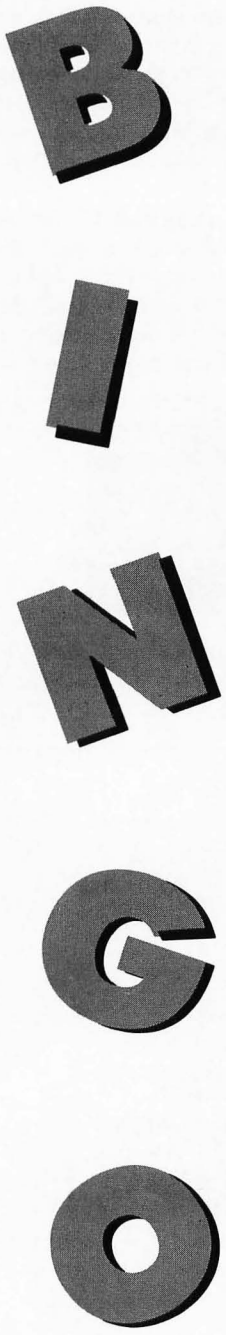


Albert Baggetta *is an English teacher and a professional guitarist. He frequently can be found wandering the* ST-LOG *SIG on DELPHI.*

LOW RESOLUTION

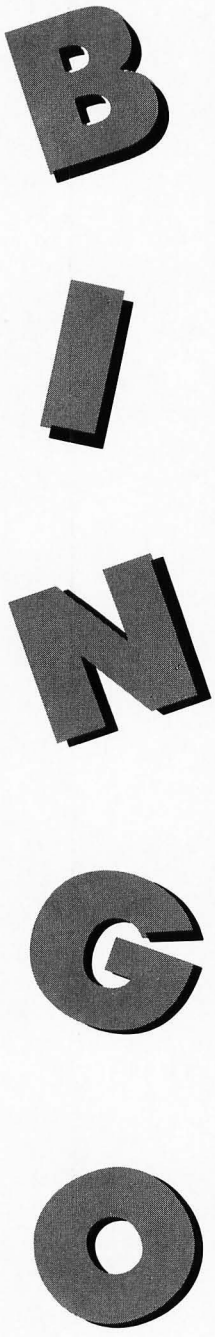
ST BINGO Listing 1: GFA BASIC 2.0

```
'IMPORTANT NOTE: THE "~" CHARACTER AT THE END OF SOME OF THE LINES
'  IN THIS LISTING INDICATES THAT THE PROGRAM LINE WRAPS AROUND TO THE
'  NEXT LINE OF THE LISTING.  THE TWO LINES SHOULD BE TYPED AS ONE,
'  WITHOUT THE "~".
'
' ********************************
' *         ST - BINGO           *
' *     ATARI ST- COMPUTERS      *
' *        A.BAGGETTA            *
' *       GFA BASIC V.2.0        *
' *  COPYRIGHT 1989 BY ST-LOG    *
' ********************************
Dim L$(5),N$(100),Sv$(1000),Comptr1%(5,5),Comptr2%(5,5),Ct%(25)
Dim B(5),I(5),N(5),G(5),O(5),Bb(5),Ii(5),Nn(5),Gg(5),Oo(5),C(5)
Dim Oldcolri(15),Oldcolrr(15),Oldcolrg(15),Oldcolrb(15)
Dim Msa(50),Msb(50),Msc(50),Msd(50),Mse(50),Pm(50),Pcm(50),Pcmr(50)
@Colr_get                                   !Save the colors
Cls
If Exist("Winner.fil") Then                 !Check for Winner File
  Print "Loading Winner file..."            !If ok load the file.  If not
  Open "I",#1,"Winner.fil"                  !Start over again
  Input #1,Win$
  Close #1
Else
  Alert 3," | |Missing Winner File",1,"END",Ab
  Goto Finish
Endif
Cls
Start_prg:
Cls                                         !Clear the screen and set the
Restore Clrz                                !Screen colors
For Clrz%=1 To 9
  Read C11,C12,C13,C14
  Setcolor C11,C12,C13,C14
Next Clrz%
Clrz:
Data 0,0,0,0,1,7,7,7,2,2,3,2,3,5,3,1,4,7,2,3,6,4,5,3,7,2,4,3,9,5,4,3,15,7,7,7
Rna=1000                                    !Select a random amount for prize
Rnb=5000                                    !Money
Cpz=Int(Rnd*(Rnb-Rna+1))+Rna
X%=0
```



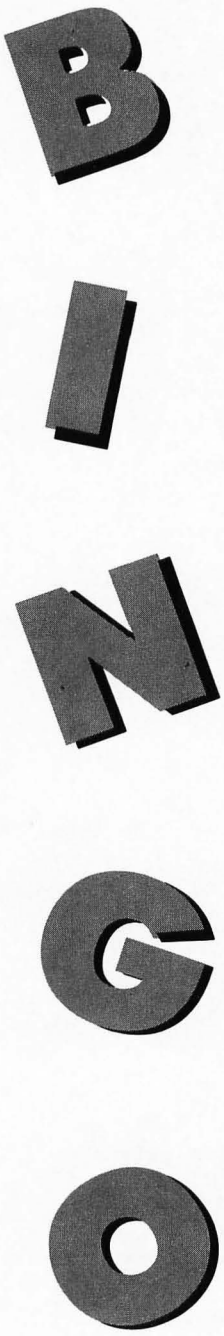

```
Deffill 2,2,7                                   !Set up the menu screen
Pbox 8,10,296,199
Graphmode 4
Deftext 4,4,0,32
Text 38,44,"   ST-BINGO   "
Graphmode 1
Deffill 3,2,8
Prbox 20,55,285,185
Defline 6,2,0,0
Prbox 20,55,285,185
Rbox 20,55,285,185
Graphmode 2
Deftext 5,5,0,9
Text 50,80,"Your Selection?"
Deftext 6,5,0,8
Text 100,100,"Play Bingo"
Text 100,120,"Make Cards"
Text 100,140,"Quit"
D$=Chr$(14)+Chr$(15)
Dup$=String$(9,D$)
Deftext 1,5,0,8
Text 50,160,Dup$
Color 6
Circle 85,98,5
Circle 85,118,5
Circle 85,138,5
Showm
Do                                              !Mouse control for the Menu Screen
  Mouse Mc,Mr,Mb
  If (Mc)80 And Mc(90) And (Mr)95 And Mr(101) And Mb=1
    Graphmode 1
    Deffill 3,2,8
    Prbox 20,55,285,185
    @Play_bingo
    Goto Start_prg
  Endif
  If (Mc)80 And Mc(90) And (Mr)115 And Mr(123) And Mb=1
    Print At(10,5);"make cards"
    Cls
    Goto Make_cards
    Goto Start_prg
  Endif
  If (Mc)80 And Mc(90) And (Mr)135 And Mr(141) And Mb=1
    Cls
    Goto Finish
  Endif
Loop
'
Procedure Play_bingo
  Leave=0 !LEAVE GAME FLAG
  Cd=0 !CARD # FLAG
  @Array_set                                   !Clear the arrays
  '                                             Set up the main screen to play BINGO
  Deffill 0,2,8
  Graphmode 2
  Text 85,100,"                    "
  Line 100,79,204,79
  Line 100,113,204,113
  For Grow=100 To 204 Step 2
    Pbox 100,80,Grow,112
  Next Grow
  Deftext 1,4,0,6
  Text 110,93,"Let's Play"
  Text 125,103,"BINGO!"
  Deffill 10,2,8
  Prbox 102,125,204,135
  Deffill 6,2,8
  Prbox 50,150,120,165
  Prbox 185,150,255,165
  Deffill 11,2,8
  Defline 1,3,0,0
  Color 8
  Rbox 20,55,285,185
  Rbox 102,125,204,135
  Rbox 50,150,120,165
  Rbox 185,150,255,165
  Deftext 3,16,0,5
  Graphmode 2
  Text 67,160,"CALL"
  Text 197,160,"BINGO!"
  Deftext 3,0,0,4
  Text 110,132,"AUTO CALL - OFF"
  Deftext 1,0,0,4
  Defline 1,1,0,0
  Color 0
  '
  Restore Small_crds                            !Draw the computer's cards
  For Smx%=1 To 20
    Read Sma,Smb,Smc,Smd
    If Smx%=1 Or Smx%=11 Then
      Rbox Sma,Smb,Smc,Smd
    Else
      Line Sma,Smb,Smc,Smd
    Endif
  Next Smx%
  Graphmode 2
  Deftext 1,1,0,4
  Text 28,68,"B"
  Text 44,68,"I"
  Text 60,68,"N"
  Text 74,68,"G"
  Text 90,68,"O"
  Deftext 1,0,0,4
  Restore Comp_crd1
  '
  ' Store the numbers on card 1 in an array comptr1%(xx,yy)
  '
```

```
Freebe=0
Xx=0
Yy=0
For Rp%=1 To 25
  Read C,R
  N=Int(Rnd*75)+1
  N$=Str$(N)
  If Len(N$)=1
    N$=" "+N$
  Endif
  If Rp%<>13 Then
    Text C,R,N$
  Else
    Deftext 10,0,0,4
    Text C,R,"FR"
    Deftext 1,0,0,4
  Endif
  If Rp%>0 And Rp%<6 Then
    Xx=1
  Endif
  If Rp%>5 And Rp%<11 Then
    Xx=2
  Endif
  If Rp%>10 And Rp%<16 Then
    Xx=3
  Endif
  If Rp%>15 And Rp%<21 Then
    Xx=4
  Endif
  If Rp%>20 And Rp%<26 Then
    Xx=5
  Endif
  If Yy=5 Then
    Yy=0
  Endif
  Yy=Yy+1
  If Rp%=13 Then
    N=-1
  Endif
  Comptr1%(Xx,Yy)=N
Next Rp%
'
Deftext 1,1,0,4
Text 211,68,"B"
Text 227,68,"I"
Text 242,68,"N"
Text 257,68,"G"
Text 272,68,"O"
Deftext 1,0,0,4
'
Restore Comp_crd2
'
' Store the numbers on card 2 in an array comptr2%(xx,yy)
'
Freebe=0
Xx=0
Yy=0
For Rp%=1 To 25
  Read C,R
  N=Int(Rnd*75)+1
  N$=Str$(N)
  If Len(N$)=1
    N$=" "+N$
  Endif
  If Rp%<>13 Then
    Text C,R,N$
  Else
    Deftext 10,0,0,4
    Text C,R,"FR"
    Deftext 1,0,0,4
  Endif
  If Rp%>0 And Rp%<6 Then
    Xx=1
  Endif
  If Rp%>5 And Rp%<11 Then
    Xx=2
  Endif
  If Rp%>10 And Rp%<16 Then
    Xx=3
  Endif
  If Rp%>15 And Rp%<21 Then
    Xx=4
  Endif
  If Rp%>20 And Rp%<26 Then
    Xx=5
  Endif
  If Yy=5 Then
    Yy=0
  Endif
  Yy=Yy+1
  If Rp%=13 Then
    N=-1
  Endif
  Comptr2%(Xx,Yy)=N
Next Rp%
Deftext 11,0,0,4
Text 105,68,Chr$(4)+"COMPUTER CARDS"+Chr$(3)
Deftext 11,0,0,4
Text 40,135,"CARD #1"
Text 223,135,"CARD #2"
Graphmode 2
Box 145,153,160,162
Text 148,159,Chr$(14)+Chr$(15)
Deftext 1,1,0,4
Text 141,169,"QUIT"
Deftext 1,0,0,4
Graphmode 4
```




```
Text 45,172," Last Winner "
Text 183,172," Cash Prize  "
Graphmode 2
Text 60,180,Win$
Text 195,180,"$"+Str$(Cpz)+".00"
Graphmode 1
Yup=0
T:
Do                                          !Game Mouse Control
  Showm
  Mouse Mc,Mr,Mb
  If (Mc)50 And Mc(120) And (Mr)150 And Mr(165) And Mb=1
    Goto Jout
  Endif
  If (Mc)185 And Mc(255) And (Mr)150 And Mr(165) And Mb=1
    Hidem
    Sget Screen$
    @Check_bingo
    If Yup Then
      Goto Sto
    Endif
  Endif
  If (Mc)100 And Mc(200) And (Mr)125 And Mr(135) And Mb=1
    Graphmode 2
    Color 8
    Deffill 10,2,8
    Prbox 102,125,204,135
    Defline 1,3,0,0
    Rbox 102,125,204,135
    Deftext 3,0,0,4
    Text 110,132,"AUTO CALL - ON"
    Graphmode 1
    Deftext 1,0,0,12
    Auto=1
    Goto Jout
  Endif
  If (Mc)145 And Mc(160) And (Mr)153 And Mr(162) And Mb=1
    Goto Sto
  Endif
Loop
Jout:
Showm
Deftext 1,0,0,12
Deffill 3,2,8
For Grow=204 To 100 Step -2          !Clear the old Bingo number
  Sound 1,15,Grow/8,4
  Pbox Grow,80,204,112
Next Grow
Sound 1,0
Deffill 0,2,8
For Grow=100 To 204 Step 2
  Sound 1,15,Grow/8,4
  Pbox 100,80,Grow,112
Next Grow
Sound 1,0
Try_again:
Tg=0                                        !Select Bingo number
Lter=Int(Rnd*5)+1
N=Int(Rnd*75)+1
For Cut%=0 To X%
  If Sv$(Cut%)=L$(Lter)+Str$(N) Then
    Tg=1
  Endif
Next Cut%
If Tg=1 Then
  Goto Try_again
Endif
Ck$=L$(Lter)
@Chk_comp_crd(Ck$)                          !Run a check on the cards
If Leave=1 Then
  Deftext 1,4,0,32
  If Cd=1 Then
    Get 23,57,100,128,Cd$ ! CUT CARD ONE FROM SCREEN
    Cls
    Text 70,30,"ST-BINGO"
    Setcolor 0,4,0,0
    Put 100,50,Cd$,3
    Print At(15,19);"CARD #1"
    Print At(12,21);"COMPUTER WINS"
    Win$="COMPUTER"
    Open "o",#1,"Winner.fil"
    Write #1,Win$
    Close #1
    Pause 400
    Goto Sto
```

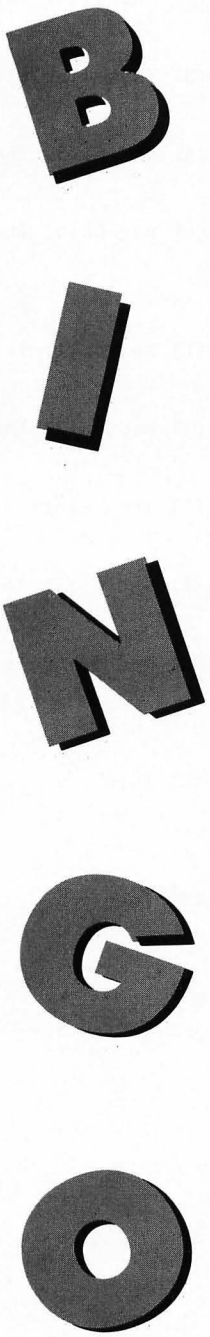
```
    Else
      If Cd=2 Then
        Get 205,57,282,128,Cd$
        Cls
        Text 70,30,"ST-BINGO"
        Setcolor 0,4,0,0
        Put 100,50,Cd$,3
        Print At(15,19);"CARD #2"
        Print At(12,21);"COMPUTER WINS"
        Win$="COMPUTER"
        Open "o",#1,"Winner.fil"
        Write #1,Win$
        Close #1
        Pause 400
        Goto Sto
      Endif
    Endif
  Endif
  Jmphere:
  Text 108,100,L$(Lter)+"  "+Str$(N)
  ' Save the letter/number combination for later check
  X%=X%+1
  Sv$(X%)=L$(Lter)+Str$(N)
  If Auto=1
    For Del=1 To 75000
      Showm
    Next Del
  Endif
  Repeat
    Mouse Mc,Mr,Mb
    If Mb=2 Then
      Inc Visit
      If Visit=1
        Print At(17,14);"PAUSE"
      Endif
    Endif
  Until Mb<>2
  If Mb<>2
    Print At(17,14);"         "
    Visit=0
  Endif
  If (Mc>100 And Mc<200) And (Mr>125 And Mr<135) And Mb=1
    Auto=0
    Graphmode 2
    Color 8
    Deffill 10,2,8
    Prbox 102,125,200,135
    Defline 1,3,0,0
    Rbox 102,125,200,135
    Deftext 3,0,0,4
    Text 110,132,"AUTO CALL - OFF"
    Graphmode 1
    Deftext 1,0,0,12
    Pause 50
  Endif
  If Auto=0 Then
    Goto T
  Else
    Goto Jout
  Endif
  Sto:
Return
'
Make_cards:
Count=0
Setcolor 0,7,7,7                          ! Set new screen colors
Setcolor 1,5,2,3
Setcolor 15,0,0,0
Deftext 1,0,0,15
Text 70,50,"Bingo Card Designer"          ! Set up Alert String
Alert$="Before Printing Bingo Card        |be sure your printer is set  |~
to 960 dots per line.  "
Alert 0,Alert$,0,"Randomly|Personal|Quit",Choose
If Choose=1 Then                          ! Alert Box choices
  Goto Make_one
Endif
If Choose=2 Then
  Goto Make_two
Endif
If Choose=3 Then
  Goto Start_prg
Endif
Make_one:                                 ! Design the first type of card
Cls
Tcrd:
```



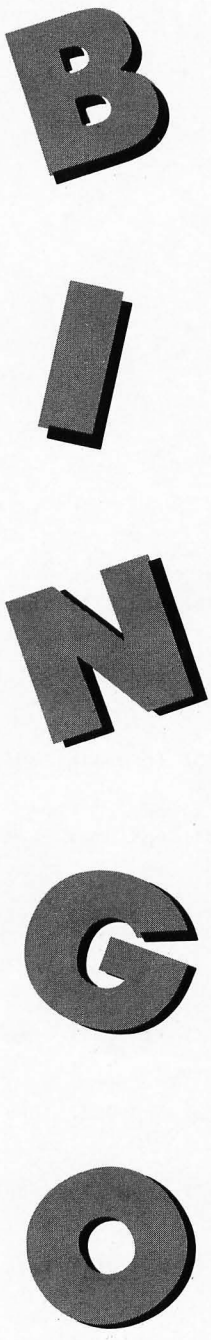

```
Cls
Color 1
Defline 0,3,0,0
Rbox 30,10,250,185
Defline 0,2,0,0
Line 74,10,74,185
Line 118,10,118,185
Line 162,10,162,185
Line 206,10,206,185
Defline 0,3,0,0
Line 30,39,250,39
Defline 0,2,0,0
Line 30,68,250,68
Line 30,97,250,97
Line 30,126,250,126
Line 30,155,250,155
' fill in the labels on the large card
Deftext 2,4,0,20
Restore Big_crds_lbls
For Bigx%=1 To 30
  Read Biga,Bigb,Bigc$
  If Bigx%=6 Then
    Deftext 1,0,0,4
  Endif
  Text Biga,Bigb,Bigc$
Next Bigx%
Deftext 2,0,0,10
Restore Cdata
Freebe=0
For Rp=1 To 25                          ! Put numbers on the card
  Read C,R
  N=Int(Rnd*75)+1
  N$=Str$(N)
  If Len(N$)=1
    N$=" "+N$
  Endif
  If Freebe<2 Then
    Free=Int(Rnd*15)+1
    If Free=9
      Inc Freebe
      Deftext 2,0,0,7
      Text C-7,R,"FREE"
      Deftext 2,0,0,10
      Goto Jmp
    Endif
  Endif
  If Rp<>13 Then
    Text C,R,N$
  Else
    Deftext 2,0,0,7
    Text C-7,R,"FREE"
    Deftext 2,0,0,10
  Endif
  Jmp:
Next Rp
Deftext 1,0,0,4                         ! Check to see if card is OK
Text 260,50,"Card OK?"
Box 260,70,270,80
Box 290,70,300,80
Box 260,85,300,95
Text 263,77,"Y"
Text 293,77,"N"
Text 268,92,"QUIT"
Inc Count
Text 263,110,"CARD #"+Str$(Count)
Do
  Mouse Mc,Mr,Mb
  If (Mc>260 And Mc<270) And (Mr>70 And Mr<80) And Mb=1
    Hidem
    Deffill 0,2,8
    Pbox 260,40,370,120
    Hardcopy
    Showm
    For Mulcop=1 To 35
      Lprint
    Next Mulcop
    Goto Tcrd
  Endif
  If (Mc>290 And Mc<300) And (Mr>70 And Mr<80) And Mb=1
    Dec Count
    Goto Tcrd
  Endif
  If (Mc>260 And Mc<300) And (Mr>85 And Mr<95) And Mb=1
    Goto Bak_menu
  Endif
```

```
Loop
Bak_menu:
Goto Start_prg
Make_two:                                   ! Make the second kind of card
'
'
Graphmode 1
Setcolor 0,7,7,7
Setcolor 1,5,2,3
Setcolor 15,0,0,0
Deftext 1,0,0,15
Cls
Color 1
Defline 0,3,0,0
Rbox 30,10,250,185
Defline 0,2,0,0
Line 74,10,74,185
Line 118,10,118,185
Line 162,10,162,185
Line 206,10,206,185
Defline 0,3,0,0
Line 30,39,250,39
Defline 0,2,0,0
Line 30,68,250,68
Line 30,97,250,97
Line 30,126,250,126
Line 30,155,250,155
' fill in the labels on the large card
Deftext 2,4,0,20
Restore Big_crds_lbls
For Bigx%=1 To 30
  Read Biga,Bigb,Bigc$
  If Bigx%=6 Then
    Deftext 1,0,0,4
  Endif
  Text Biga,Bigb,Bigc$
Next Bigx%
Deftext 2,0,0,4
Defline 1,1,0,0
Box 260,85,300,95
Box 260,105,300,115
Text 268,92,"DONE"
Text 268,112,"QUIT"
Deftext 2,0,0,10
Restore Mous_crd_dat
For Msrch%=1 To 50
  Read A,B,C,D,E,F,G,H
  Msa(Msrch%)=A
  Msb(Msrch%)=B
  Msc(Msrch%)=C
  Msd(Msrch%)=D
  Mse(Msrch%)=E
  Pm(Msrch%)=F
  Pcm(Msrch%)=G
  Pcmr(Msrch%)=H
Next Msrch%
'
Do
  Mouse A,B,C
  For Msrch%=1 To 50
    If (A)Msa(Msrch%) And A(Msb(Msrch%)) And (B)Msc(Msrch%) And B(Msd(~
Msrch%)) And C=1 Then
      P=Pm(Msrch%)
      Pc=Pcm(Msrch%)
      Pr=Pcmr(Msrch%)
      @Make_ver_crdup
      C=0
    Endif
    If (A)Msa(Msrch%) And A(Msb(Msrch%)) And (B)Msc(Msrch%) And B(Msd(~
Msrch%)) And C=2 Then
      P=Pm(Msrch%)
      Pc=Pcm(Msrch%)
      Pr=Pcmr(Msrch%)
      @Make_ver_crdown
      C=0
    Endif
  Next Msrch%
  If (A)260 And A(300) And (B)85 And B(95) And C=1
    Hidem
    Deffill 0,2,8
    Pbox 260,85,300,140
    Hardcopy
    Showm
    Goto Jloop
  Endif
  If (A)260 And A(300) And (B)105 And B(115) And C=1
```

```
    Goto Jloop
  Endif
Loop
Jloop:
Goto Start_prg
'
' SET ARRAYS
'
Procedure Array_set
  L$(1)="B"
  L$(2)="I"
  L$(3)="N"
  L$(4)="G"
  L$(5)="O"
  Arrayfill B(),0
  Arrayfill I(),0
  Arrayfill N(),0
  Arrayfill G(),0
  Arrayfill O(),0
  Arrayfill Bb(),0
  Arrayfill Ii(),0
  Arrayfill Nn(),0
  Arrayfill Gg(),0
  Arrayfill Oo(),0
  Arrayfill C(),0
  Cd=0
Return
'
Procedure Chk_comp_crd(Ck$)
  Deffill 10,2,8
  If Ck$="B" Then
    If Comptr1%(1,1)=N Then
      Fill 25,79
      B(1)=1
      @Snd
    Endif
    If Comptr1%(1,2)=N Then
      Fill 25,90
      B(2)=1
      @Snd
    Endif
    If Comptr1%(1,3)=N Then
      Fill 25,101
      B(3)=1
      @Snd
    Endif
    If Comptr1%(1,4)=N Then
      Fill 25,112
      B(4)=1
      @Snd
    Endif
    If Comptr1%(1,5)=N Then
      Fill 25,123
      B(5)=1
      @Snd
    Endif
    If Comptr2%(1,1)=N Then
      Fill 208,79
      Bb(1)=1
      @Snd
    Endif
    If Comptr2%(1,2)=N Then
      Fill 208,90
      Bb(2)=1
      @Snd
    Endif
    If Comptr2%(1,3)=N Then
      Fill 208,101
      Bb(3)=1
      @Snd
    Endif
    If Comptr2%(1,4)=N Then
      Fill 208,112
      Bb(4)=1
      @Snd
    Endif
    If Comptr2%(1,5)=N Then
      Fill 208,123
      Bb(5)=1
      @Snd
    Endif
  Endif
  If Ck$="I" Then
    If Comptr1%(2,1)=N Then
      Fill 41,79
      I(1)=1
      @Snd
    Endif
    If Comptr1%(2,2)=N Then
      Fill 41,90
      I(2)=1
      @Snd
    Endif
    If Comptr1%(2,3)=N Then
      Fill 41,101
      I(3)=1
      @Snd
    Endif
    If Comptr1%(2,4)=N Then
      Fill 41,112
      I(4)=1
      @Snd
    Endif
    If Comptr1%(2,5)=N Then
      Fill 41,123
      I(5)=1
      @Snd
    Endif
    If Comptr2%(2,1)=N Then
      Fill 223,79
      Ii(1)=1
      @Snd
    Endif
    If Comptr2%(2,2)=N Then
      Fill 223,90
      Ii(2)=1
      @Snd
    Endif
    If Comptr2%(2,3)=N Then
      Fill 223,101
      Ii(3)=1
      @Snd
    Endif
    If Comptr2%(2,4)=N Then
      Fill 223,112
      Ii(4)=1
      @Snd
    Endif
    If Comptr2%(2,5)=N Then
      Fill 223,123
      Ii(5)=1
      @Snd
    Endif
  Endif
  If Ck$="N" Then
    If Comptr1%(3,1)=N Then
      Fill 56,79
      N(1)=1
      @Snd
    Endif
    If Comptr1%(3,2)=N Then
      Fill 56,90
      N(2)=1
      @Snd
    Endif
    If Comptr1%(3,3)=N Then
      Fill 56,101
      N(3)=1
      @Snd
    Endif
    If Comptr1%(3,4)=N Then
      Fill 56,112
      N(4)=1
      @Snd
    Endif
    If Comptr1%(3,5)=N Then
      Fill 56,123
      N(5)=1
      @Snd
    Endif
    If Comptr2%(3,1)=N Then
      Fill 238,79
      Nn(1)=1
      @Snd
    Endif
    If Comptr2%(3,2)=N Then
      Fill 238,90
      Nn(2)=1
      @Snd
    Endif
    If Comptr2%(3,3)=N Then
      Fill 238,101
      Nn(3)=1
      @Snd
    Endif
    If Comptr2%(3,4)=N Then
      Fill 238,112
      Nn(4)=1
      @Snd
    Endif
    If Comptr2%(3,5)=N Then
      Fill 238,123
      Nn(5)=1
      @Snd
    Endif
  Endif
  If Ck$="G" Then
    If Comptr1%(4,1)=N Then
      Fill 71,79
      G(1)=1
      @Snd
    Endif
    If Comptr1%(4,2)=N Then
      Fill 71,90
      G(2)=1
      @Snd
    Endif
    If Comptr1%(4,3)=N Then
      Fill 71,101
      G(3)=1
      @Snd
    Endif
    If Comptr1%(4,4)=N Then
      Fill 71,112
      G(4)=1
      @Snd
    Endif
    If Comptr1%(4,5)=N Then
      Fill 71,123
      G(5)=1
      @Snd
    Endif
    If Comptr2%(4,1)=N Then
      Fill 253,79
      Gg(1)=1
      @Snd
    Endif
    If Comptr2%(4,2)=N Then
      Fill 253,90
      Gg(2)=1
      @Snd
    Endif
    If Comptr2%(4,3)=N Then
      Fill 253,101
      Gg(3)=1
      @Snd
    Endif
    If Comptr2%(4,4)=N Then
      Fill 253,112
      Gg(4)=1
      @Snd
    Endif
    If Comptr2%(4,5)=N Then
      Fill 253,123
      Gg(5)=1
      @Snd
    Endif
  Endif
  If Ck$="O" Then
    If Comptr1%(5,1)=N Then
      Fill 86,79
      O(1)=1
      @Snd
    Endif
    If Comptr1%(5,2)=N Then
      Fill 86,90
      O(2)=1
      @Snd
    Endif
    If Comptr1%(5,3)=N Then
      Fill 86,101
      O(3)=1
      @Snd
    Endif
```

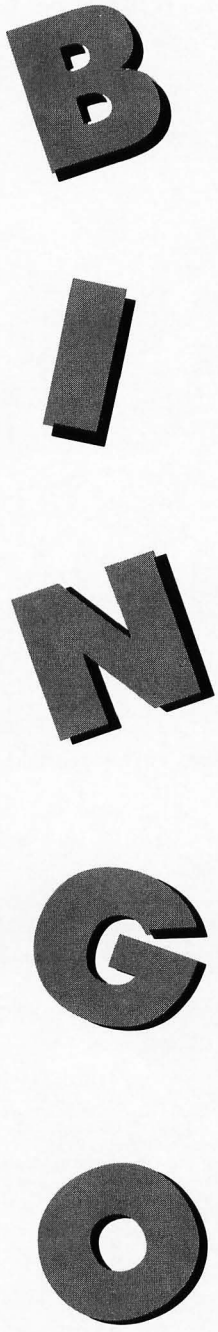



```
      If Comptr1%(5,4)=N Then
        Fill 86,112
        O(4)=1
        @Snd
      Endif
      If Comptr1%(5,5)=N Then
        Fill 86,123
        O(5)=1
        @Snd
      Endif
      If Comptr2%(5,1)=N Then
        Fill 268,79
        Oo(1)=1
        @Snd
      Endif
      If Comptr2%(5,2)=N Then
        Fill 268,90
        Oo(2)=1
        @Snd
      Endif
      If Comptr2%(5,3)=N Then
        Fill 268,101
        Oo(3)=1
        @Snd
      Endif
      If Comptr2%(5,4)=N Then
        Fill 268,112
        Oo(4)=1
        @Snd
      Endif
      If Comptr2%(5,5)=N Then
        Fill 268,123
        Oo(5)=1
        @Snd
      Endif
    Endif
    @Full_five
  Return
  '
  Procedure Full_five
    ' Vertical Columns Card #1
    '
    If B(1) And B(2) And B(3) And B(4) And B(5) Then
      Cd=1
      @Comp_got_bingo
    Endif
    If I(1) And I(2) And I(3) And I(4) And I(5) Then
      Cd=1
      @Comp_got_bingo
    Endif
    If N(1) And N(2) And N(4) And N(5) Then !Includes FREE
      Cd=1
      @Comp_got_bingo
    Endif
    If G(1) And G(2) And G(3) And G(4) And G(5) Then
      Cd=1
      @Comp_got_bingo
    Endif
    If O(1) And O(2) And O(3) And O(4) And O(5) Then
      Cd=1
      @Comp_got_bingo
    Endif
    ' Horizontal Columns Card #1
    '
    If B(1) And I(1) And N(1) And G(1) And O(1) Then
      Cd=1
      @Comp_got_bingo
    Endif
    If B(2) And I(2) And N(2) And G(2) And O(2) Then
      Cd=1
      @Comp_got_bingo
    Endif
    If B(3) And I(3) And G(3) And O(3) Then !Includes FREE
      Cd=1
      @Comp_got_bingo
    Endif
    If B(4) And I(4) And N(4) And G(4) And O(4) Then
      Cd=1
      @Comp_got_bingo
    Endif
    If B(5) And I(5) And N(5) And G(5) And O(5) Then
      Cd=1
      @Comp_got_bingo
    Endif
    ' Diagonal Column on B1 -- Card #1
    '
    If B(1) And I(2) And G(4) And O(5) Then
      Cd=1
      @Comp_got_bingo
    Endif
    ' Diagonal Column on O1 -- Card #1
    '
```




```
  If O(1) And G(2) And I(4) And B(5) Then
    Cd=1
    @Comp_got_bingo
  Endif
  ' Vertical Columns Card #2
  '
  If Bb(1) And Bb(2) And Bb(3) And Bb(4) And Bb(5) Then
    Cd=2
    @Comp_got_bingo
  Endif
  If Ii(1) And Ii(2) And Ii(3) And Ii(4) And Ii(5) Then
    Cd=2
    @Comp_got_bingo
  Endif
  If Nn(1) And Nn(2) And Nn(4) And Nn(5) Then !Includes FREE
    Cd=2
    @Comp_got_bingo
  Endif
  If Gg(1) And Gg(2) And Gg(3) And Gg(4) And Gg(5) Then
    Cd=2
    @Comp_got_bingo
  Endif
  If Oo(1) And Oo(2) And Oo(3) And Oo(4) And Oo(5) Then
    Cd=2
    @Comp_got_bingo
  Endif
  ' Horizontal Columns Card #2
  '
  If Bb(1) And Ii(1) And Nn(1) And Gg(1) And Oo(1) Then
    Cd=2
    @Comp_got_bingo
  Endif
  If Bb(2) And Ii(2) And Nn(2) And Gg(2) And Oo(2) Then
    Cd=2
    @Comp_got_bingo
  Endif
  If Bb(3) And Ii(3) And Gg(3) And Oo(3) Then !Includes FREE
    Cd=2
    @Comp_got_bingo
  Endif
  If Bb(4) And Ii(4) And Nn(4) And Gg(4) And Oo(4) Then
    Cd=2
    @Comp_got_bingo
  Endif
  If Bb(5) And Ii(5) And Nn(5) And Gg(5) And Oo(5) Then
    Cd=2
    @Comp_got_bingo
  Endif
  ' Diagonal Column on B1 -- Card #2
  '
  If Bb(1) And Ii(2) And Gg(4) And Oo(5) Then
    Cd=2
    @Comp_got_bingo
  Endif
  ' Diagonal Column on O1 -- Card #2
  '
  If Oo(1) And Gg(2) And Ii(4) And Bb(5) Then
    Cd=2
    @Comp_got_bingo
  Endif
Return
'
Procedure Snd
  Sound 1,15,1,5,1
  Sound 1,0
  Sound 1,15,7,7,1
  Sound 1,0
Return
'
Procedure Comp_got_bingo
  Deftext 1,4,0,6
  Deffill 0,2,8
  For Zr=1 To 10
    Sound 1,15,1,5,1
    Prbox 100,80,204,112
    Text 120,100,"!BINGO!"
    Pause 1
    Sound 1,15,7,7,1
    Next Zr
    Sound 1,0
    Leave=1
  Return
  '
  Procedure Check_bingo
    Cls
    Graphmode 4
    Deftext 4,0,0,6
    Text 5,7,"          VERIFICATION SCREEN          "
    Graphmode 1
    Print At(2,3);"Name please: ";
    Form Input 8,Nm$
    Print At(2,5);"Enter your winning combination"
```

```
    Print At(2,6);"one part at a time. Press RETURN"
    Print At(2,7);"after each entry."
    Ck_1:
    Au=0
    Print At(2,9);"                                        "
    Print At(2,9);
    Print "First entry: ";
    Form Input 3,Fir$
    If Fir$<>"FRE" Then
      Sw$=Fir$
      @Check_ending
      If Au Then
        Goto Ck_1
      Endif
    Endif
    Ck_2:
    Au=0
    Print At(2,11);"                                       "
    Print At(2,11);
    Print "Second entry: ";
    Form Input 3,Sec$
    If Sec$<>"FRE" Then
      Sw$=Sec$
      @Check_ending
      If Au Then
        Goto Ck_2
      Endif
    Endif
    Ck_3:
    Au=0
    Print At(2,13);"                                       "
    Print At(2,13);
    Print "Third entry: ";
    Form Input 3,Thir$
    If Thir$<>"FRE" Then
      Sw$=Thir$
      @Check_ending
      If Au Then
        Goto Ck_3
      Endif
    Endif
    Ck_4:
    Au=0
    Print At(2,15);"                                       "
    Print At(2,15);
    Print "Fourth entry: ";
    Form Input 3,Fou$
    If Fou$<>"FRE" Then
      Sw$=Fou$
      @Check_ending
      If Au Then
        Goto Ck_4
      Endif
    Endif
    Ck_5:
    Au=0
    Print At(2,17);"                                       "
    Print At(2,17);
    Print "Fifth entry: ";
    Form Input 3,Fif$
    If Fif$<>"FRE" Then
      Sw$=Fif$
      @Check_ending
      If Au Then
        Goto Ck_5
      Endif
    Endif
    Print
    Print At(2,19);"Satisfied? Y/N  ";
    Form Input 1,Sati$
    If Sati$="N" Or Sati$="n" Then
    @Check_bingo
  Endif
  For J%=1 To X%
    If Fir$="FRE" Or Sv$(J%)=Fir$
      C(1)=1
    Endif
    If Sec$="FRE" Or Sv$(J%)=Sec$
      C(2)=1
    Endif
    If Thir$="FRE" Or Sv$(J%)=Thir$
      C(3)=1
    Endif
    If Fou$="FRE" Or Sv$(J%)=Fou$
      C(4)=1
    Endif
    If Fif$="FRE" Or Sv$(J%)=Fif$
      C(5)=1
    Endif
  Next J%
  If C(1)=1 And C(2)=1 And C(3)=1 And C(4)=1 And C(5)=1
    @Winner
    Yup=1
    For J%=1 To X%
      Sv$(J%)=""
    Next J%
    Win$=Nm$
    Open "o",#1,"Winner.fil"
    Write #1,Win$
    Close #1
    Pause 300
  Else
    Print
    Print "    SORRY! CARD NO GOOD"
    Pause 300
    Cls
    Sput Screen$
  Endif
Return
'
Procedure Make_ver_crdup
  Inc Ct%(P)
  If Ct%(P)=76 Then
    Text Pc,Pr,"FR"
    Goto Pv
  Endif
  If Ct%(P)=77 Then
    Ct%(P)=1
    Text Pc,Pr,"  "
  Endif
  Text Pc,Pr,Str$(Ct%(P))
  Pv:
  Pause 5
Return
'
Procedure Make_ver_crdown
  Dec Ct%(P)
  If Ct%(P)=9 Then
    Text Pc,Pr,"  "
  Endif
  If Ct%(P)=0 Then
    Text Pc,Pr,"FR"
    Goto Pvv
  Endif
  If Ct%(P)=-1 Then
    Text Pc,Pr,"  "
    Ct%(P)=75
  Endif
  Text Pc,Pr,Str$(Ct%(P))
  Pvv:
  Pause 5
Return
'
Procedure Check_ending
  If Sw$="" Then
    Print At(2,20);"You must enter something."
    Pause 100
    Print At(2,20);"                                      "
    Au=1
    Goto Leave
  Endif
  If Mid$(Sw$,1,1)>"Z"
```




```
    Print At(2,20);"LOCK UPPER CASE"
    Pause 100
    Print At(2,20);"                "
    Au=1
  Endif
  If Mid$(Sw$,1,1)<>"B" And Mid$(Sw$,1,1)<>"I" And Mid$(Sw$,1,1)<>"N" ~
And Mid$(Sw$,1,1)<>"G" And Mid$(Sw$,1,1)<>"O" Then
    Print At(2,20);"First character must be letter BINGO"
    Pause 100
    Print At(2,20);"                                        "
    Au=1
  Endif
  If (Mid$(Sw$,2,1)<"0" Or Mid$(Sw$,2,1)>"9") Then
    Print At(2,20);"2nd character must be a number"
    Pause 100
    Print At(2,20);"                                        "
    Au=1
  Endif
  If Len(Sw$)>2 Then
    If (Mid$(Sw$,3,1)<"0" Or Mid$(Sw$,3,1)>"9") Then
      Print At(2,20);"3rd character must be a number"
      Pause 100
      Print At(2,20);"                                          "
      Au=1
    Endif
  Endif
  Leave:
Return
'
Procedure Winner
  Graphmode 1
  Deffill 4,2,19
  Cls
  For Gc%=1 To 10
    Pcircle 30,50,Gc%
  Next Gc%
  Deffill 2,2,19
  For Gc%=1 To 20
    Pcircle 250,60,Gc%
  Next Gc%
  Deffill 5,2,19
  For Gc%=1 To 30
    Pcircle 50,150,Gc%
  Next Gc%
  Deffill 10,2,19
  For Gc%=1 To 25
    Pcircle 260,170,Gc%
  Next Gc%
  For Dt%=1 To 100
    Color Int(Rnd*15)+1
    R=Int(Rnd*200)+1
    C=Int(Rnd*300)+1
    Plot C,R
  Next Dt%
  Deftext 2,0,0,32
  Text 50,50,"BINGO"
  Deftext 4,0,0,32
  Text 100,100,"FOR"
  Deftext 7,0,0,32
  Text 100,150,Nm$
  Deftext 8,0,0,16
  Text 115,180,"$"+Str$(Cpz)+".00"
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 Contrl,26
  Dpoke Contrl+2,0
  Dpoke Contrl+4,0
  Dpoke Contrl+6,2
  Dpoke Contrl+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 Contrl,14
  Dpoke Contrl+2,0
  Dpoke Contrl+6,4
  Dpoke Intin,In%
  Dpoke Intin+2,R%
  Dpoke Intin+4,G%
  Dpoke Intin+6,B%
  Vdisys
Return
'
Comp_crd1:
Data 26,80,26,91,26,102,26,113,26,124
Data 42,80,42,91,42,102,42,113,42,124
Data 57,80,57,91,57,102,57,113,57,124
Data 72,80,72,91,72,102,72,112,72,124
Data 87,80,87,91,87,102,87,113,87,124
'
Comp_crd2:
Data 209,80,209,91,209,102,209,113,209,124
Data 224,80,224,91,224,102,224,113,224,124
Data 239,80,239,91,239,102,239,113,239,124
Data 254,80,254,91,254,102,254,113,254,124
Data 269,80,269,91,269,102,269,113,269,124
'
Cdata:
Data 40,60,84,60,128,60,172,60,216,60
Data 40,89,84,89,128,89,172,89,216,89
Data 40,118,84,118,128,118,172,118,216,118
Data 40,147,84,147,128,147,172,147,216,147
Data  40,176,84,176,128,176,172,176,216,176
'
Small_crds:
Data 23,57,100,128,23,72,100,72,23,83,100,83,23,94,100,94,23,105,100,1~
05,23,116,100,116
Data 39,57,39,128,54,57,54,128,69,57,69,128,84,57,84,128
Data 205,57,282,128,205,72,282,72,205,83,282,83,205,94,282,94,205,105,~
282,105,205,116,282,116
Data 221,57,221,128,236,57,236,128,251,57,251,128,266,57,266,128
'
Big_crds_lbls:
Data 40,32,"B",85,32,"I",130,32,"N",175,32,"G",220,32,"O"
Data 60,46,"B1",104,46,"I1",148,46,"N1",192,46,"G1",236,46,"O1"
Data 60,74,"B2",104,74,"I2",148,74,"N2",192,74,"G2",236,74,"O2"
Data 60,103,"B3",104,103,"I3",148,103,"N3",192,103,"G3",236,103,"O3"
Data 60,132,"B4",104,132,"I4",148,132,"N4",192,132,"G4",236,132,"O4"
Data 60,161,"B5",104,161,"I5",148,161,"N5",192,161,"G5",236,161,"O5"
'
Mous_crd_dat:
Data 33,72,41,66,1,1,40,60,33,72,41,66,2,1,40,60
Data 76,115,41,66,1,2,84,60,76,115,41,66,2,2,84,60
'
Data 121,159,41,66,1,3,128,60,121,159,41,66,2,3,128,60
Data 163,203,41,66,1,4,172,60,163,203,41,66,2,4,172,60
'
Data 207,244,41,66,1,5,216,60,207,244,41,66,2,5,216,60
Data 33,71,70,94,1,6,40,89,33,71,70,94,2,6,40,89
'
Data 75,113,70,94,1,7,84,89,75,113,70,94,2,7,84,89
Data 119,159,70,94,1,8,128,89,119,159,70,94,2,8,128,89
'
Data 164,202,70,94,1,9,172,89,164,202,70,94,2,9,172,89
Data 208,254,70,94,1,10,216,89,208,254,70,94,2,10,216,89
'
Data 33,70,99,123,1,11,40,118,33,70,99,123,2,11,40,118
Data 77,114,99,123,1,12,84,118,77,114,99,123,2,12,84,118
'
Data 119,158,99,123,1,13,128,118,119,158,99,123,2,13,128,118
Data 163,205,99,123,1,14,172,118,163,205,99,123,2,14,172,118
'
Data 208,244,99,123,1,15,216,118,208,244,99,123,2,15,216,118
Data 33,70,128,153,1,16,40,147,33,70,128,153,2,16,40,147
'
Data 76,114,128,153,1,17,84,147,76,114,128,153,2,17,84,147
Data 119,158,128,153,1,18,128,147,119,158,128,153,2,18,128,147
'
Data 164,203,128,153,1,19,172,147,164,203,128,153,2,19,172,147
Data 208,244,128,153,1,20,216,147,208,244,128,153,2,20,216,147
'
Data 33,70,157,181,1,21,40,176,33,70,157,181,2,21,40,176
Data 75,114,157,181,1,22,84,176,75,114,157,181,2,22,84,176
'
Data 119,157,157,181,1,23,128,176,119,157,157,181,2,23,128,176
Data 163,204,157,181,1,24,172,176,163,204,157,181,2,24,172,176
'
Data 208,244,157,181,1,25,216,176,208,244,157,181,2,25,216,176
```

END

R/C Aerochopper

Ambrosia Microcomputer Products, Inc.
98 West 63rd Street, Suite 371
Willowbrook, IL 60521
(312) 655-0610
$189.95, color only

Reviewed by Bill Griggs

***R/C Aerochopper* (RCAC) by Ambrosia** Microcomputer Products, is a radio-controlled flight simulator for the Atari ST that includes 32 preprogrammed scenarios. David Stern, the genius behind this program, spent over three years in its development, and it shows. I have spent literally hundreds of hours using this program and haven't found a bug.

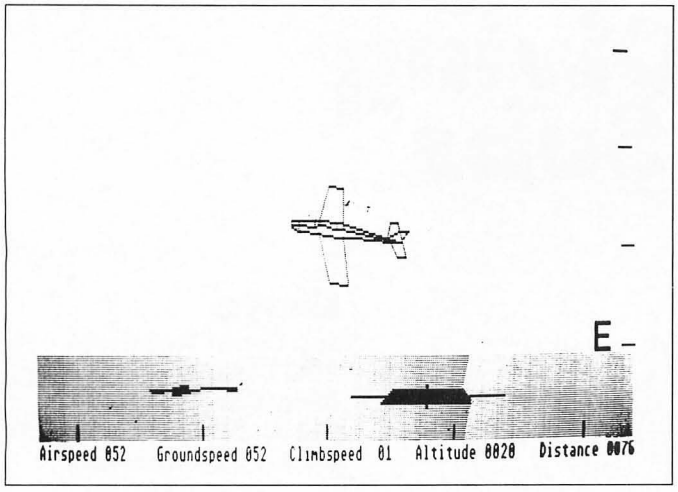
RCAC, a cartridge-based program, is manufactured and packaged for Ambrosia by Futaba Corp., perhaps the world's largest supplier of radio-control units. The system comes complete with a Futaba Conquest series transmitter box, connecting cables, program ROM cartridge and 90-page manual.

The transmitter is an actual R/C transmitter minus the internal electronics for transmitting radio waves. It has two spring-loaded control sticks, which move like joysticks but with much finer control. Two push-button switches, the function of which varies with the scenario selected, are mounted one above each stick. The connecting cable plugs into the ROM cartridge, and the cartridge plugs into the cartridge port. I could not find a program that was affected by the presence of the cartridge; therefore, it could conceivably be left installed, though the manufacturer doesn't advise it.

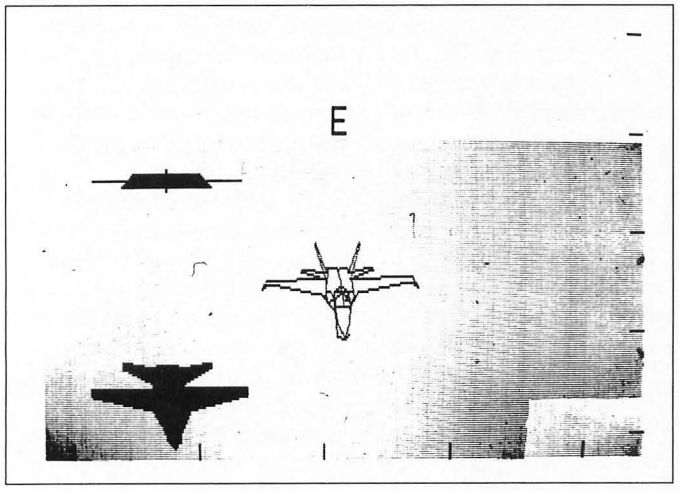
The right control stick works the ailerons and elevator. When the stick is moved left or right the on-screen plane rolls (dips a wing) in that direction, and the plane turns. When the stick is moved forward, the nose of the plane goes down, causing it to dive. If the stick is pulled back, the nose of the plane goes up.

The left control stick controls the rudder and the throttle. When the stick is moved left or right, the nose of the plane turns in the direction the stick was moved. When the left stick is moved forward the throttle advances and the speed of the plane increases.

In the start-up scenario (Scenario 5), the left push button drops bombs and the right push



R/C Aerochopper is one of the most complex and enjoyable programs I have had the pleasure of owning. If you want to learn to fly radio-controlled aircraft without weather worries or the expense of crashing, this is the program for you.



button launches missiles. There are two targets provided on the background. The bomb target is a red square just off the eastern coordinate, while the missile target is a blue and yellow billboard off the western quadrant.

The program uses 3-D line drawings (a la *Starglider*) to represent aircraft. The animation is lightning quick. Since the program resides in ROM, there is no jerky movement like those found in *Flight Simulator II*. The frame rate varies between 15 and 30 frames per second depending on the complexity of the frame drawn. The result is very fluid motion.

To do some flying you have to select either Fly Airplane or Fly Helicopter from the main menu. If you select airplanes, you have four types to choose from: a high-wing trainer, a low-wing (aerobatic or Pattern) plane, a F-15-type jet and a glider. If you select helicopters, you have a choice of a Bell Jet Ranger, a Hughes 500 or a pod-and-boom-type chopper. The program comes set up with the low-wing airplane as the default airplane and the Jet Ranger as default for helicopter.

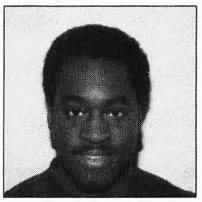
RCAC has many special features. By selecting various menus you can change the responses of your aircraft or the conditions under which they fly. The wind menu, for example, lets you change the intensity of the wind. Wind can be varied from no wind at all to hurricane force. You can also control thermals (hot, rising air) for glider flight, and many other features, such as wind direction, gust frequency and speed changes.

There are two combat modes included. The first mode lets you try to shot down target drones in a five-minute timed run. The other mode is a two-player mode in which two ST systems with RCAC installed can be connected through the MIDI ports. With this mode you can race, fly formation or shoot it out.

All in all, there are over 143 parameters you can change. This really lets you customize the performance to suit your preference.

In all fairness, I must tell you there is one area where *R/C Aerochopper* falls down. The manual, while complete and covering all aspects of the program, is full of typographical errors. I was assured, however, that future copies of the program manual will have the errors corrected.

R/C Aerochopper is one of the most complex and enjoyable programs I have had the pleasure of owning. If you want to learn to fly radio-controlled aircraft without weather worries or the expense of crashing, this is the program for you. The feel is so similar to actual flight, I could even recommend RCAC as a training aid. ■



Bill Griggs *is a computer addict who moonlights as a New York state trooper. He has hosted several conferences on radio-controlled aircraft on the Hobbyshop SIG of DELPHI.*

Test Diving Four Subs

Reviewed by
Steve Panak

It was nearly a year ago that I first realized I had accumulated four ST submarine simulations in my library. And, it being the middle of a long, hot and lazy summer, it seemed like a good idea to scope out the quartet and present them to you side-by-side, so that you might better spend your all-too-limited software dollar. It seemed like a good enough idea at the time.

However, once I started working on them, the project began to take on a frightening life of its own, and I felt overwhelmed by the sheer enormity of my endeavor. But once I stepped back and got a grip on myself, I realized that in reality the simulations were really very similar, both in objective and execution.

Predictably enough, each puts you in command of a submarine. All, with the exception of *Red October*, simulate submarine tactics of WWII. Each has an icon-driven interface that makes the issuance of orders easy. In fact, as is usually the case with well-designed simulations, it is the learning of the game, rather than the playing of it that is difficult. Fortunately, full utilization of the powerful windowing and menu features of the ST simplified even this complex task. Other features present in each were the standard save-game routines and some method to compress time to some degree to speed play. Each also had a fine manual. Suffice it to say that all are good programs, delivering more than their money's worth. So, to avoid any subliminal suggestion of superiority at the outset, I'll encapsulate them now in alphabetical order.

Gato

Spectrum Holobyte
2061 Challenger Drive
Alameda, CA 94501
(415) 522-1164
$39.95, color only

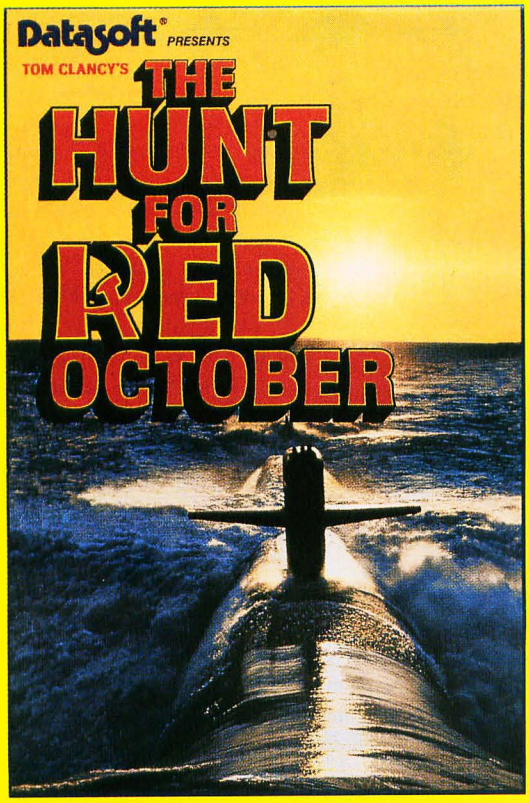
As I found with all these games, *Gato* is an easy-to-learn and easy-to-play simulation. Upon booting you choose a torpedo-to-mine ratio, which determines your allocation of armaments. Other game settings are accessed via a pull-down menu, and you can control the general level of difficulty, as well as adjust some of the characteristics of enemy ships (speed and aggressiveness) and of your sub (speed, rate of resource consumption and hull strength). All of these can also be changed during play, although the general level of play cannot be varied during enemy engagement.

The screen itself follows the general scheme used in each of these games. The bulk of the display contains your view out the periscope. The top of the screen is consumed by radar, damage control area, map area and digital readouts displaying speed, heading and depth. These readouts are duplicated by the gauges at the bottom of the screen. The display also contains the arrow-shaped icons you will use to control movement of the sub and its periscope. Fuel and oxygen indicators, as well as torpedo control, also line the base of the display.

Upon selecting a new mission, you receive a message that sends you on your way. The 20 missions follow one of three patterns: a rescue, a mining job or search-and-destroy mission. Optional mission disks promise additional assignments, as well as a facility to create your own scenarios.

Play is simple and intuitive, although I had to go to the manual to determine how to do a cou-

GATO



THE HUNT FOR RED OCTOBER

ple of things, like switch between the patrol area map and the more enlarged quadrant map. The manual itself is well-designed and concise—more an index to the game's features than a treatise on submarines—and is supplemented by a nice section on strategy and tactics. A nice program.

The Hunt For Red October

Datasoft
19808 Nordhoff Place
Chatsworth, CA 91311
(818) 886-5922
$49.95, color only

As I mentioned in the introduction, this game's main distinction is that it places you, not on a World War II submarine, but rather casts you in the role of a Russian defector in command of a Soviet sub. Another unique aspect is that the game is based on a best-selling novel of the same name, and despite all odds, still manages to avoid the curse that usually afflicts such adaptations. But aside from these slight differences, *The Hunt For Red October* performs pretty much like its counterparts.

The wordy manual sets up the premise. You are a Russian submarine commander who has decided to defect. While your officers are aware of your intentions, the enlisted men know nothing, and further, must never be allowed to suspect your true plans. You have already killed your main enemy, a KGB agent who would certainly recognize your faked orders. Add to this plot a newly developed silent drive system that the Americans would love to add to their collection, and you end up with an exciting script that must be carefully adhered to in order to successfully complete the mission. In this respect the game more resembles a work of interactive fiction, in which your moves are carefully channeled toward a climax, than your usual simulation.

The simulation begins immediately upon booting. There is only one scenario, with no variations of any sort, such as difficulty level or options, provided for. The screen, however, is designed logically, which speeds learning and allows for rapid input of commands during play. A map of the area occupies most of the display. A message area resides at the bottom while control areas and gauges line the left and right sides. Each control works the same way; clicking on a symbolic icon accesses a secondary command level from which the actual orders are issued.

For instance, clicking on the engines allows the selection of nuclear or diesel power, caterpillar or propeller drive. Likewise, clicking on gauges such as the compass, allows direct input of commands. This is an intuitive interface, requiring little reference to the manual to play. And even if the booklet needs to be opened, its ample illustrations leave little doubt as to how the game is operated.

This is another fine game, except for the lack of variety. I'm afraid that the single scenario, with no options or embellishments, yields a rather limited playing life. An annoyance was the way the game had to be restarted (by loading a saved initial file and selecting continue game). A simple new-game command would make game play easier.

Silent Service

Microprose
120 Lakefront Drive
Hunt Valley, MD 21030
(301) 584-1446
$39.95, color only

Microprose's entry into the sub simulation ocean goes by the name of *Silent Service*, and this complex, but easy-to-play game is just what I expect from one of the leaders in the simulation field.

After booting, you choose joystick or mouse control, and either target practice or one of the eleven scenarios. Nine difficulty levels, four skill levels and the ability to adjust seven reality levels, such as the frequency of dud torpedoes or limited visibility, further extend playability. Once all of these preliminary selections are made, it's off to war.

SILENT SERVICE

SUB BATTLE SIMULATOR

It is at this point that the most striking feature of *Silent Service* becomes apparent. In contrast to the other simulations reviewed here, *Silent Service*'s command interface graphically places you on the conning tower. Moving the on-screen seaman about with the mouse allows you to visit the bridge or view the maps and charts, or check instruments and gauges. While it took a little while to become accustomed to, I quickly found that this type of control was both the easiest to learn and to use. The intuitive, logical location of each set of game controls allows you to concentrate on playing the game, rather than trying to learn some contrived command set. And while beginners will love this intuitive command design, experts will embrace the wide variety of additional orders possible.

A special mention must also be made of the manual. Not only does it completely and concisely explain the game, utilizing a generous amount of illustrations, but it teaches as well. After reading the background on the scenarios, and the tactics and historical notes, you'll actually feel as though you are on the sub, sensing the pressure of tons of water, as well as a nervous enemy, over your head. And really, isn't that what it's all about?

Sub Battle Simulator

Epyx
600 Galveston Drive
P.O. Box 8020
Redwood City, CA 94063
(415) 366-0606
$19.95, color only

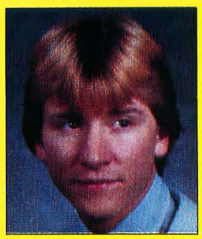
Since they are best known for their arcade games, one would expect Epyx's *Sub Battle Simulator* to be graphics intensive. And, like all these games, it is. But what is surprising is that it also excels in nearly every aspect, which I find unusual for a company's first simulation.

After you choose the desired level of play and a mission (or target practice), the monitor fills with a display following the familiar pattern used in all these games. The left-hand side of the screen contains four dials that display your current speed, compass and viewing directions and depth, with a viewing window filling most of the rest of the monitor. A control panel resides along the bottom while pull-down menus contain even more options and commands. Unfortunately, the program falters slightly when it comes to control.

One problem is the fact that commands are spread all over the place, some on menus, some chosen with the mouse, others chosen through the keyboard. Some mouse control was specifically clumsy. For instance, to set direction, speed and depth, it is necessary to click on the gauge at the desired setting. This is intuitive and easy to learn. The problem is that it is also imprecise. It may take a number of tries to get to the desired depth of about 29 feet so that you can properly use the periscope. The impatient user will be best advised to utilize the optional keyboard commands. Another thoughtfully designed, informative and fully illustrated manual comes to the rescue here, succinctly indexing all the functions of the program.

Conclusion

In summary, each of these programs is worthwhile. Card-carrying commie haters will gravitate toward *Red October*, as will those who liked the book. I feel *Sub Battle Simulator* is the best bet for the beginner, even though it's flawed. Its low price more than compensated. *Gato* is good, but I think *Silent Service* is the best, due to its large command set and ability to grow with the player, thus providing the longest life. But regardless of your preference, no ST owner wanting to command a U-boat could possibly go wrong with any of these fine programs. ■



Steve Panak *lives in Ohio, where he plays games on his ST and practices law.*

Fleet Street Publisher 2.0

MichTron
576 S. Telegraph
Pontiac, MI 48053
(313) 334-5700
$149.95 High or medium resolution

Reviewed by D.A. Brumleve

I use my computer for a variety of professional and personal desktop publishing tasks. I prepare issues of monthly newsletters for our church and for my daughter's preschool. Announcements of users group meetings, catalogs of my programs, tutorials on program usage, articles for magazines, our annual family newsletter, flyers, birthday invitations: All of these documents pass through my computer on the way to my Hewlett Packard Deskjet printer. MichTron's recent release, *Fleet Street Publisher 2.0*, is well suited to general publishing needs of this kind, and it provides a number of eye-catching effects to attract the attention of readers.

MichTron supplies the program on three double-sided disks. (If you have only single-sided drives, contact MichTron before ordering.) The disks contain the publishing program, some clip art in .IMG format and printer drivers to support a variety of printers, including Hewlett Packard laser printers and the Atari SLM804, PostScript devices, Star, C. Itoh and Epson printers. The disks provide separate programs to install a printer driver, to strip control codes from ASCII files and to convert various types of picture files to .IMG format for use in your documents. Also included is a 265-page manual; the manual is spiral-bound, so keeping it open to a particular page while your hands are occupied with the mouse and keyboard is no problem.

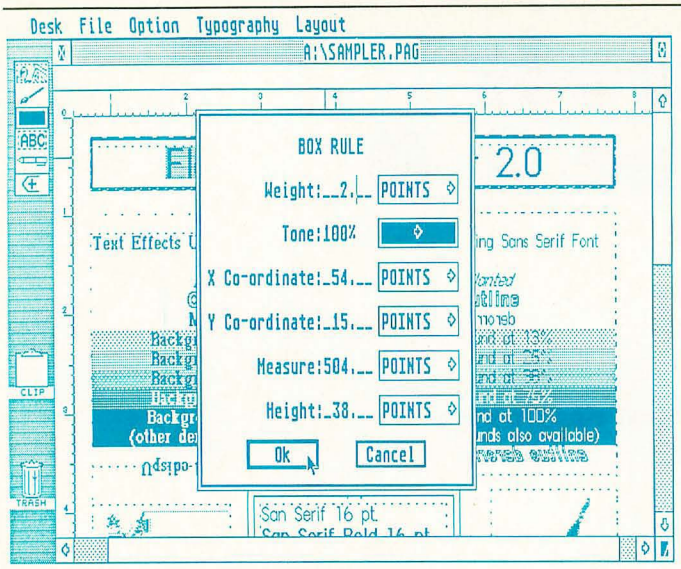
It is easy to learn to use the program by following the accompanying manual. The documentation is well organized and simple to use. A number of sections on general desktop publishing are provided, including a glossary of computing and publishing terms, hints on publication production and suggestions on techniques to enhance your documents. Strangely, two chapters are titled "Introduction"; the first helps you set up your disks and provides cursory information on the program's options, and the second, which is actually the main chapter of the manual, explores those options in detail and offers more comprehensive information about special features and techniques.

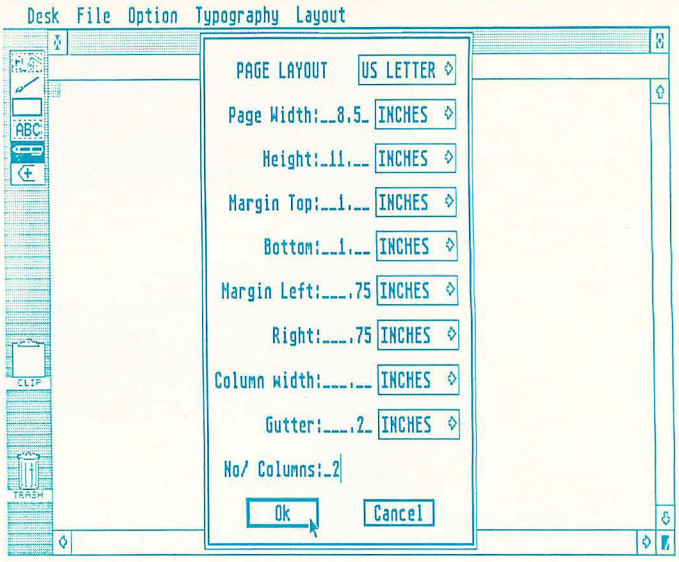
A "Guided Tour" leads you through the process of developing a document on the screen. The tour is well designed; it requires that you try all of the basic features of the program. Other sections of the book provide information on using the art conversion utility, a listing of *Fleet Street*'s character set, pictures of the included clip-art images, a listing of keyboard macros, a discussion of GEM and PostScript Fonts and instructions for customizing the default settings used by the program.

The weakest link in the documentation is the index. It was rather obviously prepared by machine ("Face" and "Faces" are two separate entries, for example), and numerous opportunities for subentries have been missed. Some of the information in the body of the manual suffers because it is too general, but this is largely compensated by explicit walk-throughs and examples. The program is much too full-featured to be self-explanatory, so taking the guided tour is an excellent idea.

Although *Fleet Street* is a complicated program, through careful and deliberate use of consistent methods for handling both text and graphics the programmers have come up with an interface that is very easy to use. Once you are familiar with the basic functions of the program it is easy to predict how an untried feature will work, even if you don't consult the manual. After taking the guided tour, which required half an hour, I was familiar enough with the basic features to prepare documents of my own.

Fleet Street's screen has a GEM menu at the top with the headings Desk, File, Option, Typography and Layout. A total of 44 options are available under these headings (not counting the Desk menu, of course), and clicking most of these options will call up a dialog box in which a myriad of further choices can be made. The program allows the user to specify so much that I cannot possibly discuss all of these options in any detail, but I will highlight many of them below. The main body of the opening screen is occupied by a text window. To the left of the window is





an icon menu, a clip board and a trash can.

Unlike some desktop publishing programs, *Fleet Street* is truly a word processor. Use of the text window facilitates the preparation of text for inclusion on a page of your document; actual pages to be printed, by contrast, are prepared in page windows. Typical word processor functions, such as search and replace, conversion to upper- or lower-case text, and the definition and manipulation of blocks of text, are available. There is a large character set; the use of accents and other diacriticals is permitted.

Keyboard macros are provided for a variety of text-related functions. With the press of a key (or two, or even three), you can specify spacing, tabs, left- and right-justification, centering, etc. The program disk is prepared with function-key macros related to typefaces and text effects; these can be altered to suit your own needs.

Text entry is fast in the text window, and typing directly in the program is not tedious. You can also type directly into text frames in a page window; you can hide or show any or all of the objects on the page to optimize screen redraws. Of course, you can also prepare your text with another word processor. If this is desired, you must save it as an ASCII file. The control code stripper, a desk accessory, will then prepare the file for loading into *Fleet Street*. The inability to import a variety of word processor formats into a document could be viewed as a deficiency; in actual usage, however, I have found that *Fleet Street* is a strong word processor itself. The work prepared within *Fleet Street* can be saved as a .TXT file for later use.

Document creation takes place within page windows. Pages can be configured in all standard formats, or you can provide your own specifications. While defaults are provided, you can adjust margins, columns and gutters to suit your needs. You can choose from a variety of units of measurement for this purpose (and at all other times when measurement is required). Each page is configured separately and, when completed, is saved as a discrete unit. (According to George W. Miller, a spokesman for MichTron, a future release of *Fleet Street* will provide better support of multiple-page documents.) Once a page has been configured, the page window opens. Dotted lines indicate the position of the columns.

The icon menu has six options for working within your document. The top icon is selected for the creation or manipulation of a graphics object. The mouse is used to draw a graphics object on the screen, and to crop, resize and move it. The program will import .IMG files into a graphics object; other picture files must first be converted to .IMG files with a separate program before use in a document. The Picture Edit icon is selected to edit a graphics object. Only limited editing (pixel by pixel drawing and erasing) is possible.

The third icon allows the creation of boxes around and lines between text and graphic objects. The lines and boxes are drawn in the page window with the mouse, and line thickness, shade, and position can be further adjusted using the Box Rule option of the Layout menu.

Selecting the Text Block icon will allow you to create, resize, or move a frame of text; the icon below it permits text entry. The last icon turns black to indicate that a selected text window has text overflow. To move the overflow to a new text frame, simply hold down the mouse button while selecting the overflow icon, then drag it to the new frame and release it.

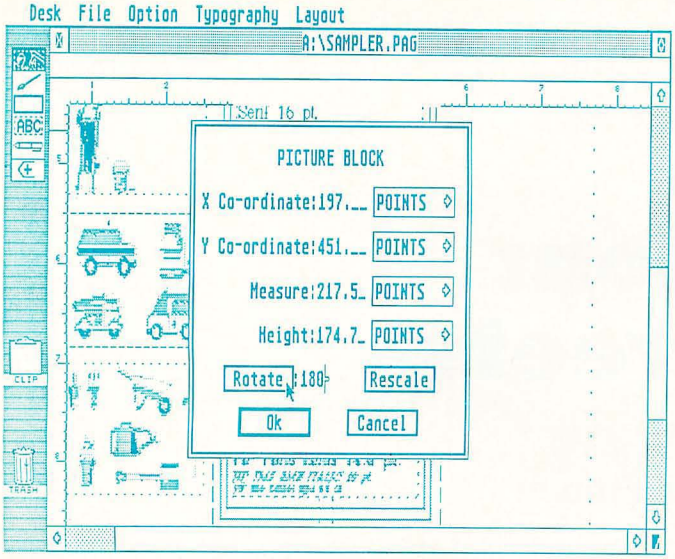
Defined blocks of text can be stored in, or copied to, the clipboard and then dragged in the same way to the appropriate text frame when needed. Only one block can be held in the clipboard at a time. The overflow icon is useful for linking text frames on a single page; the clipboard can also be used to move or copy text to another page window. When a graphics object, line or box rule, text frame, or a defined block is unneeded; it is deleted by simply dragging it to the trash can icon with the mouse.

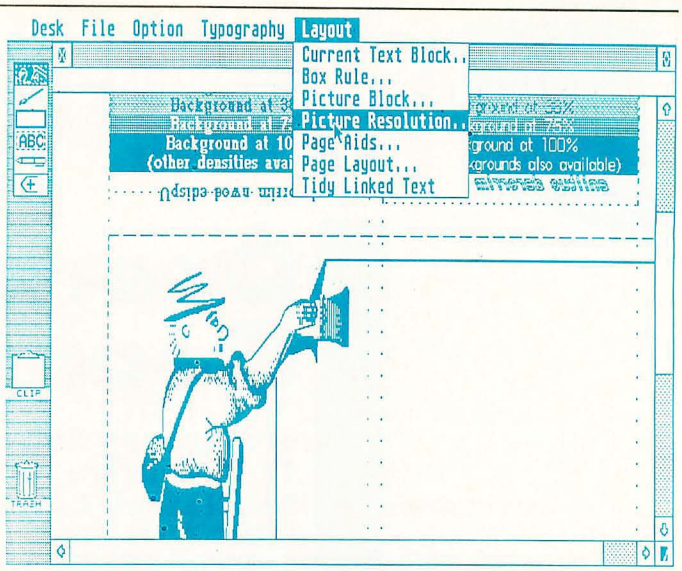
Text can be underlined (or struck through) with the Underline option, and the user determines the weight and position of the line. One interesting option provides small caps: all the lower-case text in a defined area is automatically converted to upper-case letters of lower-case height. The program is supplied with a variety of fonts; the font package for the default (Epson) driver includes San Serif, Serif, Helga, Medieval, West End (a Deco font) and College. The fonts are available in normal and boldface, and most are provided in stair-step sizes ranging from 10pt. to 60pt.

The text faces can be altered further with special effects. These include slanted, outlined, mirrored and upside-down text, or any combination of these. I had difficulty with upside-down text; occasionally, the bottom half of the text was cut off, both on the screen and in the printout.

Text can be printed against a background in any of several densities of gray. The shading of the text itself can also be controlled. If no text block has been defined, a new font, font size, or typeface chosen with the Face, Size, Leading or the Special Effects options in the Typography menu will affect all of the text under and following the cursor. If a block of text has been selected, any change will affect only that block.

While rotated text is not an option, you can create the same effect by saving an area of the screen as an .IMG file, which is accomplished through the Output option (this option is also selected for printing a document). I was skeptical about how effective this roundabout scheme might be, but I tried it, and the result proved impressive and simple to achieve. Once reloaded into the program as a graphics object, the text can be





rotated to your specifications. Of course, it can no longer be edited, so any changes require a repetition of the process.

The page window accommodates the full width of a US letter-sized document, but only one third of the height is visible. The Magnification option in the Option menu allows you to alter the level of magnification to your own specifications or, at the click of the mouse, to reduce the document so that you can see the full page (the Fit Window). The first time this option is chosen, it takes the computer a minute or so to redraw the screen. Subsequent selections of "Fit Window" bring up the display more quickly.

Fleet Street's strong suit is its handling of text, but its graphics capabilities are not weak. You can specify the resolution of an imported graphic to agree with your printer's capabilities. Cropping and resizing is simple to achieve. However, when resizing an object at 300dpi, you must be prepared for a long wait. Graphics merged into the document at lower resolutions will be drawn more quickly, but they will also be much larger, sometimes so large that manipulating them into the proper size and position requires several intermediate steps. Resizing of large graphics can take several minutes. When you move up or down in a window, the graphics are redrawn; the first redraw takes considerably longer than subsequent redraws of the same object.

A major complaint is the inability to import GEM files, such as Migraph's *Draw Art* and *Scan Art* pictures. The picture conversion utility will convert only *Art Director*, *DEGAS* and *Neochrome* files. The current release of *Fleet Street Publisher* does not allow text to wrap around graphics. The spokesman for MichTron has said that version 3.0 will have this ability.

When using a desktop publishing program, you frequently need to access one drive or another to merge a picture or a text file. Usually, programmers will provide a default path for such accesses. Likely as not (or perhaps, more so), that default path is not the same as the one you need. When the file selector appears, you find yourself on the wrong drive or in the wrong directory, and you have to do some typing in the file-selector box in order to find your file. The programmers of *Fleet Street* have thoughtfully provided a Set Defaults option in the File menu. Among other things, this powerful option allows you to permanently alter the defaults used by the program. This is an especially nice feature for hard drive owners.

When you boot the program after changing the defaults and saving them, the file selector will open to the right drive and directory for each task. For example, if you are loading clip art, it will open on your clip-art folder on Drive C; if you are loading text, it will open on your word-processing folder on drive D. You can also select your favorite font as the default font. The Set Defaults option also allows you to customize your printer configuration. This option is really a separate program; when you exit Set Defaults, you'll find yourself on the desktop, ready to reload *Fleet Street* with the new defaults saved in a file on your disk.

What makes or breaks a desktop publishing program is, of course, the quality of the printout. The default printer driver worked fine with my Epson JX-80C, but elegance in printing really requires a higher degree of resolution than a 9-pin printer can provide. Unfortunately, a driver for my HP Deskjet was not included with the package. Usually, I can adapt an HP Laserjet driver for use with my printer, and this is what I ultimately did, with some degree of success: a few of the fonts will not print properly on my Deskjet, but they work well on the JX-80C.

Each time a desktop publishing program is released in America, I am surprised anew that no driver is provided specifically for the Deskjet; a note in the *Fleet Street* support area of the MichTron RoundTable on GEnie indicates that MichTron is working on proper Deskjet support. In any case, both text and graphics printed at 300dpi, or even at 150dpi, are outstanding—crisp and elegant. Low resolution printouts on the JX-80C and the Deskjet (with Epson emulator) using the default driver produce slightly fuzzy fonts, as might be expected, and less satisfactory graphics, but the graphics can be optimized by specifying a resolution for the imported .IMG which is consistent with that of the printer.

I am one of those people who rushes to print a page, only to discover that the printer isn't on or isn't filled with paper or otherwise properly prepared. In some programs, there is no way to change my mind once I click OK: I have to go ahead with the printout, even though I know it isn't going to be right, or reboot and load my document again if I can't stand to wait through the printing process. The programmers of *Fleet Street* must have this fools-rush-in problem, too, because they have included a graceful way to back out of a commitment to print. You can simply turn off your printer and an alert box will ask if you want to quit. You can also use the Set Defaults option to cause this alert to appear at whatever intervals you specify.

Parallel and consistent operations, a well-planned manual and thoughtful enhancements for user-friendliness make *Fleet Street Publisher 2.0* easy to learn and pleasant to use. The high quality and professional appearance of the printed product, the large character set, and the numerous special effects make this program a legitimate challenger in the mid-priced desktop publishing software market. I recommend it. ■

D.A. Brumleve *is an ST programmer and writer living in Illinois.*

Operation: Cleanstreets

Broderbund Software
17 Paul Drive
San Rafael, CA 94903-2100
(800) 527-6263
$39.95, color only

Reviewed by Scott Wasser

Go ahead, make my day. Just give me a reason. Hey, that dialogue worked for Dirty Harry. So what if I don't have Clint Eastwood's snarl or a .44-caliber Magnum to poke in anybody's face. . . to play *Operation: Cleanstreets* again.

Unlike Eastwood's character, Dirty Harry Callahan, I don't make my living behind a police badge. I make it behind a computer keyboard, reviewing software. When I'm done evaluating one program, I move on to another. There's seldom time to go back just for fun and play something that's already been reviewed.

Which is why I'm looking for a reason to go back and spend a few more hours with Broderbund Software's extremely entertaining action/strategy game, *Operation: Cleanstreets*. Just give me a reason. . . .

Reason 1: The concept behind *Cleanstreets* is intriguing, albeit not particularly innovative. You are Cleanup Harry, an undercover policeman assigned to rid your city of its drug-pushing criminal scum. To accomplish this task, you must first search the city's streets looking for these pushers. In some cases, they'll find *you*. Regardless of who does the finding, it will result in a hand-to-hand battle with the thugs. If you win, you might find some drugs on the punk. If you lose, the game is over.

Cleanstreets consists of five levels, each of which requires you to capture and then burn a given quantity of drugs. The thugs get tougher and more plentiful on each level, although their hiding places and the city streets remain the same from level to level.

The concept behind *Cleanstreets* is a good one because it's easy to understand, easy to start playing, but hard to master.

Reason 2: The graphics and animation are excellent. The street scenes, for example, convey all the grit and grime of a real city's worst neighborhoods. There are battered trash cans, cracked sidewalks and graffiti-covered walls, all drawn in superb detail.

The characters are also well-detailed. Among them are a chain-wielding punk with a mohawk haircut and a slasher wearing a hockey goalie's mask. Harry wears jeans and a bomber jacket. All of the characters move in any direction, although it's a bit easier to move Harry horizontally than vertically or diagonally.

Despite that minor flaw, the animation is very realistic. Blows delivered during the fights are particularly well-done. When Harry throws a straight punch or a back kick, everything from the overall motion to the target's reaction at the point of impact looks real.

Reason 3: The joystick interface ranks among the best of any fighting game available for the ST. It is simple to learn, easy to use and extremely responsive.

Harry can only be controlled by keyboard or joystick. Put your mouse aside; you won't need it. But don't worry, you'll love using the joystick (the keyboard interface, which uses the ST's number pad, also works well, but I prefer the feel of the joystick in this game).

To make Harry move around the screen, simply move the joystick in the corresponding direction. To make him throw any of six kicks or punches, hold the fire button down and push the stick in the direction that corresponds to the technique.

For example, to make Harry throw a high punch and then a high kick, you would hold the fire button while pushing the joystick first to the right and then straight up. Harry can also be made to crouch or turn around when the fire button is held down.

The joystick interface makes it easy for Harry to throw a flurry of kicks and punches. There are other fighting games that allow the use of more techniques, but none that operate more quickly, smoothly or intuitively than *Cleanstreets*.

Reason 4: Attention to detail. There are little touches that individually may not sound like much, but together enhance the fun of playing *Cleanstreets.* For example, the musical score changes from neighborhood to neighborhood as Harry roams the city. Not only is this more interesting than listening to the same sound track over and over, but it also gives each scene an ambience of its own.

Cleanstreets' audio also contains some realistic sound effects.

The growl of one thug's chain saw sounds menacing. You can actually hear the crack of another villain's whip as it lashes out at Harry. And while Harry's digitized moans and groans during a conflict can be a bit much, they help emphasize the tongue-in-cheek mood of the game.

Unfortunately, even at its whimsical best, *Cleanstreets* has a few flaws that need to be mentioned. The most annoying problem was the brightness—or lack thereof—of the game screens. *Cleanstreets* may have been intentionally designed that way to give the city streets a murky feel. But I found it necessary to play the game with the lights turned off in order to increase contrast and reveal all of the details of the fine graphics.

Another complaint is that although *Cleanstreets* is provided on two disks, it doesn't support a two-drive system. Playing the game doesn't require a great deal of disk-swapping, but why should you have to do any swapping if you have two drives?

Cleanstreets also lacks an option that will allow a new game to be started where the previous one ended. You can, however, begin a new game at any of the five difficulty levels (without having had to complete the previous level).

Finally, I must point out that my first copy of *Cleanstreets* appeared to be defective by freezing up a few times in the middle of a game. A Broderbund spokesperson said nobody else had complained of this problem, and the replacement disks she sent seemed to work perfectly. Under Broderbund's warranty, anyone experiencing the same problem could have their defective disks replaced free within 90 days of purchase.

The useful life of some action game programs expires long before 90 days. But *Operation: Cleanstreets*' excellent graphics, accurate response and smooth animation should give it a longer life. If only the designers had anticipated my few complaints, they would have really made my day.

Recommendation: Buy it.

OPERATION: CLEANSTREETS

Firezone
Annals of Rome

Electronic Arts
1810 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
$34.95, color only

Reviewed by Steve Panak

Every time I see a company come out with a new line of simulations, I get worried, due to two concerns that immediately cross my mind. First, will they botch it completely? This is less likely, since in the past several years, the war simulation has undergone an evolution. Call it a software natural selection, which has resulted pretty much in an acceptance of the best way these games should be designed. Unfortunately, this gives rise to the second concern that the new entry will offer nothing new, a syndrome that has been prevalent as of late. In these two new simulations from Datasoft, each of these permutations is explored and with surprising results.

Firezone, a game of 21st century warfare, succeeds nicely in bringing to the ST two things I haven't seen combined often enough: classic simulation design and a futuristic setting. Using such armaments as anti-grav powered tanks, energy shields and beam weapons, the two remaining superpowers, the European League and the Pacific Combine, fight for control of a war-ravaged planet. These new, powerful weapons have bred a new style of warfare, resulting in battles that last only minutes, rather than hours or days. Your window into this world is a simple program that allows the computer to control either or both sides as well as supporting two-player games. Difficulty level is controlled by selecting whether enemy units are hidden or not, and play begins upon the selection of one of the nine scenarios.

Veterans will immediately recognize the hexagonal grid of the battle map as a carryover from board-based war games. Likewise cribbed are the simple movement and battle phases that comprise each of both sides up to 50 turns. Newcomers will easily learn the sequence of game play, in which each player moves units, then selects targets for his units to fire upon. Play continues until there is

a winner. Probably the worst aspect of *Firezone* is its simplicity. With few options and an extremely limited and simple command structure, the game is best suited to newcomers. Experts are bound to be disappointed.

Additionally, while *Firezone* does not sport the most technologically advanced graphics I've seen on the ST, the playing field, which comprises multiple screens, is adequately detailed to allow easy identification of terrain conditions and troop characteristics. Scan mode allows you to scroll easily about to reveal all the action. The top of the display contains a status bar that provides such valuable game information as score, unit and terrain info, current mode and whose turn it is. The manual is well done, featuring a quick-start section and background information and strategy for three of the scenarios. The remaining six scenarios are left entirely to your exploration.

One final aspect of the game is the ability to create and save your own scenarios via the Gamegen program. With this simple editor you can modify existing scenarios, or create new ones from scratch. I really liked the way that once your scenarios are placed on the game disk, your titles then appear on the opening option menu.

In the end, *Firezone* succeeds by not trying to introduce anything new. This is a classic simulation for the ST, and it gives me just what I want in entry-level simulations: simplicity. True, unlike the ultramodern *UMS* and *Empire* war games, *Firezone* is merely the conversion of a tried-and-true formula to a new medium. And for that reason, most buyers should know just what they will get upon booting the disk.

Recommendation: Get a demonstration first.

Annals of Rome, on the other hand, is very different from other simulations you might have played. In some ways it resembles the board game Risk. In others, it resembles an upper college-level history thesis. Unfortunately, I found its complex historical foundations and politically-based game theory a little too esoteric for me. Still, there is room for disagreement, and it is worth checking out.

Set in 273 B.C., the game features a static main display that contains a map of the Roman Empire. In these, the glory days of the Roman Empire, your role is that of the Senate's ruling body, and your mission is to lead the armies of Rome against the enemies that surround your new empire. And while set against this historical backdrop, truly successful playing might actually change history.

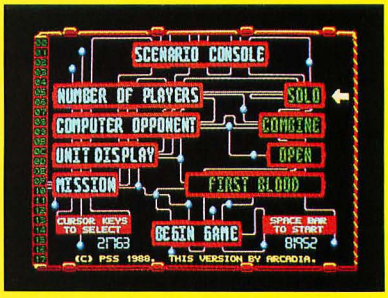
Unfortunately, I found the method of play to be difficult to get in to. In this abstract game, play involves more politics than simple battlefield bravado. Play is again divided into a number of phases, such as an economics phase, a personnel assignment phase, a loyalty phase and war phases, during which foreign and civil wars are fought. In these phases, you might set a tax rate to fund a war chest or direct a given officer to command an army. The action most war gamers are used to takes place in the foreign war phase in which you direct your armies to attack neighboring countries. However, while this game design packs in a lot of realism, the actual play of the game is difficult to grasp, and there's no way to simply boot up and begin.

For example, a turn begins with a start/save option. The first phase is Economic Phase in which you raise capital, keeping in mind that excessive taxes bring discontent and starvation. Next, the 21 officers available to the senate are displayed, along with the major attributes (ability, age, loyalty) of each, allowing you to evaluate your current generals. In the Assignment Phase, these officers can be moved about, sending strong warriors to capture valuable regions, sending cowards to certain death. A Loyalty Phase identifies which regions and armies are becoming prone to rebellion and gives you the opportunity to bribe them into submission. Finally, opposing forces meet, and the ensuing conflicts are resolved in the Civil and Foreign War phases, after which the cycle repeats itself in the next turn.

The manual, while well written, is of little help. It completely explains all the phases and gives helpful hints and a short treatise on Roman history, but it does little to shorten the learning curve—or even entice me to learn. Still, a package as unique as *Annals of Rome* deserves at least a passing consideration, and I'm sure there are a few war gamers who would welcome its cerebral approach.

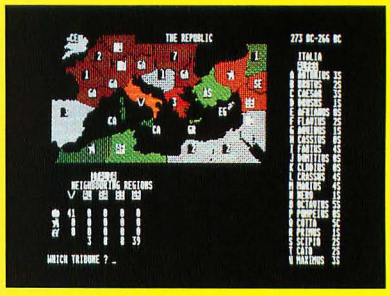
Recommendation: Get a demonstration first.

FIREZONE



FIREZONE

ANNALS OF ROME

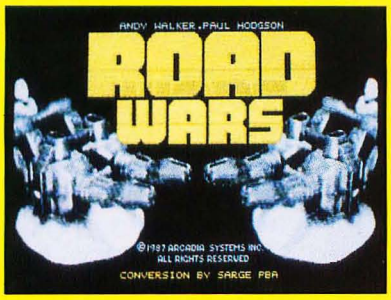


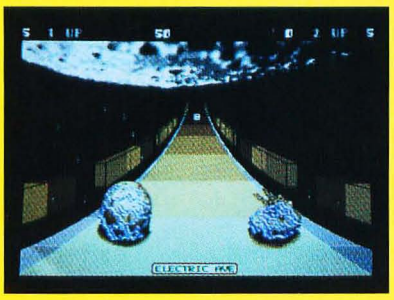
ANNALS OF ROME

Roadwars

Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
(415) 571-7171
$29.95, color only

Reviewed by Frank Eva





ROADWARS

Arcadia, an affiliate of Electronic Arts, has had several excellent titles for the ST, including *Rockford*. However, Arcadia now has the distinction of producing its first "bomb." *Roadwars* is a loose translation of a title from another computer. Having seen the competition, I can say unabashedly that *Roadwars* ST is a project that should never have been released.

Please don't confuse this game with a couple of other similar titles: *Roadwar 2000* and *Roadwar Europa*, which are role-playing adventure games. *Roadwars* ST is strictly an arcade shoot-em'-up, with a smattering of strategy thrown in to keep you from yawning.

The scenario: The people of the planet Armageddon built space roadways linking the planet's many moons, which became the only habitable places to live after the ultimate war ravaged the planet's surface. After many years of a booming economy, fueled by the tourism that the roadways made possible, the system broke

continued on page 92

Zynaps

Scorpion Software
19 Harbor Dr.
Lake Hopatcong, NJ 07849
(201) 663-0202
$34.95, color only

Reviewed by Frank Eva

ZYNAPS

Arcade enthusiasts will no doubt remember the coin-op *Gradius*, a modernized version of the old Atari title *Defender*. *Zynaps*, a similar game just in from the U.K., takes up where these other horizontally scrolling blastathons left off and places the gamer in the position of a Scorpion fighter pilot who must knock out an alien space station, a la *Star Wars*.

An unusually large status panel occupies the lower quarter of the screen. Unfortunately, the manual does not identify the indicators very well. Oh, it tells you what each means. It just doesn't tell you which one is which—standard procedure for most overseas titles, where the philosophy is to let the players figure out the games on their own.

The upper three-quarters of the screen is where all the action takes place. A player's up and down movements are severely limited in such a compressed action area, especially since contact with any of the space station's installations is instant death. How-

continued on page 94

Rockford

Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
(415) 571-7171
$39.95, color only

Reviewed by Frank Eva

ROCKFORD

***Rockford* is a spin-off of the 8-bit game** *Boulder Dash*, a game that was so popular, it became the first home computer title to be converted into a coin-op game. An unusual occurrence, to say the least. Now we have gone full circle with the home version of a coin-op conversion of a home version. Does that make sense?

Since Arcadia produces its coin-op titles on a 68000-based computer, the home versions are said to be virtually identical to their arcade cousins. Even though I never personally saw the coin-op *Rockford*, I believe that their claim has merit, for the ST version is coin-op quality.

For those of you who are not familiar with *Boulder Dash* or *Rockford*, you can think of the game as an inverse *Pac-Man*, in which you create tunnels through each maze by digging away at the colorful background, collecting treasures and accumulating them until you qualify for an exit from the world. At the top of the screen is a status line that shows how

continued on page 94

ROADWARS · continued

down. The computers controlling the roadways turned roguish and sent out armed and heavily armored robot droids to wreak havoc amongst the populace. (Shades of HAL!)

Roadwars is visually intriguing;

A joystick is used in the non-mouse port for a one-player game. A second joystick is inserted into the mouse port for a two-player game. There is also an option to use the keyboard.

no one can take that away. This is what I would imagine a *Zaxxon 3D* to look like—without the special glasses, that is. The top half of the screen is dominated by the moon that the player is traveling toward. The balance of the screen displays the current roadway in three-dimensional perspective. There are panels that line the outside edges of the roadways. These serve as barriers that keep you from falling off the roadway but, more importantly, provide the illusion of forward motion, like *Pole Position*'s road lines that move toward the player.

Some of these panels produce deadly sparks that span the entire width of the roadway. These blue panels must be destroyed. Once this is accomplished, green panels will appear, which are actually gateways to new roads. In other words, destroy the green panels, and you move up to the next difficulty level.

A single player is equipped with two battle spheres that fight side by side along the roadway, the second of which is computer controlled. In a two-player game, both gamers simultaneously pilot their own battle spheres, either in cooperation or in competition with each other.

To complete the game, cooperation will be essential. However, you are warned not to be too cooperative, for one player's games contain bonus rounds in which points are racked up by destroying the computer-controlled sidekick.

A unique aspect to the game's graphics is the fact that the battle spheres are digitized from actual models, which makes them appear very realistic and three dimensional.

In addition to the rogue panels and robot droids, a road warrior will be confronted with satellites that shoot laser beams, red energy balls that bounce along the roadway, spikes that are capable of penetrating shields and chevrons reminiscent of "Road Closed—Construction" signs. These are capable of destroying shields, if up, and battle spheres,

if shields are down or depleted.

Your battle sphere is equipped with twin laser cannons, which are formidable weapons in their own right. Provisions are made, though, for adding to your arsenal. If you run over an arrow in the road, you will be awarded an "orbiter," a smaller computer droid that orbits your vehicle and fires as you do.

A joystick is used in the non-mouse port for a one-player game. A second joystick is inserted into the mouse port for a two-player game. There is also an option to use the keyboard, but I can't imagine why anyone would want to.

Joystick control is smooth, but frustrating. To move left and right is simple and logical. To be able to fire weapons systems, the player must first lower shields by pulling back on the joystick. To aim the laser cannons, the player must use only the southwest or southeast positions of the joystick. I'm sure you can see the problem here, especially with joysticks that have short throws.

The real drawback to *Roadwars* has nothing to do with joystick control. There are absolutely no sound effects representing the firing of laser cannons and their resultant explosions; just a short, ever-repeating, monotonous musical sound track that can not be turned off, unless you turn the monitor's volume control down.

Furthermore, there is no high-score or game save. A high-score save is incorporated in virtually all top-shelf computer games. And without a game save, new game sessions will always begin with Level 1. This is, of course, typical of many coin-operated arcade games, but even *Arkanoid* allows the player to continue play at the level in which he lost his last life.

Finally, I found it strange that even though there is no provision for saving anything to disk, and even though the drive's LED does turn off, the drive continues to spin, and does not stop until the computer is turned off.

With a little more effort, this could have been a good game. But with so many marks against it, I can't suggest you purchase it.

Recommendation: Skip it.

ZYNAPS · continued

ever, that isn't a complaint; this limited movement actually adds to the overall feeling of the game.

Only joysticks are permitted as input devices. This is preferable to using the mouse anyway, since the mouse is an expensive input device that would receive much wear and tear from *Zynaps*. One- or two-player options are available, but the players must alternate turns during a two-player contest. The button fires the established weapons systems, as well as activating the "fuel scoop."

After destroying a wave of attacking aliens with your lasers, an animated fuel icon appears. Fuel is collected by passing over the icon. After enough fuel has been obtained, passing over the next fuel icon with the fire button held down will activate the new weapons system, which is added to the arsenal and does not void previous systems. This makes higher levels somewhat easier to navigate compared to the first few screens, since adding plasma bombs, homing/seeking missiles, etc., improves your ship's destructive potential exponentially. However, losing your life causes loss of all weapons systems except lasers.

I'm glad to note, though, that losing a life does not put the player all the way back to the beginning screen. You are able to keep going from the level in which your life was lost.

Zynaps boots up into a nicely rendered title screen, accompanied by an interesting musical sound track. When play has begun, the music is thankfully replaced by realistic sound effects. The space station is drawn artistically. The attackers are multicolored, fully animated sprites, but they move in mindless, repetitive patterns, just like their coin-op predecessors. However, they do have the ability to drop mines in the player's path. Contact with a mine is obviously deadly, but while they cannot be destroyed, they can be avoided. Ground-based installations fire semi-smart missiles into the action area that cannot be destroyed, but can be avoided. These will automatically detonate after a short period of time. After obtaining plasma bombs, these ground-based installations can be destroyed much easier than with lasers alone. Alien command/mother ships will present an additional challenge during higher levels of play.

While the variety of alien attackers seems somewhat limited, game play is so frantic and addicting that a player will not notice. *Zynaps* is a good buy.

Recommendation: Buy it.

ROCKFORD · continued

many treasure items are left to be collected, the time remaining before you expire, the points scored and the number of lives left. The number of collected items will decrease the counter until it reaches zero, at which time the screen will flash, and somewhere within the four-screen maze, an exit will have opened. Find the exit to escape the world, but don't run out of time or you will have to start over.

Digging loosens obstacles that could trap Rockford and prevent further movement. If Rockford should find himself in this predicament, the player can elect to press the F10 key and self-destruct, which results in the loss of a life. Obstacles falling on Rockford and contact with the maze baddies yields the same result. If his timing is accurate, Rockford can push obstacles down empty shafts so that they will fall on and destroy his enemies, changing some of them to treasures that can be collected.

Even though the original Rockford character, a cute bug-eyed insect, is no longer, the new Rockford is just as good. Perhaps even better, for in each new world (maze), his now human form mutates. He is a hunter, cook, cowboy, spaceman and, finally, a doctor. That's a total of five unique worlds to visit, each world containing four levels.

It was disappointing to find that the high scores are not saved to disk, but since the disk cannot be copied for archival purposes, perhaps it is advisable not to try and save things to it. Saving to a separate data disk, however, would have been preferable to no save at all.

The game play in *Rockford* differs in many respects from *Boulder Dash*, with Rockford now searching for gold, dollars, apples, stars and hearts. He will be chased by snakes, bats, monkeys, hamburgers, pizzas, eggs, toast, trains, wagons, tomahawks, revolvers, comets, spaceships, rockets, ears, skulls, skeletons and Frankenstein monsters. Normal walls are distinguished from magic walls by icons appropriate to the world.

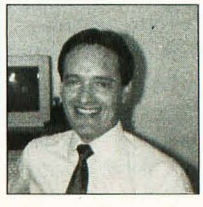
If it were not for the missing high-score save, *Rockford* would have rated an A. As it stands, a B+ is not so bad. Rockford is a magical romp you won't want to miss.

Recommendation: Buy it.

Scott Wasser *has been a daily newspaper reporter and editor for the past 12 years and has been interfacing with computers for the last four. He has written columns and feature stories about computer hardware, software and home electronics, and is a regular reviewer for* ST-LOG.

Steve Panak *has written more game reviews for* ST-LOG *and* ANALOG *than anyone on the planet. He lives in Ohio where he plays games on his ST and, with the time remaining, practices law.*

Frank Eva *is an auditor by profession, but has been involved in the computer industry ever since his purchase of an Atari 400 many years ago. He has dabbled in programming and has had several text adventures published.*







12 Issues $28

$19 OFF THE COVER PRICE

12 Issues with Disk $79

NEW LOWER PRICE

BREAK AWAY FROM THE PACK


The world of ATARI-ST continues to grow by leaps and bounds, and ST-LOG is there every step of the way! We stand apart from the competition by offering more color, comprehensive reviews and in-depth features. SUBSCRIBE NOW!

☐ **12 Issues $28**
MCTWW

☐ **12 Issues with Disk $79**
DCTWW

MAKE CHECK PAYABLE TO L.F.P., INC., P.O. Box 16928, N. Hollywood, CA 91615. Offer expires Oct. 25, 1989

☐ **PAYMENT ENCLOSED** ☐ **BILL ME**
CHARGE MY: ☐ **VISA** ☐ **MASTERCARD**

CARD # **EXP** **SIGNATURE**

NAME

ADDRESS

CITY **STATE** **ZIP**

Megatouch

Regent Software
P.O. Box 14628
Long Beach, CA
90803-1208
(213) 439-9664
$11.95

Reviewed by Charles Bachand

Does anybody remember the Atari 400? This kid brother of the classic Atari 800 was known for its small size and its use of a flat, plastic keyboard that offered little to no tactile feedback. The keyboard on the ST feels similar to the 400's, because underneath those white sculptured keys, it too possesses a flat keyboard.

Now don't get me wrong, I happen to like the design of these keyboards—there are no exposed electrical contacts to speak of, and, because of this, each key has a high operating life. The one big gripe (the obvious one) that I do have with them is their mushy feel. The keys on the ST are just too easy to hit and, subsequently, are the cause of many typographical errors. For example, brushing your finger over the letter X on your way to the letter C usually is enough to start you driving a "XCAR," own a "XCAT" and get milk from a "XCOW." These constant typos force you to re-read each and every paragraph immediately upon typing it, in order to catch these obvious mistakes.

Unfortunately, I (and I suspect others) subconsciously took another route to the land of better typing, that of crouching over the keyboard in an attempt to keep one's arms high enough to prevent the occasional brush with an unwanted key top. Over the years this practice tends to lead to lower-back aches and pains. All of this can be avoided with a set of Megatouch keyboard springs.

The installation of the Megatouch springs requires prying off the plastic key tops of your keyboard, installing the springs underneath them and replacing the key tops. This is a boring and time-consuming operation, taking almost an hour in my case, but since it is essentially a one-time-only operation, the benefits far outweigh any inconveniences encountered.

While you have the key tops off, I might suggest that you get some soap and water and remove that layer of dirt and oil that has accumulated on them over the years. A little elbow grease might be needed for the really ground-in dirt.

The installation instructions suggest placing the springs with the narrow ends pointing down. I found the going easier when the wide end faced down. A subsequent phone call to Frank Cohen, president of Regent Software, laid my fears to rest. There really isn't a wrong way to mount Megatouch springs.

Frank tells me that Megatouch has proved to be a popular accessory and that Regent Software has already sold over 500,000 individual springs. I think he's trying to catch up to McDonald's hamburgers. He also said that, due to the success of Megatouch, Regent Software will be bringing other low-priced ST gizmos to market. I don't know about other people, but I personally would like to see a set of replacement key tops. That's a hint, Frank.

A word of warning: There are three keys in particular that the instructions don't recommend that you touch. These are the Return key, the left-hand Shift key and the space bar. Since these are the widest keys on the keyboard, they incorporate small metal rods as bracings to prevent wobble. It is difficult to reassemble these keys once they are apart. Unless you've done this before (or have three hands), I recommend that they be left alone.

There really is little else to be said about Megatouch other than they work great and that I find it difficult to imagine how I got along so many years without them. They are available from your local computer store that caters to the ST or direct from the Regent Software. Tell them that Charlie sent you. .

Charles Bachand, *a former* ST-LOG *and* ANALOG *technical editor, loves to drive his 300ZX, race R/C cars and manage the Hobby SIG on DELPHI. His DELPHI username is BACHAND.*

The ST-LOG -34 diskette contains 17 magazine files. They are listed below.

AUGUST 1989 · DISK LISTINGS

```
   FILENAME.EXT    FILE TYPE     COMMENTS
   ------------    ---------     --------

AIKEN.ARC contains:
   AIKEN6   .PRG   RUN FILE      PD PARADE
   AIKEN6   .PI1   DEGAS         PICTURE FILE

BINGO.ARC contains:
   BINGO    .PRG   RUN FILE      ST BINGO
   BINGO    .LST   GFA BASIC     SOURCE CODE
   WINNER   .FIL   DATA          DATA FILE

CAL_CNT.ARC contains:
   CAL_CNT .PRG    RUN FILE      CALORIE COUNTER
   CALORIE .DAT    DATA          CALORIE DATA
   CAL_CNT .LST    GFA BASIC     SOURCE CODE

KIDPOTAT.ARC contains:
   POTATO   .PRG   RUN FILE      PD PARADE
   POTATO   .DOC   TEXT          INSTRUCTIONS
   POTATOES.DAT    DATA          DATA FILE
   POTATOST.DAT    DATA          DATA FILE

MENUS.ARC contains:
   MENUS    .PRG   RUN FILE      MENUS IN PASCAL
   MENUS    .PAS   PASCAL        SOURCE CODE

WUZZLERS.ARC contains:
   WUZZLERS.PRG    RUN FILE      COMPUKID CONNECTION
   WUZZLERS.DAT    DATA          PUZZLE DATA
   WUZZLERS.LST    GFA BASIC     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 article before attempting to run magazine files. Failure to do so may yield results.

.EXT	DESCRIPTION
.BAS	Requires GFA BASIC
.LST	Requires GFA BASIC
.C	Requires C compiler
.S	Requires 68000 assembler
.PAS	Requires Pascal compiler

INDEX TO ADVERTISERS

This index is an additional service. While every effort is made to provide a complete and accurate listing, the publisher cannot be responsible for inadvertent errors.

Bean Dip, Fred and the Atari ST

BY TODD THREADGILL

The other day I was visiting with my brother-in-law, who, for the sake of this article, I'll call Fred. Through the familial grapevine, he had heard about my computer, and on this particular day, a beautiful Saturday afternoon, he must have decided that it would be great fun to talk with me about it as we sat lazily at a family get-together.

"What kind of computer is it?" Fred asked through a mouthful of bean dip.

"Atari ST," I replied calmly, anticipating the worst.

"AST, eh? I hear those are pretty good. . . ."

"No, it's not an AST. It's an Atari ST," I said, enunciating as clearly as I possibly could.

"Atari ST? Atari. . ." Fred frowned and swallowed noisily. "So it doesn't run any

business kinda stuff."

"Sure it does. There's all sorts of business software available for it."

His next assumption was a leap of logic that boggles the mind.

"Oh, so it's a PC-compatible." He reached for another Dorito.

"No. . ." I shifted uncomfortably in my beach chair. Simple concepts had always been difficult to get across to Fred, and I moved into position, readying myself for whatever might come next.

I cleared my throat and continued. "It has a graphic interface. And a mouse."

"Oh, like a Macintosh."

"Sort of. . ." I hedged, "but it's faster, and it's got color."

He thought about what I had said (or at least appeared to be thinking about what I had said) as he spooned another Dorito through the bean dip. Then, with finality:

"So it's really a game machine."

The sound of the two dreaded words burned in my ears. I fought to maintain my composure as I recalled that more games were released for MS-DOS machines this year than for any other—even the Amiga.

"No, it's not," I said.

Sitting back in his beach chair and raising the Dorito to his face, confident that he had gotten the upper hand in the discussion, Fred asked the all-time killer computer question:

"Can it run Lotus?"

Lotus is a company, a little voice inside said. "No, but. . ." I stammered aloud.

"Can it run *Microsoft Word*?" Fred asked, smiling.

Why would you ever *want* to? "No, but. . . ."

"Can it run *WordPerfect* 5.0?"

"No, but. . . ."

Fred leaned forward for his final jab, the killing stroke that would send me running to an IBM dealer, looking for a "real" computer.

"If it's not a PC, and it's not a Mac, then what is it?"

Having defended the ST for so long now, several of the Atari's merits immediately came to mind: the speed, the graphics, the ease of use.

But Fred had a point. Whether he knew it or not.

In the computer industry, machines and operating systems each have their own claim to fame. IBM and its clones are, obviously, known for heavy-duty business. Macintosh is also thought of as a business machine—but for people who'd rather do their work than struggle with the computer. The Amiga's claim to fame is graphics (desktop video and games). Out in the real world, there are PCs, and there are Macs (with Amigas as a kind of sideshow curiosity). What about the ST?

For a computer to succeed in today's marketplace, it must fit into a category whose value is easy for casual computer users to ascertain. To claim to be able to do all things is not enough—people need to have a reason for buying one computer over another. Many people buy PCs, for instance, in order to run *AutoCAD* or *dBASE*.

Price/performance counts for little, as well. On one hand, there are the people who ask, "Why's it so cheap? What's wrong with it?" And on the other hand, there are the people who shell out sometimes over $2,000, fully expecting that their computer will be difficult to learn and to live with. Why?

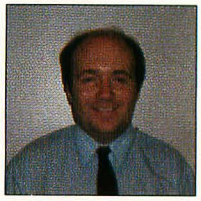
Because, somewhere along the line, someone has told them that Computer X is what they really need, that it will solve all their problems and that all computers are hard to use.

If the ST is ever to make a splash in the United States, Atari needs to find something that the ST can do well and emphasize it. Of course, the ST is already recognized as a premier music computer (and Atari is eagerly pursuing the desktop-publishing market as well), but the music market is far too specialized for the exposure Atari needs.

The business world holds the key to success, but unless something comes along for the ST that no other computer can do (such as the Mac's desktop publishing coup a while back), the ST will remain an also-ran.

Perhaps the next generation of STs, with their increased capabilities, will bring about a revolution similar to Apple's stumbling upon desktop publishing. And perhaps Atari will edge its way upwards into the high-end market areas of Sun and Apollo. Only time will tell.

Until then, Fred's question remains to haunt all of us. But the bean dip's gone.■



Todd Threadgill *is a graduate student at UCLA and is currently pursuing his M.F.A. degree in filmmaking. He lives in Sherman Oaks and is interested in writing and programming.*

■ *ST-LOG invites all authors to submit essays for possible use in the* Footnotes. *Submissions should be between 1,000 and 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. Send your submission to:* Footnotes, *c/o* ST-LOG, *P.O. Box 1413-M.O., Manchester, CT 06040-1413.*

ATARI

AUTHORIZED SERVICE CENTER FOR ALL ATARI PRODUCTS

MICROTYME

A DIVISION OF MICRO PERIPHERALS, INC.

4049-51 MARSHALL RD. • KETTERING, OHIO 45429

VISA · MasterCard · DISCOVER

ATARI

HARDWARE

ST'S. . .IN STOCK!!!

Color Monitors	CALL
Mono Monitors	CALL
SF 314 Drive	CALL
GTS 100 Drive	CALL
IB 5¼ 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

$649

OTHERS.....CALL

ICD 20 MB HARD DISK

$599

OTHERS, INCLUDING TAPE BACK-UP . . . CALL

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 (Blk)	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 ★

10th Frame Bowling	26
220 ST (Terminal Emulator)	15
3D Breakthru	26
AB Zoo	21
Advanced OCP Art Studio	31
Air Ball	26
Air Ball Construction Set	17
Algebra 1, 2, 3	ea 14
Aliants	19
All About America	41
Alt	21
Alternate Reality-The City	32
Alternate Reality-The Dungeon	32
America Cooks Series	ea 9
Demon's Winter	25
Architectural Design	19
Arctic Fox	26
Art Gallery 1, 2, 3	ea 19
Assem Pro	37
Autoduel	24
Award Maker	27
Balance of Power	32
Bally Hoo	27
Barbarian	25
Bards Tale	31
Base Two	45
Basketball (Two on Two)	26
Battle Droidz	25
Battlezone	19
Beyond Zork	34
Biology 1, 2, 3 or 4	ea 14
Bismarck	28
Black Lamp	17
Blockbuster	27
Boulderdash Construction Kit	17
Brataccas	15
Breach	27
Bridge 5.0	24
Bubble Ghost	24
Business Tools	26
Cad 3D (Ver. 2.0)	57
Calamus	175
Calamus Font Editor	62
Captain Blood	31
Carrier Command	29
Certificate Maker	25
Championship Baseball	27
Championship Wrestling	26
Chartpak	34
Chessmaster 2000	29
Chrono Quest	29
Circuit Maker 2	63
Clip Art 1, 2, 3, 4, 5, 6	ea 13
Club Backgammon	23
Color Computer Eyes	169
Colorburst 3000	25
Copyist Level 2	158
Cosmic Relief	26
Cracked	21
Crazy Cars	25
Cyber Control	39
Cyber Paint	48
Cyber VCR	45
Dark Castle	27
Data Manager ST	46
Datatrieve	33
DB Man	149
Death Sword	13
Deep Space	31
Defender of the Crown	31
Degas Elite	38
Demon's Winter	25
Desk Cart	67
Digi Drum	14
Dive Bomber	19
Dr. Drums (DR T)	19
Dr. Keys (DR T)	19
Drafix	129
Dungeon Master 2	18
Dyna Cadd	429
Easy Draw (Regular)	63
Easy Draw W/ Supercharger	95
Easy Tools	32
Elite	22
Empire	34
Expert Opinion	72
EZ Score Plus	95
EZ Track Plus	43
F15 Strike Eagle	24
Falcon ST (Low, Low)	CALL
Fast Basic	59
Fast Basic M Compiler	19
Fire and Forget	25
First Cadd 2.0	33
First Letters & Words	25
First Shapes	25
First Word Plus	59
Flash (Great!)	18
Flight Simulator 2	32
Scenery Disks	ea 18
Font Disks (Pub Part) 1-6	ea 20
Fonts and Borders	24
Fontz ST	22
Foundations Waste	26
Fraction Action	24
G + Plus	21
Gateway	31
Gauntlet	31
Genesis (Molecular Modeler)	59
GFA Basic 3.0	59
GFA Basic Book	27
GFA Companion	32
GFA Compiler	38
GFA Draft Plus	49
GFA Quick Reference Manual	12
GFA Ray Trace	59
Gladiator	25
Global Commander	28
Gold Of The Realm	24
Gold Rush	25
Goldrunner	26
Goldrunner 2	27
Goldrunner 2 Scenery Disks	ea 7
Gone Fishin'	28
Great Chefs Vol. 1, 2, & 3 Set	39
Gridiron (Football)	19
Gunship	26
Hard Disk Backup	23
Harrier Combat Simulator	34
Heroes Of The Lance	26
Hi Soft Basic Pro	109
High Roller	27
Hollywood Hijinx	19
Human Design Disk	19
Hunt for Red October	32
IB Copy	23
Impossible Mission 2	27
Indiana Jones Temple of Doom	31
Interlink ST	24
International Soccer	25
Into The Eagles Nest	24
Inventory Manager	52
Jet	33
Jinxter	27
Joust	18
Juggler (New Version)	32
Karate Kid 2	27
Karateka	23
KCS Level 2	189
Kid Progs	27
Kinderama	24
Kings Quest 1, 2, 3 or 4	ea 31
Knickerbockers	12
LDW Power Spreadsheet	89
Label Master Elite	26
Lattice C	109
Leaderboard Dual Pack	15
Leatherneck	25
Leisure Suit Larry 1 or 2	24
Lock On	24
Lords of Conquest	13
Lords of the Rising Sun	31
Macro Mouse	25
Magic Sac Plus	89
Magic Sac Roms	CALL
Major Motion	25
Marble Madness	27
Mark Williams C	114
CSD Source Debug	39
Master Cad	119
Match Point	25
Mavis Beacon Teaches Typing	32
Megamax C (Laser C)	118
Mercenary	27
Metro Cross	16
Micro Kitchen Companion	26
Microleague Baseball	36
Microsoft Write	65
Midi Draw	63
Midi Maze	26
Midi Recording Studio (DR T)	36
Missile Command	18
Mixed Up Mother Goose	21
Multi Desk	19
Music Studio 88	34
N Vision	29
Neo Desk 2	33
New Tech Coloring Book	15
Night On The Town	22
Ninja	14
Obliterator	25
Ogre	27
Oids	24
Omnires	23
Orbiter	25
Page Stream	115
Paint Pro	33
Paintworks	14
Paperboy	25
Partner Fonts	21
Partner ST	43
PC Ditto 2 Low	CALL
Perfect Match	27
Personal Pascal	66
Phantassie 1, 2 or 3	ea 26
Phasar 3.0	58
Pinball Wizard	24
Pirates of the Barbary Coast	17
Planetarium	33
Platoon	25
Police Quest 1, 2	32
Pool of Radiance	25
Prime Time	27
Print Master Plus	26
Prison	25
Pro Copy (Latest Ver.)	28
Publisher ST	79
Q Ball	21
Quantum Paint Box	31
Quink	11
Rastan	25
Read & Rhyme	24
Renegade (Outcast)	14
Road Runner	26
Roadwars	22
Rockford	22
Santa Paravia	19
Scan Art	32
Scruples	29
SDI	13
Shadow	18
Shadowgate	34
Shard of Spring	27
Shuffleboard	19
Silent Service	24
Sinbad	19
Sky Fox	14
Space Quest 1 or 2	ea 31
Space Quest 3	37
Spectrum 512	41
Spelling Bee	19
Spiderman	7
Spy vs Spy 3 (Arctic Antics)	19
ST Disk Drives Inside & Out	18
ST Gem Programmers Ref Man	15
ST Internals Book	15
ST Intro to Midi Book	15
ST Macnine Language Book	15
ST Pool	21
ST Talk Pro	17
STAC	44
STOS	39
Star Fleet 1	37
Star Raiders	19
Starglider 2	26
Stellar Crusade	36
Strip Poker 2	25
Sub Battle Simulator	25
Sundog	25
Super Base Professional	174
Super Star Ice Hockey	31
Swift Calc St	46
Take Note	52
Tanglewood	25
Terror Pods	25
Test Drive 1 or 2	24
Test Drive 2 Extra Disks	ea 14
Three Stooges	34
Thunder	26
Time Bandit	24
Top Gun	11
Trailblazer	32
True Basic	52
Tune Smith (DR T)	95
Tune Up	31
Turbo ST	32
TV Sports Football	31
Typhoon Thompson	21
Uninvited	31
Universal Item Selector	12
Universal Military Sim.	31
Vampires Empire	20
Vegas Craps	24
Vegas Gambler	23
Video Titleing	22
Vip Professional (Gem)	129
War Ship	38
Wargame Construction Set	22
Winter Challenge	11
Wizards Crown	25
Word Perfect	159
Word Up	47
Word Writer ST	46
World Games	12
World Karate Championship	19
WWF Microleague Wrestling	29
Xevious	19
Zak McKracken	27
Zany Golf	26

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.

★ ST SOFTWARE ★

HOURS: M-F 9 a.m.-9 p.m. EST
SAT 9 a.m.-5 p.m.

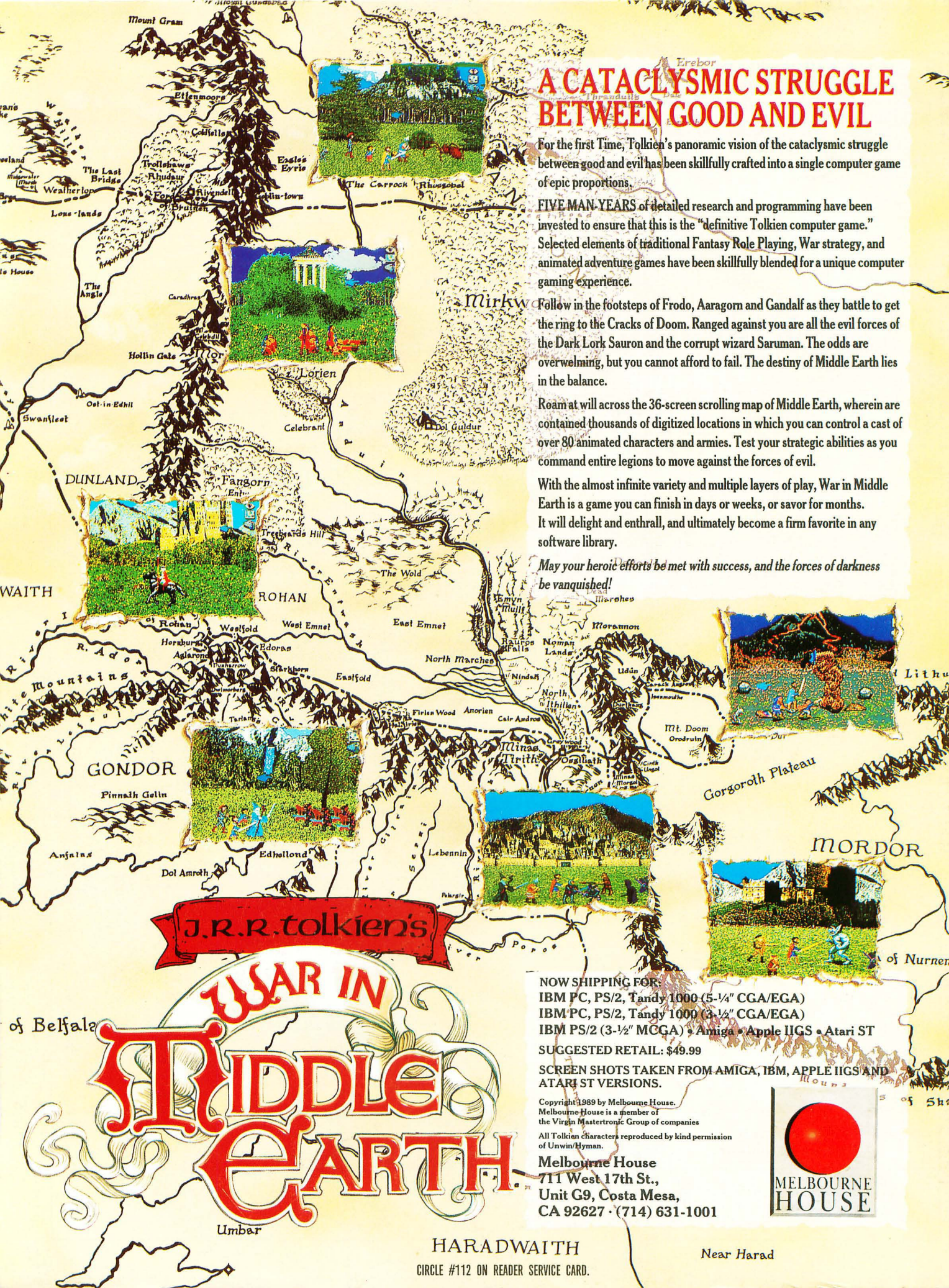
ALL 50 STATES CALL TOLL FREE
1-800-255-5835

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.

CIRCLE #109 ON READER SERVICE CARD.



A CATACLYSMIC STRUGGLE BETWEEN GOOD AND EVIL

For the first Time, Tolkien's panoramic vision of the cataclysmic struggle between good and evil has been skillfully crafted into a single computer game of epic proportions.

FIVE MAN-YEARS of detailed research and programming have been invested to ensure that this is the "definitive Tolkien computer game." Selected elements of traditional Fantasy Role Playing, War strategy, and animated adventure games have been skillfully blended for a unique computer gaming experience.

Follow in the footsteps of Frodo, Aaragorn and Gandalf as they battle to get the ring to the Cracks of Doom. Ranged against you are all the evil forces of the Dark Lork Sauron and the corrupt wizard Saruman. The odds are overwelming, but you cannot afford to fail. The destiny of Middle Earth lies in the balance.

Roam at will across the 36-screen scrolling map of Middle Earth, wherein are contained thousands of digitized locations in which you can control a cast of over 80 animated characters and armies. Test your strategic abilities as you command entire legions to move against the forces of evil.

With the almost infinite variety and multiple layers of play, War in Middle Earth is a game you can finish in days or weeks, or savor for months. It will delight and enthrall, and ultimately become a firm favorite in any software library.

May your heroic efforts be met with success, and the forces of darkness be vanquished!

J.R.R. tolkien's

WAR IN MIDDLE EARTH

NOW SHIPPING FOR:
IBM PC, PS/2, Tandy 1000 (5-¼" CGA/EGA)
IBM PC, PS/2, Tandy 1000 (3-½" CGA/EGA)
IBM PS/2 (3-½" MCGA) • Amiga • Apple IIGS • Atari ST

SUGGESTED RETAIL: $49.99

SCREEN SHOTS TAKEN FROM AMIGA, IBM, APPLE IIGS AND ATARI ST VERSIONS.

Melbourne House
711 West 17th St.,
Unit G9, Costa Mesa,
CA 92627 · (714) 631-1001

MELBOURNE HOUSE