

# SoftSide™

#32

THREE DOLLARS

## OPERATION: SABOTAGE



*William A. Chase '82*

# WHO ARE YOU THIS TIME?



Are you the seeker of lost treasures in an enchanted realm of magical beings? Perhaps you're an astronaut, thousands of light years from earth, searching the galaxy's rim for the fabulous treasures and advanced technologies of a long-dead civilization. Maybe you're the plunderer of ancient pyramids in a maddeningly dangerous land of crumbling ruins and trackless desert wastes.

## Imagine.

You wish to soar to other worlds, to behold wonders never beheld by mortal eyes, to dream unrestrained to the furthest limits of your imagination.

## Imagine now that you can.

The ADVENTURE SERIES by Scott Adams has been reviewed by **every** major microcomputing magazine, receiving only the highest praise for its mind-puzzling challenge & refreshing originality. Tens of thousands of adults and children have matched wits with the Adventure Master. Now, it's your turn. If you crave the challenge and panorama of the exotic, touched with sparkling humor, then the Adventure Series is for you.

With the incredible Adventure Series by Scott Adams,  
**YOU ARE WHOMEVER YOU WANT TO BE,  
 WHENEVER YOU WANT TO BE!**

**Adventure**  
 INTERNATIONAL  
 A DIVISION OF SCOTT ADAMS, INC.

BOX 3435, LONGWOOD FL 32750  
 (305) 862-6917 (QUESTIONS)  
**ORDER FROM YOUR FAVORITE DEALER  
 or CALL TOLL FREE (800) 327-7172**  
 (ORDERS ONLY PLEASE )

	SINGLE TAPE	12 PACK TAPE	SINGLE DISK	TRIPLE DISK	12 PACK DISK
APPLE	\$19.95 24K	\$129.95† 24K	\$29.95* 48K	\$69.95* 48K	—
ATARI	\$19.95 24K	\$129.95† 24K	—	\$39.95 40K	\$129.95† 48K
CP/M (Z-80)	—	—	—	—	\$129.95†
TRS-80 MOD 1&3	\$19.95 16K	\$129.95† 16K	—	\$39.95 32K	\$129.95† 32K
TRS-80 MOD 2	—	—	—	—	\$129.95

Adventure Hint Book (All 12) . . . . . \$ 7.95  
 Adventure Hint Sheet (Single) . . . . . 1.00  
 Commodore VIC (Cartridge) . . . . . 39.95  
 T.I. 99/4 (Tape or Disk) . . . . . 29.95  
 Requires Adventure Command Module

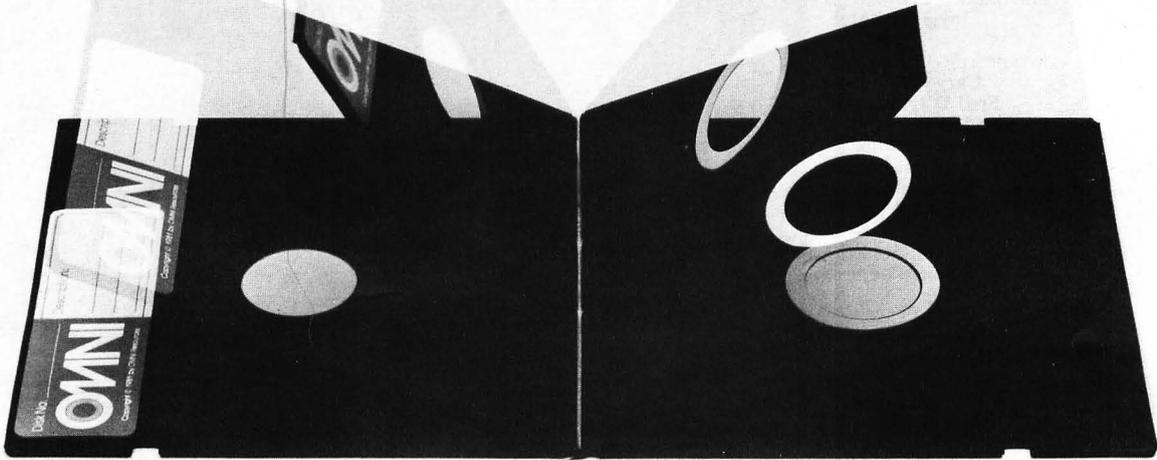
\*Scott Adams' Graphic Adventure Series  
 Apple Disks-Full Color Hi-Res Graphics.  
 †12 Adventure Hint Book Incl.

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

SHIPPING & HANDLING ARE EXTRA

# All disks have two sides.

# But only Omni's Flip/Floppy lets you use both sides in single-sided drives.



## Double your capacity.

Now you can double disk storage capacity as well as save on media costs if you have an Apple, TRS-80, Zenith, Northstar or any single-sided, 5 1/4" disk drive. Omni's Flip/Floppy lets you record on both sides . . . so you can consolidate long programs and get twice as much storage in one disk at far less cost than two conventional disks with the same capacity.

## Premium quality, too.

Like our full line of flexible disks, each Flip/Floppy is critically certified at more than twice the error threshold of your system; rated for more than 12 million passes without disk-related errors or significant wear; precision fabricated with such standard features as reinforced hub rings; and backed by a full, 90-day money-back guarantee.



### Omni Resources

4 Oak Pond Avenue, Millbury, MA 01527

Dealer inquires invited.  
Software houses:  
call for duplicating services.

## Plus

### a FREE storage case if you

order now. With every ten-pack you order (even conventional disks), you'll get a \$5.00 protective plastic case, absolutely free.

Be sure to indicate system name and model # at right.

#of		Price Per	Total
10 Packs	5 1/4" Disks	10 Pack	Price
_____	Flip-Floppy reversible	\$40.00	\$ _____
_____	Double side/double density		
	(equivalent to 20 conventional disks)		
_____	Single side/single density	\$30.00	\$ _____
_____	Single side/double density	\$40.00	\$ _____
_____	Double side/double density	\$45.00	\$ _____
	Shipping and handling		\$ 2.00
	5% sales tax (Mass. only)		\$ _____
	Total		\$ _____

## Call toll free (800) 343-7620

## In Mass: (617) 799-0197

Call if you're not sure which disk is compatible with your system. Call for prices on 96 tpi and special formats. We offer an unconditional money-back warranty.

- Check  
 COD

- Master Card  
 VISA

Card # \_\_\_\_\_ Exp. \_\_\_\_\_

System and model # \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

Tel. \_\_\_\_\_

EDITOR-IN-CHIEF  
**Randal L. Kottwitz**

SOFTWARE EDITOR  
**Bill Kubeck**

PROGRAMMING STAFF  
**Rich Bouchard**  
**Alan J. Zett**

EDITORIAL ASSISTANT  
**Joyce Smith**

CONTRIBUTING EDITORS  
**Cary Bradley**  
**Fred D'Ignazio**  
**Sheldon Leemon**  
**Lance Micklus**  
**Mark Pelczarski**  
**Joan Truckenbrod**  
**Allen L. Wold**

ART PRODUCTION MANAGER  
**Lynn Wood**

PRODUCTION STAFF  
**Lynda Fedas**  
**Denise Lafleur**

PUBLISHER  
**John G. Grow**

ASSOCIATE PUBLISHER  
**Nancy Lapointe**

ADVERTISING  
**Sue Rowland**  
**Bob Mackintosh**  
**Christopher Smith**

CUSTOMER SERVICE  
**Cindy Schalk**

DEALER SALES  
**Kathie Maloof**  
**Irene Stanton**

STAFF  
**Mary Edwards**  
**Donna Jean**  
**Steve Justus**  
**Doris Miller**  
**Cindy Zawacki**

FOUNDER  
**Roger W. Robitaille, Sr.**

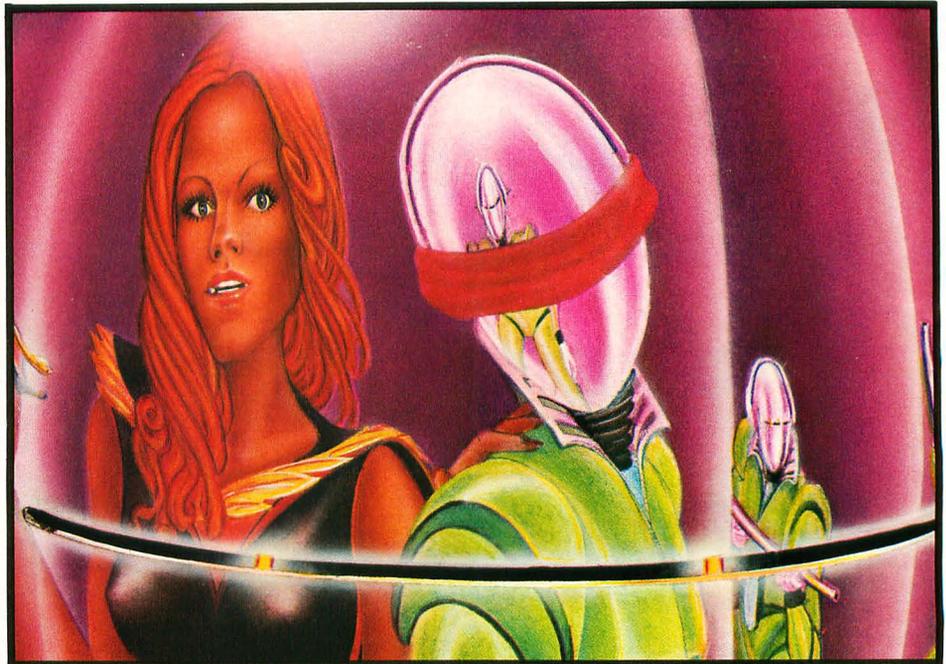
*SoftSide* Vol. 5, No. 11

SUBSCRIPTION INQUIRIES should be sent to *SoftSide* Publications, 100 Pine Street, Holmes, PA 19043.

EDITORIAL AND ADVERTISING CORRESPONDENCE should be sent to *SoftSide* Publications, 6 South Street, Milford, NH 03055. Telephone (603) 673-0585.

*SoftSide* (ISSN 0274-8630) is published monthly by *SoftSide* Publications, Inc., 6 South Street, Milford, NH 03055. Telephone 603-673-0585. Printed at Lorell Press, Avon, MA. Second class postage paid at Milford, NH, Avon, MA, and at additional mailing offices. Subscription rates: US, and Canada, \$30/12 issues. First Class US, APO, FPO, Mexico, \$40/12 issues. Other foreign countries, \$62/12 issues. Media subscription rates: US Magazine and Cassette, \$75/12 issues. APO, FPO, Canada and Mexico, add \$20/12 issues. Other foreign add \$50/12 issues. All remittances must be in U.S. funds. Entire contents Copyright © *SoftSide* Publications, Inc., August, 1982. All rights reserved. POSTMASTER: Please send form 3579 to *SoftSide* Publications, 100 Pine Street, Holmes, PA 19043.

Apple<sup>TM</sup>, ATARI<sup>®</sup>, IBM<sup>®</sup>, and TRS-80<sup>®</sup> are registered trademarks of The Apple Computer Company, Warner Communications, International Business Machines Corporation and the Tandy Corporation, respectively. Envyrn, Envyrnment, Envyrnese and diversions thru Envyrn are registered trademarks of Roger W. Robitaille, Sr.



Cover illustration by Bill Giese

## FRONT RUNNERS

29

### Operation: Sabotage

by Ray Sato

Translations by Ron Shaker and Rich Bouchard

Encryption modifications by Rich Bouchard, Bill Kubeck and Alan J. Zett.

In another of our encrypted adventures, you'll be challenged to infiltrate an alien installation on Mars. They are planning to build an incredible defense shield and you must steal the plans and destroy the complex. Good luck — you may get out alive.

47

### C.A.T.S. (Parts II & III)

by Jon Voskuil

Translations by Alan J. Zett

With this month's completion of the Computer-Assisted Testing System, you'll be able to administer the tests you created with the first module, score the students on their responses and keep accurate records for future reference.

*SoftSide*

## FEATURES

14

### Sri Lanka Imports At the NCC

by David D. Busch

A tongue-in-cheek look at a fictitious company's introduction of such unique products as a black phosphor monitor at the National Computer Conference.

17

### A Few Words From the Phantom Programmer

Our mysterious contributor offers a few words of advice on the use of I and O, (Or is it 1 and 0?) as variables in program listings.

18

### Anatomy of an Adventure

by Peter Kirsch

The creator of many of *SoftSide's* Adventures of the Month will take you step by step through the process of writing an original adventure and translating it to other systems. He's included his own "Adventure Skeleton" in this valuable tutorial.

22

## The National Computer Conference — What's a Mainframe?

by George Blank

This year's NCC revealed that the microcomputer is a contender for almost all serious computing applications. Here's a comprehensive report on the introduction of systems from such unexpected manufacturers as Sony and Epson.

25

## Entertainment Tomorrow

by Fred D'Ignazio and Allen L. Wold

Disney Studios' *TRON* has brought a new age of animation to the film industry. The authors give an extensive background on the computer graphics companies responsible for these developments accompanied by some exciting scenes from the movie.

96

## Machine Head

by Spyder Webb

---

## DEPARTMENTS

---

5 Editorial

6 Input/Output

9 Hints and Enhancements

11 Calendar

64 General Information

Concerning *SoftSide* Line Listings, *SWAT* and Media Versions

93 Market/Side

94 New Products

96 Advertisers Index

---

## APPLE™/SIDE



66 Enhanced Disk Version

**CLOZE TEST** by William D. Hedges

The Classroom Teacher's Diagnostic Reading Test enables the teacher or parent to determine whether a student is able to read materials with understanding. It is also a valuable self-improvement tool for reading comprehension.

72 Article

**APPLE DISKOURSE** by Cary W. Bradley

You're on your way to having a valuable disk utility with this installment of the series — *Disk Snooper*. It will allow you direct access to the secrets hidden away on your diskettes.

67 Review

**WIZARDRY — Scenarios I & II**

Reviewed by Hartley G. Lesser

---

## ATARI®/SIDE



78 Enhanced Disk Version

**NEAT LIST** by Frank Roberts

This useful listing utility will make the line output of your programs much easier to understand. Statements will be listed on individual lines and REMarks will be framed by asterisks. Debugging will never be the same.

79 Article

**ATARI® SILENCER** by John J. Anderson

If you or your family are annoyed by the constant output of the speaker on your ATARI® console, here's the solution. With only a few inexpensive parts, you can put a switch on that speaker so you are in control.

82 Review

**DEADLINE** Reviewed by Eric F. Wolcott

---

## TRS-80®/SIDE



84 Enhanced Disk Version

**UP PERISCOPE** by Ron Potkin

Exciting sea-battle action comes to the screen of your TRS-80®. This wargame/simulation is packed with destroyers, submarines and a convoy of vessels to keep two players fast on the keys in dramatic competition.

Reviews

87 **ALIEN DEFENSE** Reviewed by Andre Chen

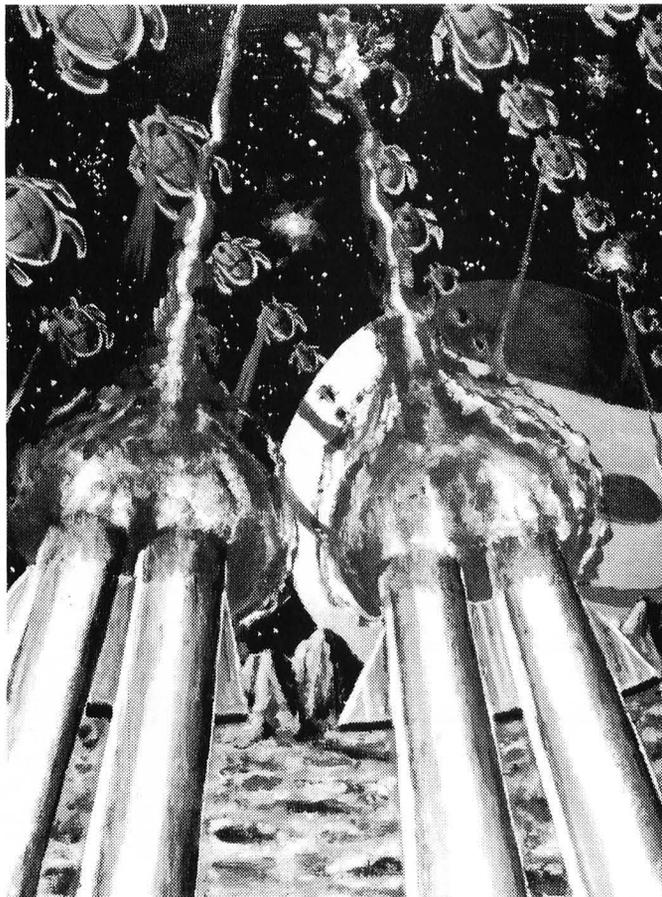
88 **NEWSSCRIPT 7.0** Reviewed by Harry Temple

92 **THE TAS ADVENTURE SYSTEM**

Reviewed by Mark E. Renne

# MOONBASE IO

● the battle for the moons of Jupiter ●



Blast your way through the alien mine fields! Defend the moonbases from an attacking alien armada! It won't be easy. To win you will need to mount a heroic assault on the alien mother ship.

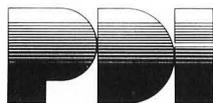
**Moonbase Io** combines three exciting arcade adventures in one exciting game. The machine-language program by John Konopa uses advanced graphics and sound effects. Action is fast and exciting — varying levels of skill required to go from one part of the game to the next.

Requires 24K ATARI™ computer with disk and cassette.  
Cassette version available soon.

BEYOND SOFTWARE's arcade-adventures are the next generation in computer games for the ATARI™ computer.

A voice-activated program will help you meet and overcome the challenge — it may be an alien invasion, a fiendish murderer preying on a country village or a treasure trove buried deep in the shark-infested sea.

**Moonbase Io** is available at fine computer dealers. Or, directly from PDI for \$29.95 plus \$2.00 shipping and handling.



Program Design, Inc., 11 Idar Court, Greenwich, CT. 06830 203-661-8799

ATARI™ is a registered trademark of Atari, Inc.



# Microcomputers Reach the Age of Adolescence

by Randal L. Kottwitz

For several years now, we have been speaking of the microcomputer industry as being in its infancy. Businesses and consumers alike have been "getting their feet wet" in order to be prepared for the infant to grow up. Indeed, many consumers are still looking at the purchase of computers such as the Timex/Sinclair as a way of finding out whether computers are for them. However, those who saw the potential impact of this technology several years ago are now the people producing the machines and software to truly utilize the technology from a more current than futuristic point of view. Today's computer buyer can be enticed by what the machine can do for him when he gets it home rather than on the basis of what it can do for him when the proper software has been developed. It's been said that VisiCalc® has sold more Apples™ than Apples™ have sold copies of Visicalc®. It would not be inappropriate to say that the industry has progressed beyond its infancy, into adolescence.

The point of this observation? As a human being suffers from growing pains and confusion during his adolescence, so is the microcomputer industry suffering similar symptoms. The software developers who have been working with the current popular systems have now progressed to a point that they are more aware of the limitations of those systems than they are awed by potential applications. This situation would be a source of only minor frustration if no solution existed. However, their consternation is doubled as yet another generation of microcomputers takes its position in the marketplace, replete with answers to their frustration. Developers who, several years ago, were awed by the potential of 48K RAM and mini-floppy disk storage are now crying for more RAM, more storage and more speed — all available as 64K RAM chips, hard disk storage and 16-bit microprocessors become an affordable reality.

It's easy for us at *SoftSide* to observe the progression of software authors. When *SoftSide* began publishing in 1978, it was rare that we saw a program submission which required more than 4K. Today, it's becoming increasingly difficult for us to find quality programs which utilize less than 32K. In the same vein, our authors are becoming more and more oriented to the utilization of disk storage. As an author matures in his knowledge of the marvel of technology behind his keyboard, how can he help but want to make use of its every capability? Although we are acutely aware of the number of people who have just spent their last available cent to purchase a 16K, cassette based system, it's become impossible for us to make every piece of software we publish conform to that standard. Indeed, it would be unfair to try to satisfy all of our subscribers with software designed for the lowest common denominator of the systems we support. Our answer is to attempt to provide "something for everyone." However, we cannot be sure how long that can remain a reality — especially as RAM upgrades and disk drives become more and more affordable. You may wish to take that as a foreboding of the future direction of *SoftSide*. That it is, but more importantly, it is an observation on the future of the microcomputer industry as a whole. One need only look at the specifications of computers being brought to market by such consumer oriented companies as Sony (see *The National Computer Conference — What's a Mainframe?* elsewhere in this issue) to realize that 64K RAM and disk drives as an entry level system configuration is soon to be the standard.

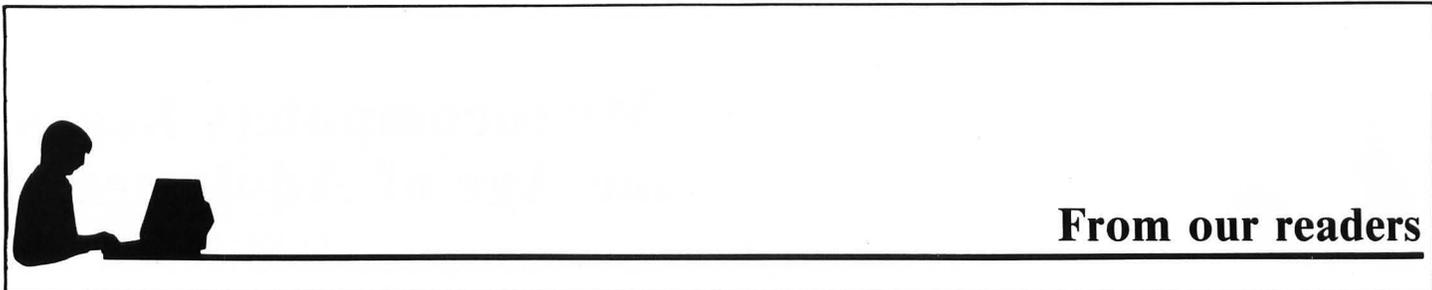
I was told when I bought my first microcomputer (48K, interface, printer and a disk drive) that I was being foolish if I thought it would be the last computer I would purchase. I agreed at the time that this was certainly a racing industry and its technological growth

would dictate future upgrades. Little did I know how soon I would be hungering for a larger system with better sound, color and graphics capabilities. I'm sure there are many *SoftSide* readers who are having the same feelings of frustration.

Take a moment and peer through the looking glass at the world of tomorrow with me. The frustration with which we are now coping will seem insignificant as we attempt to determine which of the central information services we wish to enter our homes. No longer will the memory size or storage capabilities of our system have any meaning, for every piece of information we could possibly desire will be available through a cable connection — awaiting only the touch of a few keys to summon it to our screen. There will be a great deal of this information even more readily available as images of most of the books and artworks of the world will reside on the small collection of laser discs on our library shelves. Just imagine when one of our grandchildren asks, "What's a mini-floppy?"

There's no doubt that the rate of change in the microcomputer industry is experiencing exponential growth. It is truly comparable to the rate of change a human being experiences in his adolescence. There is one comforting thought as we wonder about the future of this field we've chosen to join — somehow, each of us came out of our own adolescence more mature and refined. The home computer industry will emerge from this turbulent period of change with something of the same results — better directed with a more complete understanding of its purpose. ☺

Randal L. Kottwitz  
Editor-in-Chief



## From our readers

### INPUT

#### Alaskan Apples™

Dear *SoftSide*,

I only recently received your flyer requesting extended subscriptions. Due to the unavoidable delays associated with weekly mail planes here in the bush of Alaska, I have been unable to meet your deadline of May 30. I hope you will still honor the commitment I would like to make to *SoftSide*. I think it is simply the best on the market for home computer hacks! For us, pounding away on our little Apples™, hooked up to a 15 KW diesel generator, the enjoyment and diversity of software we receive each month is more than worth the subscription price.

Keep up the good work.

James R. LaRiviere  
Lime Village, AK

#### Australian ATARI®'s

Dear *SoftSide*,

A new Australian users' group for ATARI® owners was formed in January of this year, with the goals of promoting the ATARI® 400/800 Home Computer System, instructing both beginners and advanced users in programming techniques, exchanging hints, tips and ideas amongst members, and generally enjoying ourselves.

Meetings are held at six p.m. on the first Monday of every month (or the second Monday if it clashes with a public holiday) at the offices of:

I.P. Sharp Associates  
8th Floor, Carlton Center  
55 Elizabeth Street  
Sydney (between King Street and  
Martin Place)

Meetings are not restricted to ATARI® owners. Interstate ATARI® owners are also invited to write for the names of contacts in their state.

The group is called *ATARI® Computer Enthusiasts (N.S.W.)*. We are loosely affiliated with *A.C.E.* in the U.S.A., which has several branches throughout North America and membership worldwide.

Membership to *A.C.E. (N.S.W.)* is \$15 joining fee and \$15 annual subscription in common with other user groups. This is

reduced to \$10 joining fee and \$10 annual subscription for students under 18. (All prices in Australian dollars.)

Subscriptions or postal enquiries may be directed to:

*ATARI® Computer Enthusiasts (N.S.W.)*  
78 Ayres Road  
St. Ives, N.S.W.  
AUSTRALIA 2075

Phone enquiries: Garry Francis  
(02)2-0933 Ext. 354 (B.H.), (02)789-1397  
(A.H.) or Paul Phillips (02)449-6286  
(A.H.).

Garry Francis  
Earlwood, N.S.W., AUSTRALIA

#### ATARI® SWAT

Dear *SoftSide*,

The *SWAT* program in issue 30 is a great idea. It should eliminate a lot of the frustration those ever-present typos can cause.

One curious thing, though. I typed in the *Dungeon of the Gods* program (ATARI® version) and ran the *SWAT* program on it. I found that, while the line numbers and length numbers matched your listing, not one of the *SWAT* codes I generated was the same as your table. Yet, the program seems to run fine...strange.

Now a request. How about *SWAT* tables for programs from previous issues? I still have a few programs which need debugging, but so far have defied all my efforts. Why not publish tables in an upcoming issue for, say, one entire previous issue? Or perhaps you could make tables for a year of *SoftSide* available for separate purchase.

At any rate, keep up the good work.

Alan Varner  
Old Greenwich, CT

#### SWAT From the Past

Dear *SoftSide*,

Once upon a time way back in October 1980, I received through the mail an absolutely free, try before you buy, kick the tires, slam the doors, try it you'll like it, bet you can't live without us, complimentary copy of *SoftSide*. A few hunt and pecks later and I was HOOKED! So, out came the old checkbook and my addiction was established.

It was a bleak, lonely Fall and cold, bitter Winter of agony as I daily checked my mailbox for my first "bought and paid for" issue to arrive. Finally, as the winds began to howl in March, the mailman wheeled in with a flourish, lurched to a stop, lept from his vehicle, and raced to my door with the long coveted first issue. Never before or since in the history of the post office has such diligent care been given to so important a document as this.

It was with bated breath that I nervously sat and turned the first few pages to the table of contents to whip up an appetite for the feast I knew would soon follow. Imagine for a moment my extreme pleasure when I discovered not only were there four good programs for my TRS-80® in that issue, (more than enough, at my slow hack rate, to last until the next issue arrived) but two of the programs utilized sound routines — that extra ingredient that pulls more of the senses into the interaction of man and machine.

Over the next few days I somehow managed, despite obligations to work, family, meals, sleep, etc., to get *Strategy Strike* typed in. At last, the golden moment had arrived when, poised hesitantly over the keyboard, I could type those three little letters R, U, N.

Now, my fingers are not all that misshapen or overly large, nor is my keyboard unusually small, but somehow the two adjoining letters U and I were hit sequentially, followed immediately by an N and ENTER while typing that simple three letter word, so that I had inadvertently typed "RUIN". Thus, a new BASIC command was born and instantly understood by my computer. For, although the computer responded with the usual SYNTAX ERROR, I suspect some internal skulduggery because, to this day, the program doesn't work quite right.

Each month, I eagerly await the next issue in hopes of seeing the necessary corrections to make this baby fly, but so far, not so good. I've dumped it to the printer and checked line for line against the published versions, made some fixes, redumped, rechecked, etc., but still just can't discover what's wrong.

Having just received the new, non-dated format, Issue 30, I discovered what might be a possible solution to my problem. Your

continued on page 8

# GOODBYE MARY LOU, HELLO MICROCOMPUTER

Sorry 'bout that, Mary Lou!

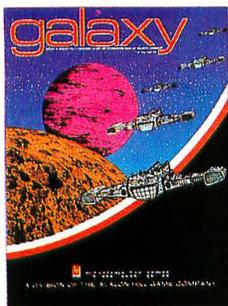
But your boyfriend has a new PET®. In fact, he may even have an Atari®, Apple II®, TRS-80®, or TRS-80® Color Computer, too . . . any of which plays an Avalon Hill Microcomputer Game.

Here's what's BRAND NEW from Avalon Hill:



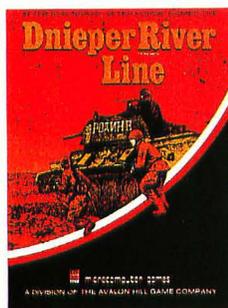
## VOYAGER

A solitaire computer game that challenges the human player to explore the four levels of an alien spacecraft's maze-like corridors and rooms in 3-D simulated graphics, all the while avoiding robots programmed to blast any intruders. In order to win, the human must destroy all power generators and escape or hunt out and annihilate the killer robots. **VOYAGER** comes with color-animated graphics and sound capabilities for computers so equipped.



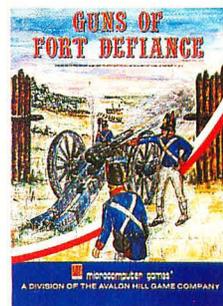
## GALAXY

Have you ever wanted to conquer the universe? In **GALAXY**, players send their galactic fleets out to explore and conquer the universe, solar system by solar system. The planets discovered may be barren worlds or they may possess immense industrial capacity and defensive ships to resist colonization. **GALAXY** comes with sound effects (for computers with sound capability)



## DNIEPER RIVER LINE

A fictionalized engagement between the Russian and German forces in the southern Ukraine in 1943. The game challenges you, the German commander, to repel Russian efforts to breach the Dnieper River defensive positions. Soviet units, controlled by the computer, seek to overrun the thin German line and capture sufficient objectives to attain victory. **DNIEPER RIVER LINE** has four levels of difficulty and comes complete with over 300 illustrated counters and a mounted mapboard.

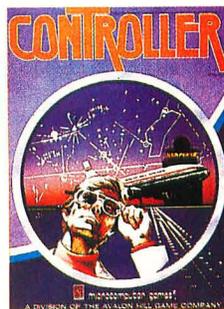


## GUNS OF FORT DEFIANCE

In this exciting arcade game, you are the commander of a 19th century artillery piece in a besieged stockade. For each shot you must specify a type of ammunition—ball, canister, shell or spherical case—and fuse length (if applicable), and set the elevation and deflection of the cannon. The computer controls the enemy forces, randomly attacking with cavalry, infantry or another artillery piece.

## CONTROLLER

A **real-time** simulation of air traffic control in which you will have to guide the approach and landing sequence of up to 8 aircraft. There are three types of aircraft: Light Planes, Airliners, and Private Jets, with each type having a different rate of climb, turning ability, stall speed, ceiling, fuel consumption and fuel capacity. **CONTROLLER** transforms your microcomputer screen into a realistic "radar scope"; also, each aircraft's heading, velocity, and altitude is continuously displayed on a separate chart next to the radar scope.



## COMPUTER FOOTBALL STRATEGY (not shown)

Thrilling computer version of Avalon Hill's famous board game. Based on the award-winning Sports Illustrated game of professional football.

Call Toll-Free For Credit Card Purchasing:  
Ask for Operator SS. 1-800-638-9292

SOFTWARE GAMES	With Cassette For:				With Diskette For:				PRICE	
	TRS-80 Color	TRS-80 Models I & III	APPLE II* +	PET CBM	ATARI 400/800*	TRS-80 Models I & III	APPLE II* +	ATARI 400/800*		IBM PC
*Computer Foot. Strat.						32K				21.00
*Controller							40K			30.00
*Galaxy		16K	16K	16K	16K					20.00
						32K	48K	32K	48K	25.00
*Guns of Ft. Defiance		16K	32K	16K	32K					20.00
						32K	48K			25.00
*Voyager	16K	16K	32K	32K	24K					20.00
							48K	32K	48K	25.00
*Dnieper River Line		32K	32K	32K	48K					25.00
						32K	48K	48K		30.00
*Foreign Exchange		16K								20.00
	(NOT SHOWN ABOVE)									

Avalon Hill  
Microcomputer Games, Dept. SS  
4517 Harford Road  
Baltimore, Maryland 21214

Send me the games circled on chart at left. Specify quantity, and cassette or diskette for your type of computer. I add 10% for postage.

NAME \_\_\_\_\_

STREET \_\_\_\_\_

CITY \_\_\_\_\_

STATE \_\_\_\_\_ ZIP \_\_\_\_\_

SWAT program could just save the day if only you'd publish a SWAT table for each program in past issues. I know that would require an entire issue in itself, so how about making SWAT tables for past issues available by mail to your subscribers for specific programs of interest to them?

In conclusion, I'd like to compliment you once again for an excellent publication with what most of us want — programs and sound programming advice, not "mumbo-jumbo" in obscure computerese way over the heads of most of us hackers.

Art Prence  
Flagstaff, AZ

substantial enhancements as well. Therefore, were we to publish SWAT tables run on those programs at this time, they would not necessarily accurately correspond to the code you see printed in your issue. We're looking into the problem more thoroughly as we do feel such a tool would be valuable. A possible solution would be to engage in a program of publishing not only the SWAT tables, but a combined listing of all the bug reports and enhancements we've published for a given program from the past as well.

Due to the unique manner in which ATARI® BASIC stores variables, there has been a great deal of consternation over the ATARI® version of SWAT. The program and SWAT tables published in issues 30 and 31 are correct. The solution to the problem many of you have been having is to LIST the program to be SWATed to tape or disk and then ENTER it back into memory before appending SWAT. See the *General Information* page in this issue for further details.

In many cases, when *SoftSide* reviews a product for one of the systems we support, versions of the same product are available for one or more of the other systems. With this issue, we will start a cross-referencing system to let you know that there's a review in one of the other "Sides" that pertains to a product available for your system. Be sure to look at the first page of the review section of the "Side" for your computer to see if there are other reviews in the magazine which would be of interest to you.

At the recent *Applefest* in Boston, all visitors to the *SoftSide* booth were encouraged to enter a free drawing for a one year subscription to *SoftSide DV*. At the conclusion of the show, we drew the name of Stuart Lipman of South Windsor, CT from the hundreds of entries. Mr. Lipman uses his Apple II™ computer at the office and had succumbed to the pressure from his wife and children to bring the computer home only a few days before *Applefest*. Now, not only will his family enjoy the art of computing at home, but, they'll have a monthly source of software from *SoftSide DV* as well.

We have quite an issue planned for you next time in *SoftSide*. We'll be celebrating our fourth anniversary and the subject for this special issue will be microcomputer graphics. I am more than happy to announce that we will be publishing the first in a series of program modules for all three systems presenting Envyrn™, the graphics oriented database manager we introduced for the TRS-80® in our anniversary issue last year. This will be a continuing series of programs, much along the lines of our *Developing Database* series. We'll be publishing extensive tutorials on the use of each of the modules. When you're finished, you'll not only have a dynamic tool, but an extensive background on how to use it as well. In addition, we'll have reviews of the latest graphics packages available for your computer, and an article on the philosophy and techniques of "electronic art." Until then, happy hacking!

## OUTPUT

by Randal L. Kottwitz

Once in every few issues, *SoftSide* publishes a program or article which causes a great stir among our readers and fills the editorial department's mailbox for weeks. SWAT has done just that. The letters you see above are only a sampling of the many we've received requesting SWAT tables for our programs from past issues. Regrettably, we are not able to do so at this time. We do our best to keep our master disks for duplication updated with any bug reports we receive from our readers and any

## GET SERIOUS . . .

Uncompromised design delivers superior quality and reliability. Today's latest technology allows your Atari 400 to run up to 50% cooler and provide truer video clarity. We guarantee it.

So let's get down to business.

## 48K RAM for the ATARI 400



Send certified check or money order. Visa and Mastercard welcome. N.Y. residents please include sales tax. Dealer inquiries invited. Atari is a registered trademark

In USA - 3648 Southwestern Blvd., Dept. S Orchard Park, N.Y. 14127 Tel: (716) 662-7219

In CANADA - 2 Robert Speck Parkway, Suite 1500-S, Mississauga, Ontario L47-1H8 Tel: (416) 273-6820



# HINTS & ENHANCEMENTS



From our readers

## APPLE™ SOLITAIRE JOYSTICKS

Here is an enhancement for the Apple™ version of *Solitaire* (May, 1982) to use joystick instead of keyboard control. This speeds up game play and allows you to enjoy several hours of "just one more game."

Lines 1330 through 1390 were rewritten to read paddles 0 & 1 and pushbutton 0 & 1 instead of the GET A\$. Line 1391 was added to go along with the reversed logic in line 1390. Keyboard read and clear were added at lines 1400 and 1415 for end of game. Finally, line 1460 was revised to display the joystick movement diagram.

```
1330 RL = PDL (0);PD = PDL (1)
1340 IF RL > 190 THEN GOSUB 200
      : GOTO 1330
1350 IF RL < 50 THEN GOSUB 240:
      GOTO 1330
1360 IF PD < 50 THEN GOSUB 280:
      GOTO 1330
1370 IF PD > 190 THEN GOSUB 410
      : GOTO 1330
1380 IF PEEK ( - 16287) > 127 THEN
      GOSUB 100: GOTO 1330
1390 IF PEEK ( - 16286) > 127 THEN
      1392
1391 GOTO 1400

1415 POKE - 16368,0

1460 HOME : PRINT "B1 = TO FOUND
      ATION ^ PICK UP CARDS
      MOVE LEFT <--> MO
      VE RIGHT B0 = NEXT CARD
      V DROP CARDS E =
      END GAME";
```

Robert J. Hayosh  
Farmington, MI

## APPLE™ SOLITAIRE SHUFFLING

I have immensely enjoyed the *Solitaire* program published in the May issue, and have one suggestion for improving it. The

author is to be commended for including a randomizing routine in the Apple™ version, since the Apple™ does not do this on its own. However, as the program is written, only 65536 different randomizations of the deck of cards are possible, since this is the number of possible seeds for the random number generator (in line 1220 of the main program).

This may not seem like a very great restriction; in fact, I don't believe I've played quite that many games yet myself! But it does exclude a rather large number of possible shufflings, when you consider the fact that 52 cards can be shuffled in 2 times 3 times 4 times 5 times...(etc.)...times 52 different ways. That works out to something over 8 times 10 to the 67th power (an 8 followed by 67 zeros).

A number of methods of increasing the randomization factor would be possible, but the simplest might be to avoid re-seeding the random number generator each time the deck is shuffled. If it were seeded only upon running the program for the first time, then, at each sitting, one of 65536 possible *series* of shufflings would be initiated, each series being unique. This can be accomplished by adding a line #5 to the main program, which would be exactly the same as the present line 1220, and deleting line 1220 (in other words, moving line 1220 to line 5); and then making the following changes to lines 1396 and 1440:

```
1396 TEXT : HOME : IF A$ < > "N"
      THEN CLEAR : RESTORE : GOTO 1110
1440 TEXT : HOME : CLEAR :
      RESTORE : GOTO 1110
```

To enhance the possibilities still more, one could extract an additional random number from location 78 or 79 each time the deck is shuffled, to generate a few dummy random numbers before the deck is actually shuffled. The following addition would take a few more seconds of initialization, but would gain still greater randomization:

```
1225 RN = PEEK(78) : FOR I = 1 TO RN
      : J = RND(1) : NEXT
```

Jon Voskuil  
Milford, NH

## ATARI® MICROTTEXT PAGE NUMBERING

I'm writing to thank you for your publication of *Microtext 1.2* and to add another useful feature for ATARI® users.

If lengthy manuscripts or documents are composed on the word processor, it is often useful to have pagination (numbering of the pages) during the printout. This feature can easily be added to *Microtext 1.2* with the following additions. During multipage printing, if the operator selects an option, page numbers will be inserted at the bottom of the page beginning with the second page of the manuscript.

I hope others will find this modification useful and I look forward to seeing other enhancements to the program.

To add pagination, make the following additions:

```
LINE 7035 ? : ? "Pagination? (Y for yes)"
LINE 7036 INPUT S$$
LINE 7037 IF S$$ = "Y" THEN
G = 1 : E = 0
LINE 7154 IF G = 1 THEN FOR W = LIN
TO 58 : LPRINT " " : NEXT W
LINE 7155 IF G = 1 THEN
E = E + 1 : LPRINT " " : E
LINE 7614 IF G = 1 AND LIN > 59 THEN
E = E + 1
CHANGE LINE 7615 IF G = 1 AND
LINE > 59 AND E > 1 THEN
LPRINT : LPRINT " " : E : LIN = LIN + 2
LINE 7616 IF G = 1 AND LINE > 59 AND
E > 1 THEN FOR J = 1 TO 66 - LIN :
LPRINT " " : NEXT J : LINE =
0 : GOTO 7620
LINE 7617 IF LIN > 59 THEN FOR J = 1
TO 66 - LIN : LPRINT " " : NEXT J : LIN = 0
```

Don't forget to DIMension S\$\$ in line 125.

Again, thank you, and I'll look forward to future *SoftSides*.

Bruce Tanner  
Okemos, MI

## ATARI® DELETE LINE FUNCTION FOR MICROTTEXT

My compliments to Jon Voskuil for his program, *Microtext 1.2*, which appeared in

# GO FOR IT!!

Have you ever written a program that you wanted to sell...

but didn't because you were afraid it wasn't good enough? Well, I ask you, just what is "good enough"? If you're submitting an arcade game to a commercial software house, "good enough" means machine language, high speed, flashy

graphics, and fancy copy-protection. In that league, "good enough" is downright exotic. On the other hand, if you're submitting to **SoftSide**, "good enough" is a little closer to home. Not that we'll publish just anything, mind you. We do have firm stan-

dards and the programs we accept must be well-written, interesting, and informative. It's just that we don't have to come up with Super Raster Invader Man every month, and if we don't, you don't. So your program may just be "good enough." If it is, we'll pay

for it, so why not let us take a look at it? You'll find submission guidelines listed in this issue. Go for it. It's easier than you think. Write to:

**SoftSide Publications**  
Dept AG1  
6 South Street  
Milford, NH 03055

the April, 1982 issue. After typing in all of the ATARI® version, I found that everything worked fine except the DELETE LINE function of the text editing subroutine. Changing line 9220 to the following gives neat removal of the intended line, followed by the upward movement of the text below.

```
9220 POSITION 2,V1:PRINT $$:POSITION 2,V1:IF EL LN-1 THEN FOR J=EL TO EL+X:PRINT $$:PRINT "(ESC)(CTRL)-(UP)"; T$(LP(J-1)+1,LP(J)):NEXT J
```

One other point — *Microtext 1.2* is loaded with small whole numbers; i.e., 0, 1, 2, etc.. I saved over 1800 bytes by changing the small whole numbers used repeatedly to variables; i.e., N0=0, N1=1, N2=2, etc., and replacing the numbers with the corresponding variable throughout the program. (Caution: Your ATARI® may "go to sleep" during such extensive editing!)

David G. Bick  
Loudonville, OH

## ATARI® Variable Table

Anyone who has had to squeeze a program into as little memory as possible on an ATARI® has probably found out that the variable table sometimes clings onto variables that are no longer used by the program. I have recently run across a very easy method for finding just what variables are in the table without using a large program to find and list them. Instead, you can simply use the command SAVE "E:" and the computer will attempt to put the program to the editor in the same way it would to a cassette or disk, variable table first. After typing in the command, allow the screen to fill up half-way and then hit the break key. Beginning somewhere in the first line of the display, you will see the names of the program's variables, all with the last (or only) character in inverse. In the case of a string or array, that character will be either the \$ or ( in inverse.

Ken Stailey  
Springfield, MA

## TRS-80® Model III MICROTTEXT

*Microtext 1.2* by Jon R. Voskuil, as published in *SoftSide*, April, 1982, easily lends itself to upper and lower case writing without the use of the "@" symbol, since the Model III has lower case built in. With the changes listed below, the program works very well on the Model III with no noticeable delays in reading Control Characters. Delete line 530 and change the following lines:

```
2050 IFC=1020RC=70THENRETURN
2100 IFC=1140RC=82THENGOSUB3000:GOTO200
2200 IFC=1150RC=83THENGOSUB4000:GOTO200
2300 IFC=1080RC=76THENGOSUB5000:GOTO200
```

```
2400 IFC=1130RC=81THENEND
2500 IFC=1120RC=80THENGOSUB7000:GOTO200
2600 IFC=1010RC=69ANDLN>1THENI=LN-1:PP=P:GOSUB9000:GOSUB3000:GOTO200
3130 X=ASC(X$):IFX=1010RX=69THENGOSUB9000:GOTO3000
```

These changes allow the use of lower case characters and the calling of the various control subroutines without having to use the shift key to capitalize the control code after pressing CLEAR. However, in the edit mode, the "D" and "X" must be shifted. I decided not to make this change in order to make it more difficult to accidentally delete text.

F. Keith Byrum  
Englewood, OH

## TRS-80® TITAN

*TITAN (SoftSide, December 1981)* is really great! We enjoy all the action and appreciate the chance to use some gray matter while playing. So many games depend only on the luck of the draw — **Booring!** We discovered what we think is a bug in the TRS-80® version. In a two player game, the ending status scores were 11 and 5 for "Actaeon" and "Bellona," respectively, yet the mining contract was awarded to "Bellona." I investigated the end-of-game phase. It appears that the intentions of the authors were to award the contract based on the status points. If there was a tie the award would be made on the basis of an algorithm which takes into account veins, efficiency, manpower, and credit. If that is so, there is an error in the code which computes the winner. We have changed the code between lines 8530 and 8590 to the following:

```
8522 ' Compute winning score (Z). Find
number of players with that score (Y).
If only 1, find out who and print.
8523 ' If more than 1, use algorithm to
recompute status. Repeat sequence until
only one winner, then print him.
8530 Z=-1:Y=0:FORX=1TOPL:IFK(X,0)>ZTHEN
Z=K(X,0)
8540 NEXTX
8550 FORX=1TOPL:IFK(X,0)=ZTHENY=Y+1
8560 NEXTX
8565 IFY=1THENGOTO8566ELSEGOTO8570
8566 FORX=1TOPL:IFK(X,0)=ZTHEN8590
8567 NEXTX
8570 FORX=1TOPL:IFK(X,0)=ZTHENGOSUB8580
8571 NEXT X:GOTO 8530
8580 K(X,0)=K(X,16)*100+K(X,15)+K(X,14)+
K(X,13)+RND(9):RETURN
8590 PRIN@960,"The supervisor of";MID$(
A$, (X*10)-9,10); " has won the right mine
TITAN!";:GOTO8700
```

G. Butler  
Amherst, NH ☉

# CALENDAR

**September 1-3**

**Indiana Computer Expo (ICE)**  
Indianapolis Convention Center, Indianapolis, IN

This exposition will feature exhibits of computer software and computer-related materials and services.

Contact: Ernie Kerns and Associates, Suite 201, 2555 East 55th Place, Indianapolis, IN 46220, (317)259-8111.

**September 11-12**

**New Jersey Microcomputer Show and Fleamarket**

**Holiday Inn North at Newark International Airport, Newark, NJ**

This show will include over 50 commercial exhibitors and 200 fleamarket sellers. (The computer fleamarket will be held indoors in case of rain.) Featured will be hardware, software, and accessories for all popular systems, including Apple™, TRS-80®, ATARI®, Pet™, Heath/Zenith, ZX-80/81, S-100, IBM® and others.

Contact: Gengore Corporation, 3001 Rte. 27, Franklin Park, NJ 08823, (201)297-2526.

**September 16-19**

**Applefest/Minneapolis**  
Minneapolis Auditorium and Convention Hall, Minneapolis, MN

**September 16-19**

Applefest is a conference convention and exposition. The exposition features Apples™ and Apple™ compatibles including computers, software, peripherals, accessories and publications. Admission is \$5 for adults.

Contact: Northeast Expositions, 822 Boylston Street, Chestnut Hill, MA 02167, (617)739-2000.

**September 23-25**

**Computer Showcase Expo**  
New York Coliseum, New York, NY

**September 30 — October 2**

**Computer Showcase Expo**  
Brooks Hall, San Francisco, CA

Computer Showcase Expos are a series of regional public shows for entry level computerists and users of small business and personal computer systems.

Contact: The Interface Group, P.O. Box 927, 160 Speen Street, Framingham, MA 01701 (617)879-4502. Outside Massachusetts, (800)225-4620.

If you or your organization are sponsoring or know of an event you think would be of interest to *SoftSide* readers, please send complete information to:

*SoftSide Publications*  
Calendar Editor,  
6 South Street,  
Milford, NH 03055.

Be sure to include complete information concerning dates, location, subject matter and a contact name, address, and phone number. ☺

# Translation of the Month Contest



## You could win a FREE *SoftSide* DV or CV Subscription!

*SoftSide's* Translation of the Month has been so well received by our readers, we're offering a greater author incentive than ever before. No, we can't give you a job at the U.N., but we will award a one-year subscription to *SoftSide DV* or an 18-month subscription to *SoftSide CV* for a high-quality translation of one of our past programs. That's a value of \$125 for the *Disk Version* or \$112.50 for the *Cassette Version* — you'll be rewarded every month for your translation efforts!

Here are some of the most important qualifications we look for in a translation winner.

Your entry must be a translation of one of the featured programs from a past issue of *SoftSide*. (We're particularly interested in Apple™ and ATARI® translations of some of our older TRS-80® only issues. Write for a list of suggested candidates.) In general, we're looking for translations of programs which are a CHALLENGE to translate. Some of the programs we publish are written in more or less "generic" BASIC, which can be typed into another computer with very few changes. Although these programs require the least effort to translate, they are also the least likely candidates for contest winners.

Your translation should be thoroughly tested and completely bug-free. Just converting program lines doesn't automatically ensure a workable translation. Be sure to use-test your translation as carefully as you would test a program you had written entirely from scratch.

Your translation should fully utilize the unique features of the computer for which it is written. The objective of a translation is to "fit" the capability and convention of its host computer, not simply mechanically duplicate the operation of the original program. This is especially true of programs which use graphics, and should be kept in mind for such minor features as keyboard layout (use of such special keys as arrows, ESC, CTRL, CLEAR, etc.). Also be careful with screen formatting; a word that spills over into the next line because of a PRINT statement that wasn't properly rewritten betrays such carelessness that we'll probably reject your translation automatically.

Your entry should incorporate any improvements and enhancements you can add to the original program. Don't feel that you have to limit yourself to the boundaries of the original. (On the other hand, don't go overboard and destroy the character of the original by completely rewriting it!) An enhanced translation is much more likely to catch our attention than a line-for-line duplicate, and it will have more value to our readers.

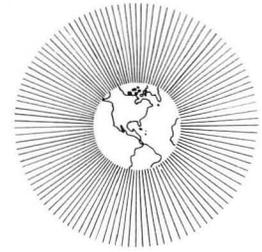
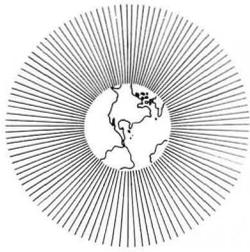
It's not necessary to include extensive documentation with your translation, only that which is different from the original. If most of the originally published documentation applies to your translation, simply say so. You should, however, include descriptions and explanations of any changes or enhancements you've made.

All Translation Contest entries must be submitted on disk or tape, with documentation in printed or typed form. Media will be returned only if accompanied by a self-addressed, stamped envelope. Send your entries to:

## SoftSide

Translation Contest  
6 South Street, Milford, NH 03055

# EVERYONE'S TALKING ABOUT EdCom<sup>TM</sup>'82



The unparalleled opportunity to expand your skills and knowledge at all levels and areas of interest in computer education.

**OCTOBER 21, 22, 23 and 24, 1982  
LOS ANGELES CONVENTION CENTER**

EXPLORE the largest educational computer showcase ever offered!

SELECT from more than 350 hours of sessions presented by internationally known speakers from across the country who will address, evaluate, analyze, and share the phenomenal developments of computers in education.

GARY BITTER - *K-12 Curriculum*  
 NICK BOHL - *Acquisition Funding*  
 G. R. BOYNTON - *University Applications*  
 HUGH BURNS - *Creative Writing*  
 DAVID BYRUM - *Science Education*  
 JUDY CHAMBERLAIN - *Strategies for the Gifted*  
 SYLVIA CHARP - *Past and Future of Micros*  
 JOHN CHILDS - *University Research and Management*  
 BETTY COSTA - *Computerize Your Library*  
 CHASE CRAWFORD - *Managerial Uses*  
 SANDRA CUNNINGHAM - *Elementary School Literacy*  
 DAN DAVIDSON - *Community College Applications*  
 MARVIN EBBERT - *Telecourses in the University*  
 JUDY EDWARDS-ALLEN - *Applying Evaluation Procedures*  
 BOB ENENSTEIN - *Administrative and Classroom Management*  
 WILLIAM ENGEL - *Instructional Computing*  
 LEROY FINKEL - *Business Education*  
 MARYLYN SUE FORD - *K-8 BASIC Programming*  
 CAROLYN GILBREATH - *Support Services for Teachers*  
 BOBBY GOODSON - *Districtwide Training for Literacy*  
 RICHARD HARTNESS - *Hardware for the Blind*  
 DAN ISAACSON - *Courseware Creation*  
 MAX JERMAN - *Program Design*  
 DEE LAMONT JOHNSON - *Special Education Software*  
 ROBERT JUDD - *Simulations*  
 KENT KEHRBERG - *MECC's Statewide Plan*

TOM KELLER - *Hardware Evaluation and Acquisition*  
 RONALD LEMOS - *Acquisition of Resources*  
 JOHN LOSSE - *Mathematics in Higher Education*  
 ANDREW MOLNAR - *Is Education Keeping Pace?*  
 DAVID MOURSUND - *Computer Literacy Without Computers*  
 TED PERRY - *Authoring Systems*  
 JIM POIROT - *Training the Teacher*  
 M. D. ROBLER - *Reviewing the Reviewers*  
 TERRI ROSEGRANT - *Reading Readiness/Special Education*  
 BARBARA SADOWSKI - *Software Modification*  
 ERNEST SAVAGE - *Vocational Curriculum*  
 DAVID SHARPE - *Authoring Systems*  
 GARY SHELLY - *Business Education*  
 TWILA SLESNICK - *Policy and Curriculum Trends*  
 VICKI SMITH - *Search for Software*  
 RANDALL STICKROD - *Creative Uses of Graphics*  
 KATHLEEN SWIGGER - *Preschool Software and Research*  
 JOHN UNDERWOOD - *CAI Strategies*  
 KATHY VOGT - *Pascal Workshop*  
 WILLIAM WAGNER - *County Van Project*  
 GEORGE WATSON - *Word Processing*  
 J. FRED WEAVER - *Mathematics CAI*  
 JERRY WILLIS - *Testing and Diagnosis/Special Education*  
 JAMES WINEBRENER - *Educational Software Evaluation*  
 KARL ZINN - *Applied Research in Higher Education*

**AND MORE!**

MicroCourses and Hands-On Sessions Will Feature

**TRS-80**

**PET**

**APPLE**

**ATARI**

**IBM**

**TI**

- Authoring
- Programming
- Word Processing
- LOGO
- Management
- Graphics

**ATTEND EdCom<sup>TM</sup>'82**  
**EXPERIENCE THE EDUCATIONAL EVENT OF THE YEAR!**  
 CO-SPONSORED BY EDUCATIONAL COMPUTER MAGAZINE

# EdCom '82

## LOS ANGELES CONVENTION CENTER

### October 21-24, 1982

## CONFERENCE & EXPO REGISTRATION INFORMATION

PRE-REGISTRATION: Please complete the form below as you wish it to appear on your name tag. Check the appropriate "Registration Fee" box below. After October 1, 1982, On-Site registration only.  
 (One day or full conference fees include admission to the exhibits)

	PRE-REG	ON-SITE REG
<input type="checkbox"/> CF — Full Conference . . . . .	\$125	\$140
<input type="checkbox"/> LA — Full Conference + 1 Luncheon . . . . .	\$140	\$155
<input type="checkbox"/> LB — Full Conference + 2 Luncheons . . . . .	\$155	\$170
<input type="checkbox"/> CO — One Day Conference . . . . .	\$45	\$55
<input type="checkbox"/> LC — One Day Conference + 1 Luncheon . . . . .	\$60	\$70
<input type="checkbox"/> LD — One Day Conference + 2 Luncheons . . . . .	\$75	\$85
<input type="checkbox"/> EO — Exhibits Only (Four Days) . . . . .	\$10	\$10
<input type="checkbox"/> LE — Exhibits Only + 1 Luncheon . . . . .	\$25	\$25
<input type="checkbox"/> LF — Exhibits Only + 2 Luncheons . . . . .	\$40	\$40

### STUDENT REGISTRATION: (Current Student ID Required)

<input type="checkbox"/> SF — Full Conference . . . . .	\$55	\$60
<input type="checkbox"/> SO — One Day Conference . . . . .	\$20	\$25
<input type="checkbox"/> SE — Exhibits Only (Four Days) . . . . .	\$5	\$5
<input type="checkbox"/> LG — Friday Luncheon . . . . .	\$10	\$10
<input type="checkbox"/> LH — Saturday Luncheon . . . . .	\$10	\$10

If you have registered above to attend one or both of the EDUCATOR IMPACT LUNCHEONS, please indicate your choice of day(s) here:

- LI — Friday Luncheon — ANDREW MOLNAR       LJ — Saturday Luncheon — SYLVIA CHARP
- TØ — I am interested in receiving information on the full and half day hands-on MicroCourses. An additional \$25 registration fee will be charged per half day session. Enrollment is limited.
- UC — Please send me information on the university credit(s) available at California State University.

NAME																		
POSITION																		
AFFILIATION																		
ADDRESS																		
CITY											STATE	ZIP						
PHONE	AREA CODE						EXT						SP2					

My registration check for the amount of \$\_\_\_\_\_ is enclosed.  
 Make checks payable to Judco Computer Expos, Inc.  
 Mail to: Judco Computer Expos, Inc.  
 P.O. Box 963  
 Scottsdale, AZ 85252

For further information call: 800-528-2355 Toll Free (Outside Arizona)  
 602-990-1715 (Arizona Only)

In the event cancellation is necessary, your written request for refund will be honored if received by October 1, 1982. A \$15 handling fee will be charged.

### EdCom '82 HOTEL HEADQUARTERS Los Angeles Hilton

Special Conference Rates:  
 Single...\$65      Double/Twins...\$80  
 (one person)      (two persons/twins)

AI — Please send additional accommodation information.

### EdCom '82 CAR RENTAL

Special Convention car rental rates have been arranged with HERTZ. Call 1-800-654-2240.  
 Give the agent the special rate code: EdCom '82 Discount ID #90038.

SAVE AS MUCH AS 42% ON YOUR AIRFARE OVER REGULAR ECONOMY FARE FROM YOUR HOME CITY ROUND TRIP TO LOS ANGELES.

FROM:	Regular Economy Fare	Special Meeting Fare	Regular Economy Fare	Special Meeting Fare
Atlanta . . . . .	\$728.00	\$524.00	Detroit . . . . .	\$708.00
Boston . . . . .	\$670.00	\$358.00	San Francisco . . . . .	\$218.00
Chicago . . . . .	\$600.00	\$360.00	Philadelphia . . . . .	\$470.00
Dallas . . . . .	\$290.00	\$156.00	Salt Lake City . . . . .	\$320.00
Denver . . . . .	\$370.00	\$230.00	Houston . . . . .	\$420.00
Miami . . . . .	\$724.00	\$310.00	Phoenix . . . . .	\$232.00
New York . . . . .	\$620.00	\$338.00	Kansas City . . . . .	\$620.00
Mobile . . . . .	\$770.00	\$360.00	St. Louis . . . . .	\$700.00

Above are sample fares. Special Meeting fare is available from most cities throughout the U.S.A. Garber Travel, official Ed Com travel agency, guarantees the lowest possible airfares will be utilized.  
 Under current regulations, air fare increases will not apply to tickets issued before the effective date of the increase. Tickets will be issued upon receipt of full payment.



1406 Beacon St., Brookline, MA 02146  
 Call toll free outside Mass.:  
 (800) 225-4570 — Request ED COM Desk

TRAVEL VIA UNITED AIRLINES



# SRI LANKA IMPORTS AT THE NCC

by David D. Busch

*Editor's Note: There were some very serious product announcements made by major companies at this year's NCC (see page 22). In contrast, we thought we'd share this humorist's postulation on some not so serious products from a not very major company.*

I was surprised at the sparse attendance at the *Sri Lanka Imports'* (SLI) booth at the National Computer Conference in Houston last June. A number of sensational products were introduced, including the firm's dynamic ROM, and the 68FF computer chip (the world's first 19-bit microprocessor). I attribute this lack of attention more to the poor location of *SLI's* booth at the event (a Galveston warehouse) than any lack of merit in the products themselves.

*Sri Lanka Imports* is, of course, a fictitious Asian supplier of space-age computer products that range from dumb terminals to dumb software. The firm, once distributed exclusively by a U.S. importer who shall remain nameless, has discarded its TRS-80®-only policy. It now encompasses a complete product line for Apple™, ATARI®, Pet™, and Sphere™ microcomputer users, and has promised new applications packages for the North Star™, Ohio Scientific™, and Cray™ systems. *SLI* also hopes to become the world's leading supplier of Space Invader games for the PDP/11.

Given the innovative nature of these new *SLI* products, it is easy to see why the fictitious firm has risen to obscurity almost overnight. Because of the low turnout at the *SLI* booth at NCC (I was the only person to stop by in more than three days), you are not likely to read reviews of these products in any other magazine. I'd suggest that you check out the following brief descriptions very carefully.

One of the most interesting items was the *SLI* "modem on a chip," which they have dubbed the Load 'Em Modem. This is an ILSI (Inconceivably Large Scale Integration) chip that somehow packs autodial, autoanswer, autowrong number, and autosuggestion features into a single component. Because the chip measures a largish 6 X 10 X 2 inches, it probably will not be incorporated into many microcomputers in the near future. Look for an outboard expansion module which will combine modem and cable-ready remote video tuning into a single unit.

I was impressed with the multiple baud rate capabilities of the Load 'Em Modem. When powered by a conventional 110 volt supply, the device communicates at 110 baud. To obtain 300 baud, you must supply 300 volts. This should be a simple matter for anyone who knows their way around a transformer. Because the modem is a single-chip device, *SLI* deemed this a simpler solution than using expensive DIP switches which must be programmed by the user.

I must caution you against carrying

this technique to its logical conclusion. Although 9600 baud communication is practical with the unit, stepping up a power supply to nearly 10,000 volts can be mildly dangerous unless you know exactly what you are doing.

I have some other tips about the modem, involving the IEEE-488 bus and use of the device with Princess telephones. However, I will save these until after the product is being shipped nationally, and I get some feedback on whether or not most buyers are surviving the baud setting procedure.

The Load 'Em Modem is possibly the first "intelligent" modem on the market. A built-in 68FF microprocessor chip examines the incoming data stream, and attempts to determine what protocols are being used. When the modem is satisfied that it has made an intelligent guess, it sends a signal to the terminal program which tells the computer what protocol is appropriate. Unfortunately, no terminal software on the market today is written to interpret this signal.

More commonly, the Load 'Em Modem will simply over-ride your CPU and handle all communications through its own microprocessor. This feat is a breathtaking technological achievement, and a bit frightening at the same time. Using the Load 'Em Modem, it is only possible to disconnect communications when the modem is good and ready.

*SLI* is also ready to tap into that vast market of home computer users who find that the \$99.00 asked by Timex for

their disposable computer is too pricey. For those who want to spend, say, \$29.95, *Sri Lanka Imports* has the answer! If you can't afford a real computer, simulate one in software.

Emulation is a very popular technique, as main frame and mini developers know. *SLI* has introduced its Computer Simulator/Trainer. The operator simply takes any ordinary cassette recorder, a television set, a typewriter (or plastic keyboard which you make yourself, if you want to simulate an RCA terminal or a Sinclair computer) and connects them with the ribbon cable supplied.

Then the "student" may sit down in front of the "CRT," type programs on the keyboard, and save them to cassette tape. I should point out that what you actually see on the screen will be garbage, your program "dumps" and "loads" will be valueless, and most of your typing time wasted. In other words, this is a VERY realistic simulation of a new user's first few sessions with a computer.

This approach seems very logical to me. Most new programmers do not turn out software that has any real application for at least six months. Certainly, putting together your own "Hangman" program is an excellent way to learn BASIC. But would you actually want to run or save such an early effort?

*SLI's* Computer Simulator offers other significant advantages. Media cost is very low: a single cassette tape may be used indefinitely. A high quality monitor or television is not required. The system may be connected to any available TV (which can double as an entertainment source while programming), oscilloscope, or cardboard box with a screen drawn on it. With the latter choice, users have reported that radiation exposure is remarkably low, and eye fatigue almost non-existent.

Look for this product, along with other trainers, such as Disk Drive Simulator, Printer Substitute, and Ersatz Software, at your local computer store.

Erasable Read Only Memory (EROM) and Erasable, Programmable Read Only Memory (EPROM) are old hat. *Sri Lanka Imports* has announced something they call "Dynamic ROM," an outgrowth of the company's research into soap bubble memory.

The chief difference between Dynamic ROM and other types of erasable ROM is that the *SLI* product

was unintended. A company researcher was looking for a way to correct some errors in 103,000 old Pet™ BASIC 3.0 ROM's that the firm had found in a trash can. Through a complicated series of procedures (a trade secret, by the way), he managed to invent the Dynamic ROM, which must be refreshed every few nanoseconds, or else it loses all of its information forever.

These Dynamic ROM's are currently being used only in *SLI's* line of One Time Software, in plug in ROM pacs for the TRS-80® Color Computer, the Sorcerer™, and ATARI® 400 and 800. For a modest fee, the user receives a game, utility, or applications program that can be run for just one session. Once the computer is turned off, the dynamic ROM loses the program and becomes useless.

*SLI* spokespersons have pointed out that most computer software on the market today is so bad that the average user will run it just once, become disgusted, and never use that program again. The One Time Software line was designed with that owner in mind.

*Sri Lanka Imports* has asked me to pass along a few tips to those of you who have already purchased *ULTIZAP*, a disk utility. Most of the complaints have come from unhappy Ap-

ple I™ and Apple II+™ owners. *SLI's* advice is as follows:

1) The Apple I™ is hopelessly outdated, and should not be used with *ULTIZAP*. In fact, we know several people who are very interested in where you got your Apple I™ in the first place.

2) *ULTIZAP* works only with TRS-80® and similar computers. Apple™ owners will have some difficulty booting the disk. But then, after DOS 3.2, 3.3, 13 and 16 sector disks, muffins, and who knows what else, you probably won't even realize that anything different is happening.

3) *ULTIZAP* is NOT designed to remove parity errors entirely. It only moves them around. If an operator has a defective sector with a parity error that makes a program unusable, the utility will move the parity error to a program that is not needed. It is doubtful that this would be of much more use to Apple I™ owners in any case, as the unit required a certain number of parity errors per sector to operate properly.

I must welcome *SLI* as a true digressor in the microcomputer industry. *Sri Lanka Imports*, bringing yesterday to you tomorrow. ☺

## SoftTakes



"HI, HILDA — I'M HOME! BRING MY DINNER TO ME AT THE COMPUTER!"

# SoftSide DV, the magazine of the future, is here!

If your computer could pick a magazine, wouldn't it prefer one in its own language? Now there's one available.

**SoftSide DV** is an enhancement of the **SoftSide** you have in your hands.

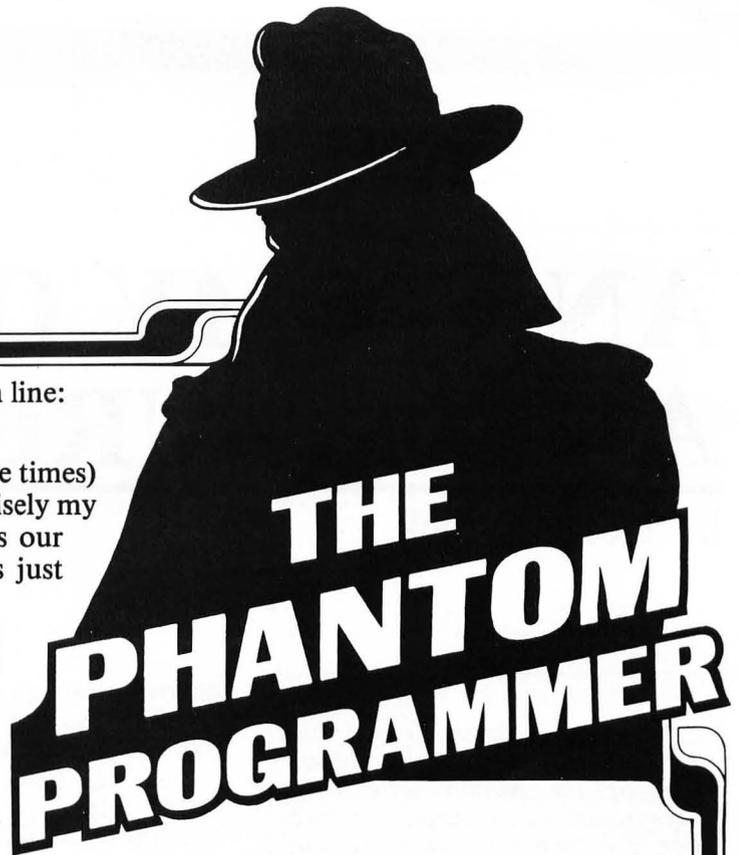
**SoftSide DV** contains not only the complete programs listed in every month's issue of **SoftSide**, but additional programs of every conceivable type, as well — multiple and Machine Language programs, modified languages, ongoing modular programs and software so extensive, it would take an entire issue of **SoftSide** just to print the code. Only the documentation for these programs will appear in **SoftSide Magazine**, **NOT** the code.

Feel as though you're missing something? You are! But, you needn't miss out on another issue. **SoftSide DV** is now available for Apple™, Atari™ and the TRS-80™. The cost to you — \$125 for 12 magazines and 12 disks, packed with some of the best software available, all delivered to your home in the next year. For orders outside the USA, please add \$36. For your convenience, we offer an installment payment plan for VISA and MasterCard holders: You pay only \$32.50 per month for four months (a total of \$130, which includes a \$5 billing charge). Please use the special DV-CV bind-in card in this issue to order.

Computerists are offered the rare opportunity of marching into a new frontier. Advance to the front of the parade by subscribing to **SoftSide DV**, the magazine of the future, available today!



# A FEW WORDS FROM...



For 5 extra points, what's wrong with this program line:

```
FOR 1 = 1 TO 100
```

Don't get upset if you had to look twice (or three times) to see the problem. Most folks do, and that is precisely my point. We have gotten into the habit of using, as our most popular variable name, a letter which looks just like our most popular digit.

This despicable habit is a remnant of the good old days of FORTRAN. In FORTRAN, the letters "I" through "N" stood for integer variables. Since loop counters are integers, these letters were heavily used to manage DO loops and the like. Naturally, the letter "I" got used more than others because it was the first in line. The fact that "I" was repeatedly elected Most Popular Variable would have been totally irrelevant were it not for the fact that "I" looks much like "1".

A similar identity crisis existed between "O" and "0". Although "O" was not nearly so popular as "I", it gained a certain notoriety due to well-intentioned efforts to clearly distinguish it from "0". It was noted that "0" and "O" did not look at all alike if a slash were drawn through one or the other. Of course, no one could ever agree on which should be slashed. It didn't really matter, as the slashed "O" (or "0") could also be misread as a "B" or an "8" if your monitor or printer were at all blurry.

As new languages were written and new computers were built, many new programmers began to learn the trade. Naturally, they learned from other programmers. Thus, whole new generations of programmers learned how to confuse "1" with "I", and "0" with "O". The problems caused by this practice were clearly recognized, and wise programmers avoided the use of "I" and "O". Old habits die hard, however, and, for every programmer who shunned the evil "I", there were 647 who used it with abandon.

Now, the magnitude of the problem makes it a true area of concern. Back when there were only a handful of programmers, the confusion of "I" and "1" or "0" and "O" did not cause any measurable loss. Now, however, there are veritable armies of programmers at all levels tapping away at terminal keyboards and the lost time is adding up. This lost time is especially painful to people who try to earn a living with their software because time is money to them. When a user misreads a printed line and his program doesn't work, the resulting customer service call costs money. When a programmer spends time tracing a virtually invisible bug, his company loses money. There is real motivation here to cut the loss by improving technique.

The simplicity of the problem and the scope of its effects make it very profitable to cure. There is no need to learn a new, more efficient language. There is no need to learn new programming procedures. All we have to do is use variable names that don't look like numbers. It certainly behooves authors who submit programs for publication to show this mercy upon the software editor who will review it and the eventual user who will make it run.

The simplicity of the problem also makes it shameful that nothing has yet been done. We must begin now. We must each take a stand. Even as non-smokers feign emphysema and asthmatic attacks to dissuade smokers, we concerned programmers must bring subtle pressure to bear on our misguided colleagues. It is time for each of us to stand up and say, "This stubborn clinging to old programming habits is inexcusable and I, for one, WILL NOT STAND FOR IT." 

# ANATOMY OF AN ADVENTURE

by Peter Kirsch

*Editor's Note: Peter Kirsch is the author of most of SoftSide's Adventure of the Month series. His background in adventure construction is extensive as, on a monthly basis, he creates an original adventure and translates it to two other systems.*

Have you ever wanted to slay a dragon, match wits with Jack the Ripper, explore deep and dangerous regions of space, or go on a fantastic treasure hunt? Playing a game of adventure will allow you to do all these things and more, without leaving the comfort of your home. An adventure lets you escape life's doldrums, forget your worries and cares — even the thought of an impending visit from your mother-in-law.

Adventure construction is an extremely challenging art. In this article, I'll try to explain how I construct an adventure, from the initial idea to playtesting the final version. I'll also explain various techniques used in translating TRS-80® adventures to the Apple™ and ATARI®.

## The Idea

The primary stumbling block to creating an adventure is developing a good idea. The days of simply finding treasure and returning it to a storage location are gone forever. An adventure should be original and have a unique quest. Most of mine have a basic storyline, which a player

discovers through exploration. I've rejected many ideas due to lack of good plot or limited subject matter.

Once you have an original and exciting plot, it must be further developed. At this point, I draw bits and pieces of various adventure maps, trying many configurations to get that perfect layout. Tentative puzzles and hazards are also jotted down. When I'm satisfied with what I have, I put my drawings and notes together and create a final version of my adventure map on a giant piece of heavy paper. All locations are numbered and any last minute changes, additions, and deletions made. (Changes might still be required during actual programming due to memory limitations. For example, my *Titanic Adventure*, as originally written, ran over the allotted memory and some locations and related puzzles had to be eliminated.)

## The Adventure Skeleton

Early in my adventure writing career, I created an adventure interpreter, or skeleton, as I call it, to serve as the backbone of each of my adventures. It has since been updated many times (now at version 4), but basically remains the same tool. With the skeleton as a base, only data and subject matter pertaining to the particular adventure need be added, saving a great deal of programming time. The skeleton consists of: room allocation, initialization, display routines, input commands, command routines, and data allocation. (See line listing following this article.)

My original adventures are created on a TRS-80®, then translated to the Apple™ and ATARI®. As I explore

each section of the skeleton, I'll discuss the differences in the various machines, and how each handles the same routine. The string handling of the ATARI® is a prime example, but Applesoft, which, in fact, is a version of Microsoft BASIC, has just as many variations. ATARI® BASIC has some added "goodies" that the others lack, not to mention easily accessible colors and sounds which can enhance an adventure. Most of the discussions pertaining to the TRS-80® will pertain to the Apple™ as well. Any significant changes will be noted.

Imagine, now, that I've got my large adventure map in front of me. I need to transfer my ideas from map to skeleton. First, I put all room descriptions at the top of the program and any visible exit represented on that line by a single or double variable (W,N,E,S,U,D,NW,NE,SW, or SE), set to its new location number. Room lines are numbered by 1's to make them easily accessible by an ON A + 10 GOTO line # statement where A is the current location. The use of low line numbers also helps prevent breaking the statement into two or more lines, since as many as 75 line numbers could possibly follow. An excellent feature of ATARI® BASIC is the use of variables as data pointers. A simple GOTO A + 10 does the trick.

Near the end of the program are the data lines. These include objects, object locations, verbs, and corresponding verb numbers. These are initially read into A\$(x), A(x), B\$(x), and D(x) respectively. Depending on the particular adventure, there may be additional data.

Ah, the problem with the ATARI®. Since ATARI® BASIC does not



support string arrays, all objects are read into a single, large string, A\$, and have additional spaces appended to make them all the same length. When called for, an object will be accessed by its position within the string. ATARI® BASIC does this with the following line:

```
210 FOR A=1 TO 40:A$(A*SZ-
SZ+1,A*SZ)=SZ$:READ X$:
A$(A*SZ-SZ+1,A*SZ)=X$:
NEXT A
```

Let's say there are forty objects in the game. SZ\$ first fills and clears that segment of the string with spaces. The object, X\$, is then read into that position. Variable A marks its position within the string, and SZ, the length, including spaces.

It may also be necessary to change an object into another object at some time. For example, DOOR might become OPEN DOOR, or BANANA could evolve into PEELED BANANA.

This is no problem if using string arrays, but in the ATARI®, A\$(x1,x2) is used, where x1 is the starting position, and x2 the ending position of an object. In essence, this has the same function as the MID\$ statement in Microsoft BASIC and Applesoft, but ATARI® BASIC also allows you to use it on the left side of the equation.

The current location of each item is held in A(x). If an object cannot physically be carried by a player, it is preceded by a minus sign. Location numbering starts with 5, since the lesser numerals are used as follows. If an item is carried, it's assigned a -1, if worn, a -2, and if ridden, as in the case of a horse, a -3. If an object is not yet in play, out of play, or used repeatedly, such as TREES in a forest, at many locations, it's assigned a 0.

The display routines are basically the same in all translations. I make further use of the ATARI®'s colors to visually enhance an adventure. Background

colors are used to set the mood of some locations: green for grass, a rich blue if you're on or beneath water, and perhaps a dull red for an eerie castle setting. The normal background color is black.

When a player issues a command, array B\$(x), or string B\$ in ATARI® BASIC, is scanned to see if a matching verb is found. If not, the computer will tell you that it doesn't understand you and branch back for another player input. Array D(x) holds a corresponding verb number, which allows the use of synonyms for the same command, such as GET and TAKE or DROP, PUT and GIVE. This number will then be used to branch to the proper command section in the program.

The TRS-80® and Apple™ both look at the last three letters of the object command to see what the player wants to do. This also alleviates some of the annoying keyboard bounce in the TRS-80®. In ATARI® BASIC, however, this method won't work since all objects have been padded with spaces. The first three letters are recognized instead. This also presents a need to specify the necessary three letters in front of an object in the data statements. LITTLE RED BRICK, for example, is listed as BRILITTLE RED BRICK. This will usually be necessary only when using the GET and DROP commands.

There are usually ten to twenty different commands in my adventures. Each command has a different section in the program. When a proper command has been entered, logic shifts to the proper section, depending on the verb number, to determine if it makes any sense. If it does, and all conditions have been met, (there are usually many), then the proper action is taken. All else defaults to "You can't do that."

This completes the basic adventure outline. Of course, each adventure will have its own logic and special features. Frequent use is also made of sound effects in the ATARI® versions. Sounds are added, such as explosions, gunshots, vehicles, running water, wild geese, and things-that-go-bump-in-the-night. There's even a small sound routine in the ATARI® version of *Robin Hood Adventure* which simulates an arrow slamming into a target.

```
1700 V3=15 TO 0 STEP -1:SOUND
1,77,6,V3:FOR V4=1 TO 5:NEXT
V4:NEXT V3
```

## Playtesting and Debugging

If you think you're done with an adventure once the code is completed, think again. Playtesting and debugging an adventure are as much a part of the process as the actual writing. That brings us to the editing capabilities of the various computers. The ATARI® has, by far, the easiest editing. The TRS-80® is next in line, and the Apple™ is a distant third. The joy of editing with the ATARI® stems from the fact that program continuation is not halted once a change has been made. This is a tremendous aid when debugging an adventure. You can continue playtesting once a change or correction has been made with all progress intact. Because of this feature, I correct any bugs I find in the ATARI® version immediately. The TRS-80® and Apple™ don't have this capability. After any correction, you'll have to RUN the program again. Both the TRS-80® and Apple™ will allow you to continue upon encountering an error *as long as no change has been made*. I go around any bugs I find for as long as possible, keeping a list of them on scrap paper, until there are too many, or a critical one pops up. A simple GOTO 300 will usually put me back into a working program.

Another plus feature when editing, or even programming, with the ATARI® is the ability to change line numbers. Retyping over an old line number with a new one, and hitting RETURN, will add the new line to the buffer and still keep the old one intact. In essence, we now have two identical lines. If the old line is no longer needed, it can be deleted. This feature is very handy when the need to shift lines arises while editing. It can also save some programming time when you need to repeat a line or create a nearly identical line with minor changes.

As stated previously, the Apple's™ editing capability is nothing to write home about. It does have a screen editor, as does the ATARI®, but once a line is entered and listed, editing the format of the listing can become quite a chore, especially for novice programmers. With practice, however, one can become quite adept.

Debugging an adventure can be a time-consuming task. You can very easily spend a whole weekend looking for that one, elusive bug. Then, after you think you've found your last one, another will pop up. Believe me, I know! Doing translations is also a debugging aid. Errors will be less extensive than in the original version, but

they will be there, and you'll probably find something you missed the first time around.

Playtesting an adventure consists of three stages: initial run, thorough playtesting, and fine tuning. All my translations receive the same treatment as the original. My large adventure map is in front of me and a location tester, PRINT A, is temporarily added into the program which checks to see if you're in the proper room.

Initial playtesting consists of running through the adventure once in chronological order, doing exactly what's necessary when it's necessary, and nothing else. Any errors at this point are corrected, and when the program behaves exactly as it should, a grueling session of thorough playtesting begins. Every command is tested, with every combination of objects in every conceivable and absurd way. Every possible message is also printed out. All responses from a player are considered. Any corrections will require further playtesting.

The last stage of playtesting now begins. I'll check any routines I put off until later, such as making sure all possible ways that you can get killed work properly. Getting killed does not halt program continuation, as a rule. Typing GOTO 300 will reincarnate you before that fateful moment. Die-hard adventurers will not want to cheat in this manner, however.

Translating an adventure to the Apple™ and ATARI® is a bit cumbersome because neither BASIC has an ELSE statement. What would normally follow on the same line on the TRS-80® may require several lines with the others. Also, the THEN statement is not optional in Applesoft and ATARI® BASIC. Since I often use the variable A, care must be taken when using it immediately before a THEN in Applesoft. The Apple™ will think that the BASIC keyword AT is implied and space it as such. With the Apple™, the variable A must be enclosed in parentheses.

During the programming of an adventure, it is a good idea to periodically save the progress of your labors to cassette or disk — every 30 minutes should be sufficient. In the event of a sudden power failure, or when your ATARI® locks up for unexplainable reasons, as has happened to me on several occasions, you won't lose too much programming time. The rate at which the ATARI® CSAVES and CLOADS almost justifies the purchase of a disk drive, which is a necessity when doing any lengthy programming. I lose one or two hours of programming time per

day using a tape system. The TRS-80® has the CLOAD? verification of a CSAVE, and others *should* have this feature. This might prevent many programmers' hair from prematurely graying, and their fingernails from getting short.

## Creating Puzzles

A good adventure is well thought out and has a good plot with plenty of subject matter. Puzzles should be hard, but not impossible, to solve. I'll explain some of the ways which puzzles can be implemented.

Walking through a forest one day, you come upon a car. The ignition key, however, is nowhere to be found. Walking further, you discover a grizzly bear sitting under a tree, playing with what looks like the missing key. Fercible retrieval is not recommended. You look up, and see a beehive in the tree above the bear. There's a rock! You pick up the rock and throw it at the beehive. You watch as a swarm of bees chase the fleeing bear, who drops the key in his panic. Throwing the rock at the bear would have angered him and probably meant your demise. In this case, you would be asked at what, specifically, you wish to throw the rock.

Sometimes, an obstacle may be insurmountable — placed there just to throw the player off the track. You may spend hours trying to find a way to enter a certain building, when, in reality, it's just part of the scenery. There may also be some pitfalls from which a player is unable to recover. For example, don't enter a deep hole without a long rope tied to a post on the surface. Don't forget the ever-popular flashlight. Don't eat an item of food not meant for your consumption. If you do, you'll never get past that nasty, grinning dwarf, blocking your way, who just loves apples.

Some objects may have more than one use, so it's not wise to simply forget about an item, once used. A bottle, which you find floating in the ocean, may have a paper message inside. This same bottle might be used, later, as a container to hold water. A third use might be as a cutting tool.

With all of these steps completed, the adventure is finished. I make a duplicate for backup use and mail the originals to *SoftSide*, along with any necessary documentation, including a complete solution sheet.

Planning for my next adventure always begins long before the completion of my last, and it's always different and perhaps even harder to solve.

*\*The following adventure skeleton line listing is NOT designed to run as is. It is presented as a reference tool for the preceding article and requires considerable adaptations in order to run as a complete adventure.*

```

SS SS
SS
SS
SS      TRS-80 Model I/III BASIC      SS
SS      "ADVENTURE SKELETON"          SS
SS      Author: Peter Kirsch           SS
SS
SS SS SS SS SS SS SS SS SS SS SS SS

```

```

5  GOTO200
10 ONAGOTO0,0,0,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,3
0,31,32,33,34,35,36,37,38,39,40

```

Room descriptions go here. Use of lines 15-40 is typical. (Note: The following lines with dots are not to be typed in; they represent lines to be typed in by the programmer when writing the adventure.)

```

15 .....
nn .....
40 .....

```

The four mandatory DIMs.

```

200 CLEAR150: DIMA$(43), A(43), B$(25), D(25)

```

Initialization.

```

210 FORA=0TO43: READA$(A), A(A): NEXT: FORA=1TO25: READB$(A): NEXT: FOR
A=11TO25: READD(A): NEXT

```

Display routines.

```

300 N=0: W=0: E=0: S=0: U=0: D=0: NW=0: NE=0: SW=0: SE=0: X4=0: X5=0: CLS: P
RINT"YOU'RE ";
310 GOTO10
357 PRINT: IFN+W+E+S+U+D+NW+NE+SW+SE>0THENPRINT" SOME EXITS ARE: "
;: FORB=1TO10: B(B)=0: NEXT
360 IFW>0PRINT" WEST";: B(1)=W
361 IFN>0PRINT" NORTH";: B(2)=N
362 IFE>0PRINT" EAST";: B(3)=E
363 IFS>0PRINT" SOUTH";: B(4)=S
364 IFU>0PRINT" UP";: B(5)=U
365 IFD>0PRINT" DOWN";: B(6)=D
366 IFNW>0THENPRINT" NORTHWEST";: B(7)=NW
367 IFNE>0THENPRINT" NORTHEAST";: B(8)=NE
368 IFSW>0THENPRINT" SOUTHWEST";: B(9)=SW
369 IFSE>0THENPRINT" SOUTHEAST";: B(10)=SE
370 PRINT: PRINT
375 FORB=0TO43: IFABS(A(B))=ATHENC=C+1: GOTO377
376 NEXT: PRINT: GOTO379
377 IFC=1PRINT" VISIBLE ITEMS:"
378 PRINT"  "A$(B): GOTO376

```

Player input commands.

```

390 ONERRORGOTO3000: C=0: X4=0: X5=0: PRINT: PRINT"WHAT DO YOU WANT
TO DO";: INPUTA$: PRINT: IFA$="LOOK"THEN300

```

Directional moves.

```

393 FORB=1TO10: IFA$=B$(B) THEN394ELSENEXT: GOTO397
394 IFB(B)<>0THENA=B(B): GOTO300ELSEPRINT"YOU CAN'T MOVE THAT WAY
."; GOTO390
397 IFA$<>"I" THEN400ELSEPRINT"YOU ARE CARRYING: "; FORK=0TO10: IFA(
C(K))=-2THENPRINT"WEARING"ELSEIFA(C(K))=-3PRINT"RIDING"
398 IFC(K)>0PRINT"  "A$(C(K))
399 NEXT: GOTO390
400 FORB=1TO25: F=LEN(B$(B)): IFLEFT$(A$, F)=B$(B) THEN450ELSENEXT:
PRINT"DON'T KNOW WHAT "CHR$(34)A$CHR$(34)" MEANS."; GOTO390
450 D$=RIGHT$(A$, (LEN(A$)-F)-1): E$=RIGHT$(D$, 3)

```

"GET" command.

```

452 IFD(B)<>1THEN490
453 I=0: FORV=0TO10: IFC(V)<>0I=I+1
455 NEXT: IFI>6PRINT"YOU CAN'T CARRY ANYMORE": GOTO390
463 FORJ=0TO43: IFE$=RIGHT$(A$(J), 3) ANDA=ABS(A(J)) THEN471
464 NEXT
465 PRINT"THERE'S NO "D$" HERE."; GOTO390
479 IFA(J)<0PRINT"THAT'S IMPOSSIBLE TO CARRY": GOTO390
480 FORK=1TO10: IFC(K)=0THENC(K)=J:A(J)=-1: GOTO5050ELSENEXT

```

"DROP" command.

```

490 IFD(B)<>2THEN600
500 FORJ=0TO43: IFE$=RIGHT$(A$(J), 3) IFA(J)<>-1ANDA(J)<>-2ANDA(J)
<>-3THEN505ELSE510
505 NEXT
510 FORK=1TO10: IFC(K)=JTHENC(K)=0:A(J)=0: GOTO514ELSENEXT
513 PRINT"YOU'RE NOT CARRYING IT": GOTO390
520 A(J)=A
530 PRINT"OK!"
590 GOTO5100

```

Rest of commands start here, a separate section for each.

```

600 .....
nnn .....

```

```

1000 PRINT"SORRY, YOU CAN'T DO THAT": GOTO390

```

Subroutine for adding, deleting, or changing an inventory item.

```

1100 FORK2=1TO10: IFC(K2)=K3THENC(K2)=K0:A(K3)=0:A(K0)=-1: GOTO510
0
1115 NEXT

```

DATA in the format of object, initial object location.

```

2000 DATA .....
nnnn DATA .....

```

DATA in the format of verbs.

```

2100 DATA .....
nnnn DATA .....

```

DATA in the format of the matching verb number.

```

2110 DATA .....
nnnn DATA .....

```

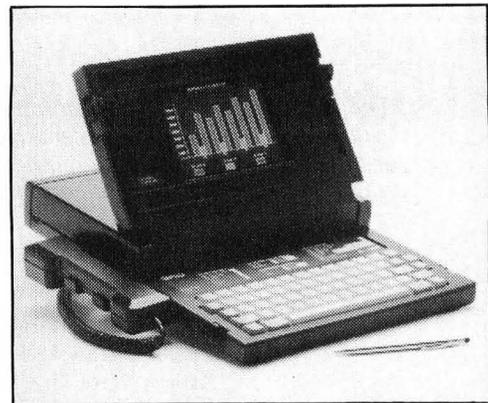
```

3000 PRINTA$ "WHAT?": RESUME390
5000 PRINT: INPUT"HIT ENTER"; A$: GOTO300
5100 FORHI=1TO2000: NEXT: GOTO300
6700 PRINT"THIS ADVENTURE
IS OVER": END

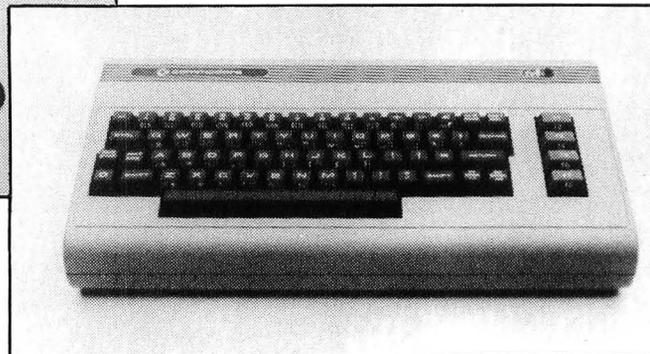
```



The Sony SMC-70 Microcomputer



GRID Systems Corporation's new Compass Computer



The Commodore 64

# The 1982 National Computer Conference: What's A Mainframe?

by George Blank

As 100,000 people gathered at the Astrodome in June for the National Computer Conference, it was obvious that microcomputers had at least arrived, if not taken over, the show. Apple was in the high status, center portion of the Astrohall, fraternizing with Texas Instruments, IBM, Xerox and DEC. All four of those companies were displaying microcomputers.

Further out in the wings of the Astrohall were the Japanese; Toshiba, Hitachi, Epson, Sony, Panasonic, Sharp, NEC, Sanyo, Casio, Okidata, and others, all ready to pounce on the US market.

Software developments at the show were not revolutionary. Almost every exhibitor was showing a *something Calc*, with prices from \$50 to the stratosphere, and all advertising features not in *VisiCalc*, such as the ability to print variable width columns. Every conceivable idea seemed to have been tried, executed in software, and copied by a dozen firms. There were 47 entries in the show directory under data base management systems, and

I'm sure that the majority were not even listed. One trend was the increasing recognition of the value of color, with many good color graphics packages demonstrated, and imaginative use of color in applications packages. *UNIX* was the "in" operating system of the show, particularly on 68000 based computers like the Fortune 3216, which was also demonstrating spectacular color graphics. Microsoft was showing *XENIX*, their version of *UNIX*.

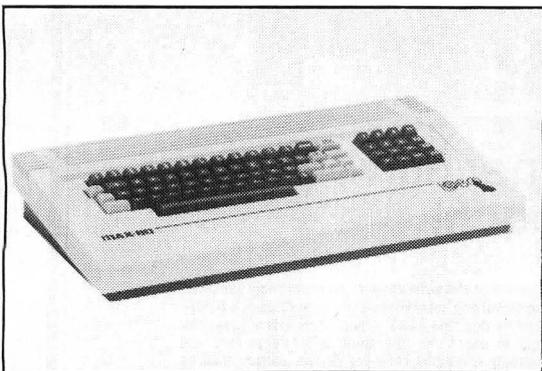
For the computers covered by *SoftSide*, the Apple™, ATARI®, and TRS-80®, the news of the NCC was no news. ATARI® did not exhibit, preferring to concentrate its efforts on the Consumer Electronics Show taking place in Chicago at the same time. Apple had the same old models II and III, and the booth staffers either pretended ignorance or became rude when asked about "Lisa," the code name for the next generation of Apples. Radio Shack had nothing more recent than the Model 16 and the second edition of the Pocket Computer.

*SoftSide*

Commodore introduced a 64K memory version of the *VIC*, clearly aimed at the ATARI® marketplace. The price, at \$600, is very aggressive and the color graphics abilities and other special capabilities are nice. Commodore had two other new machines at the show. They claim to have shipped more computers worldwide than anyone else, but with so many models, screen formats, and ROM changes, no single Commodore computer seems to have adequate software support.

There was one significant new development in the clone category, with a TRS-80® software compatible computer introduced by Lobo. The attractive and aggressively priced *Lobo Max 80* has a 5 megahertz Z-80 (more than twice as fast as the Model III), 64K of RAM, built in expansion bus, printer port, two RS-232C serial ports, battery operated real time clock and disk controllers for all standard floppy disk drives as well as a hard disk interface.

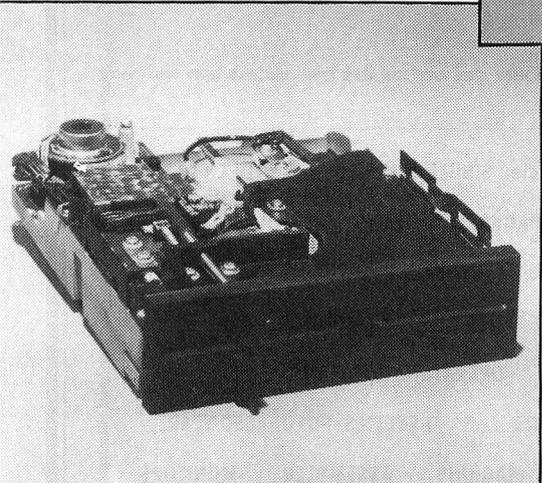
The *Max 80* will operate under both



The Lobo Max-80



Epson America's HX-20



Tandon Corporation's 48-TPI Floppy Disk Drives

LDOS and CP/M, will allow software selection of a 24 by 80, 16 by 64, or 16 by 32 character screen format, and comes with a 76 key keyboard with numeric key pad. The price of the complete 64K computer is \$800, with a green screen monitor available for another \$150. Disk drives are extra, but that can be as simple as plugging in a TRS-80® cable and disk drives, though Lobo has a line of 5 1/4 inch, 8 inch and hard disk drives for the unit.

Many companies were showing IBM products. Lobo introduced 8 inch disk drives, Personal Micro Computers a line of low cost peripherals, and many software companies that marketed TRS-80®, Apple™ and CP/M software were showing new programs, as well as new versions of old programs, for the *IBM Personal Computer*. MicroPro was showing *WordStar* in color on the IBM, and asking whether passers-by thought the color added to the value of the product.

Tandon Corporation introduced a half height, low performance mini-floppy disk drive that it is selling to manufacturers for \$50 (minimum order 10,000, please). We are certain to see this unit in several low end systems

in the future, and one unconfirmed, speculative rumor has it that ATARI® will be using it in a personal computer with built in disk drive for less than \$1000. Tandon also announced a hard disk drive available to manufacturers for less than \$1000.

This year's ultimate executive status symbol is the GRID Systems Corporation *Compass Computer*. The *Compass* is about the size of a book, weighs a little over 9 lbs., and has a 320 by 240 dot, 5" by 3 1/2" flat screen amber display built into the fold-down cover. The screen allows 24 lines of 70 characters or some very nice graphics. The *Compass* comes with a *UNIX* like operating system, BASIC, Pascal, C, PL/M, Fortran 77, an editor, a macro assembler, a relational data base manager, an electronic spread sheet, a graphics and plotting program, a word processor, and a critical path project management system. A 1200/300 baud modem with auto dial and auto answer is built in. Two modular phone jacks even allow you to plug in a handset at the same time you plug into the phone lines, so that you can have a power assisted telephone. Software is distributed by the company over communications links. The system has 256K of RAM and 256K of bubble memory built in, and sells for \$8150.

Those looking for a computer about the same size and shape as the *Compass* on a much smaller budget may be interested in the *Epson HX-20*. The \$800, 4 lb. computer comes with 16K of RAM. It has a liquid crystal display screen with a matrix of 120 by 32 dots, enough for 4 lines of 20 characters. There are 32 special graphics characters. A calculator type printer is built in, as well as a clock with calendar, timer and alarm. It has a nice 68 key standard keyboard with 5 special function keys. The unit has 4 nicad

batteries that recharge in 8 hours and last for up to 50 hours, depending on how much you use the printer or modem. Optional accessories include a microcassette drive for off-line storage, a bar code reader, an expansion unit for more RAM and ROM, and a modem.

Only those of you who have read this far deserve to know about our favorite new product. For the price of an Apple™, and about \$1000 less than a comparable *IBM Personal Computer*, you will soon be able to buy a *Sony SMC-70* computer. Unlike anyone else, whether U.S. or Japanese, Sony seems to have done everything right. The basic computer, which sells for \$1475, weighs 10 1/2 pounds, is 15 inches wide and 17 inches deep, and has 64K of program RAM and 38K of graphics and video RAM. It has a 4 megahertz Z-80 processor and runs CP/M. Text modes include either 25 lines of 80 characters or 2 pages of 25 lines of 40 characters. The character set is fully programmable, the graphics resolution ranges up to an incredible 640 by 400 dots, allowing 16 colors in the 320 by 200 dot mode. A battery backup calendar clock, audio speaker, tone generator and sound level switch are also standard.

Connectors are built in for RS-232C, parallel printer, light pen, RGB and composite video, earphones, numeric key pad and cassette tape recorder. There are five expansion I/O slots for floppy disks, extra memory, and peripherals. Additional peripherals announced were an expansion unit with five additional slots, additional RS-232C interface, IEEE-488 interface, 8 inch disk controller, 8 inch floppy and hard disk drives, video signal converter, battery backup unit, 256K cache memory storage, video cassette recorder control unit, videodisc controller, 192K RAM bank, three printers and a 16 bit adapter with an 8086 microprocessor and 256K of RAM that should run the same DOS as the *IBM Personal Computer*.

The standard floppy disk option costs \$1100 extra and includes two of Sony's new 3 1/2 inch floppy disk drives with 280K of storage per diskette. Sony was also showing an \$895 RGB color monitor with software selectable 525 and 625 line resolution.

At the show, Sony demonstrated the *SMC-70* controlling an industrial videodisc, and also showed the 16 bit adapter. With the 8086 in place, the Z-80 is used as an I/O controller, creating a very powerful system. Since I can't afford a *Compass*, I want a Sony *SMC-70* next!

# Win \$500!

## The First National Computer Owner Survey 50 Second Prizes of \$10 each!

In order to keep developing and bringing you very special hardware, software and publications, we've been commissioned to find out what you, the customer, wants and needs.

In addition to contributing to the computer owners' data base, you have a chance to win \$500...just for filling out this survey.

**JUST TAKE A FEW MINUTES, ANSWER EVERY APPLICABLE QUESTION - YOU MUST TO BE ELIGIBLE - AND MAIL TO US NO LATER THAN OCTOBER 31, 1982. FOR 20¢ YOU COULD WIN HUNDREDS.**

Entrants must be computer owners or users and answer every applicable question. A random drawing, eligibility approval and list of winners will be verified by a Notary Public. Winners will be notified by Dec. 31, 1982. Grand Prize winner gives IRV Brechner Enterprises the right to use name and photo in future surveys and advertising. No purchase necessary. Limit one entry per person. Entries must be postmarked no later than Oct. 31, 1982 and reach us by Nov. 15, 1982. Prizes include one cash award of \$500, and 50 cash prizes of \$10 each. All survey entries become property of IRV Brechner Enterprises; none will be returned. All prizes will be awarded by Nov. 31, 1982. All Federal, State and Local taxes are responsibility of the winner. This contest void where prohibited by law. For a prize winner list, send a self-addressed stamped envelope to IRV Brechner Enterprises, Box 264WOB, West Orange, N.J. 07052.

NAME \_\_\_\_\_ ADDRESS \_\_\_\_\_

CITY/STATE/ZIP \_\_\_\_\_ AGE \_\_\_\_\_ SEX \_\_\_\_\_ Circle: SINGLE MARRIED # CHILDREN \_\_\_\_\_

COMPUTERS(S) YOU OWN OR USE (Circle) APPLE ATARI TRS-80 IBM PC HEWLETT-PACKARD HEATH ZENITH OSBORNE

XEROX DEC TEXAS INST. NORTH STAR COMMODORE VECTOR CROMEMCO OTHER OWNED HOW LONG

# DISK DRIVES \_\_\_\_\_ OWN HARD DISK? \_\_\_\_\_ OWN MODEM? \_\_\_\_\_ PRINTER BRAND \_\_\_\_\_ MONITOR BRAND \_\_\_\_\_

APPROX. # DISKETTES OWNED \_\_\_\_\_ BRAND PREFERENCE \_\_\_\_\_ OPERATING SYSTEM(S) \_\_\_\_\_

AMOUNT OF MEMORY (Circle) 8K 16K 24K 32K 48K 64K 128K MORE # DISKETTES PURCHASED/YEAR \_\_\_\_\_

LANGUAGES YOU PROGRAM WITH OR USE (Circle) BASIC FORTRAN COBOL MACHINE ASSEMBLER MONITORS

FORTH ALGOL PASCAL C ADA APL LISP CAI DO YOU WRITE YOUR OWN SOFTWARE (Circle) YES NO

APPROXIMATE NUMBER OF SOFTWARE PROGRAMS YOU OWN PER CATEGORY:

EDUCATION \_\_\_\_\_ BUSINESS \_\_\_\_\_ GAMES \_\_\_\_\_ SCIENTIFIC \_\_\_\_\_

HOBBY \_\_\_\_\_ HOME USE \_\_\_\_\_ OTHER \_\_\_\_\_

YOUR PROFESSION \_\_\_\_\_ PRIMARY USE FOR YOUR COMPUTER \_\_\_\_\_

APPROX ANNUAL INCOME (Optional) \_\_\_\_\_ MOST RECENT EDUCATION LEVEL (Circle) HIGH SCHOOL SOME COLLEGE

COLLEGE GRADUATE SOME GRADUATE SCHOOL MASTER'S DEGREE OTHER \_\_\_\_\_

WHICH PUBLICATIONS DO YOU SUBSCRIBE TO OR READ REGULARLY (Circle all that apply) APPLE ORCHARD BYTE CALL-APPLE

COMPUTE COMPUTERWORLD CREATIVE COMPUTING DESKTOP COMPUTING INTERFACE AGE INFOWORLD

MICROCOMPUTING MICRO MICROSYSTEMS NIBBLE PEELINGS II POPULAR COMPUTING PERSONAL COMPUTING

SOFTSIDE OTHERS \_\_\_\_\_

APPROX # COMPUTER BOOKS OWNED \_\_\_\_\_

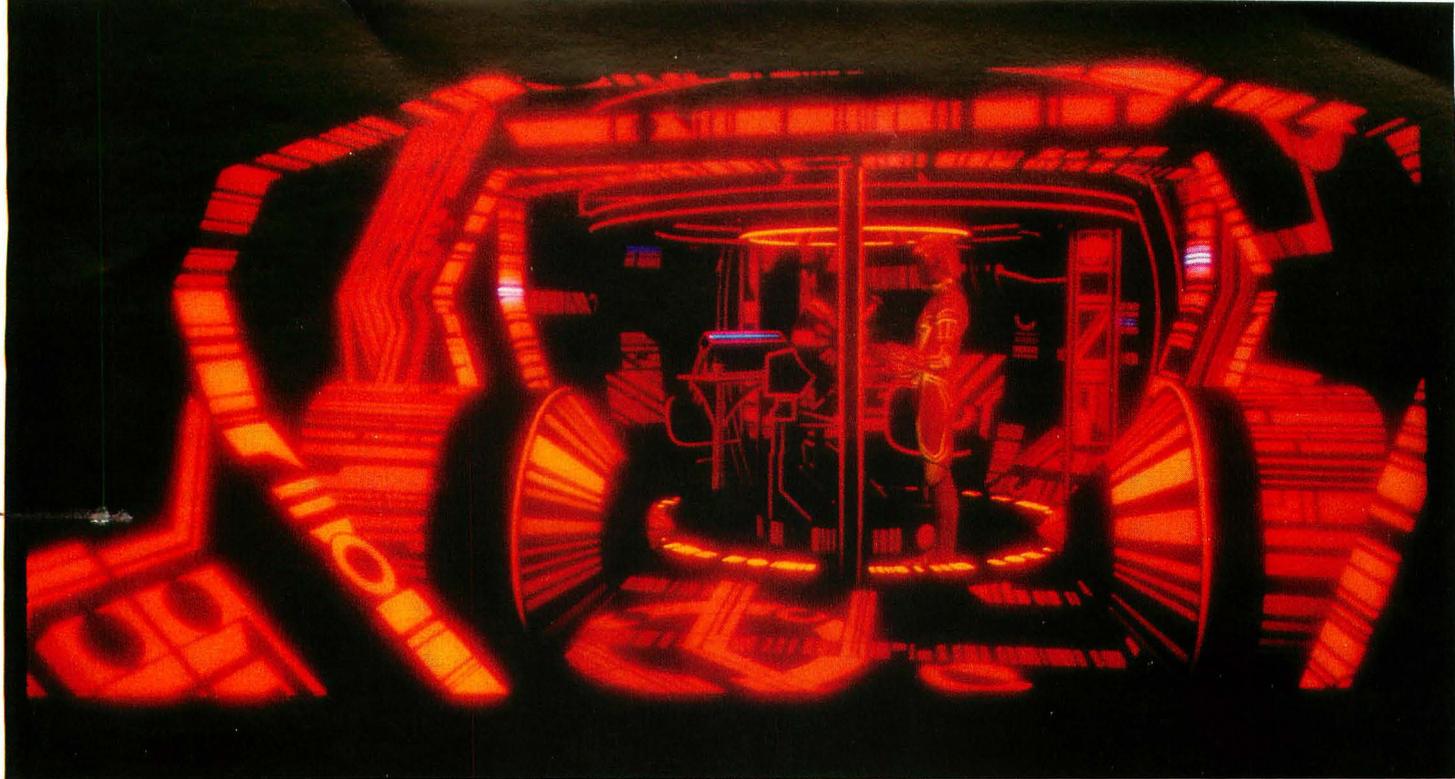
WHAT NEW PRODUCTS, IDEAS, HARDWARE, SOFTWARE, PUBLICATIONS, ETC. DO YOU WISH TO SEE COME ABOUT?  
Please be specific and use additional paper if necessary. Staple to survey when completed.

Signature \_\_\_\_\_

Mail all completed surveys by Oct. 31, 1982 to:  
NATIONAL COMPUTER OWNERS' SURVEY • BOX 264WOB • WEST ORANGE, N.J. 07052

55

\*Apple, Atari, TRS-80, IBM PC, Hewlett-Packard, Heath, Zenith, Osborne, Xerox DEC, Texas Inst., North Star, Commodore, Vector & Cromemco are all registered trademarks.



## ENTERTAINMENT TOMORROW

### THE WORLD OF

# TRON

#### *Shadows and Magic*

*"I have a little shadow that goes in and out with me. And what can be the use of him is more than I can see"*

Robert Louis Stevenson, "My Shadow"

by Fred D'Ignazio and Allen L. Wold

Arthur C. Clarke's Third Law reads: "Any sufficiently advanced technology is indistinguishable from magic." Disney Studio's new film, *Tron*, was made by technology so advanced it proves Clarke's theories. *Tron* is about a magical journey into the heart of that technology — the bizarre and enchanting world inside the computer, inhabited by electronic beings who are a shadowy reflection of humans.

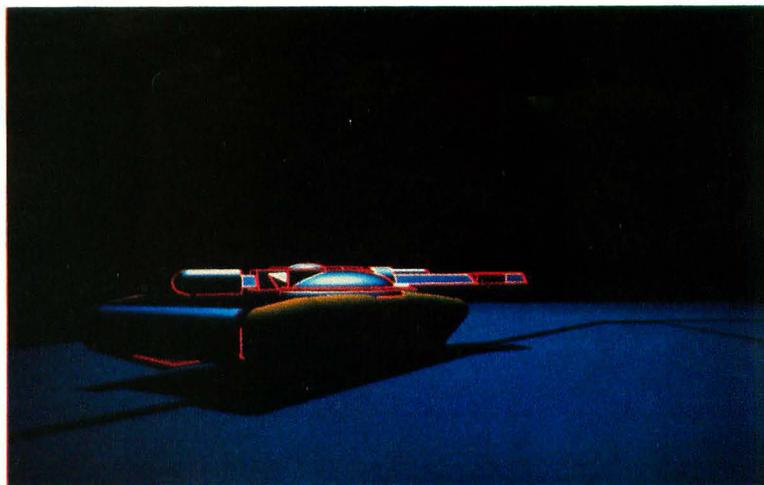
*Tron* is the story of Flynn, a young computer expert and video game genius. Flynn's old company, ENCOM, stole some popular and profitable arcade games that he invented. In frustration, Flynn leaves ENCOM and starts his own arcade. From a terminal, hidden away above his amusement palace, he tries to sneak onto the ENCOM system and uncover the

evidence he needs to prove the theft of his game programs.

With the help of friends still at ENCOM, Flynn sneaks back into the company's building to a terminal with privileged access to the ENCOM computer system. Unfortunately, MCP, ENCOM's sentient, evil "Master Control Program," detects Flynn's presence at the terminal and blasts him with a new laser teleportation device. The laser transforms Flynn into an electronic being, and MCP sucks him into the bizarre world of the computer.

Until his encounter with MCP, Flynn was a videogame inventor — a master at creating exciting battles between electronic gladiators who sprang to life at the drop of a quarter. Inside the computer, Flynn confronts his own creations. The deadly weapons he had

*SoftSide*

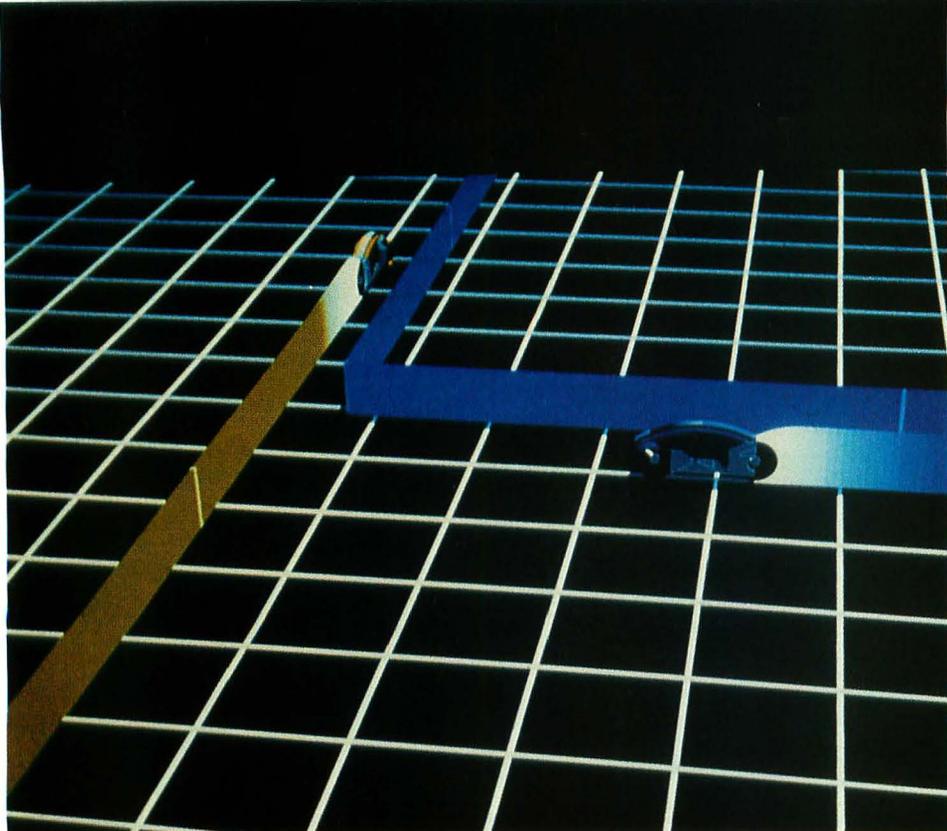


designed — mere images on the display screen — are now real and threaten to kill him.

Fortunately for Flynn, he is not the only rebel force opposing MCP. Another is TRON, a program/being dedicated to overthrowing MCP and returning control of the huge ENCOM computer back to its human "users." Flynn and TRON join forces and battle the warriors and weapons thrown at them by MCP and its villainous champion, SARK. That battle, and Flynn's attempts to escape to the real world, dominate the remainder of the movie's plot.

#### **An Anti-Computer Movie?**

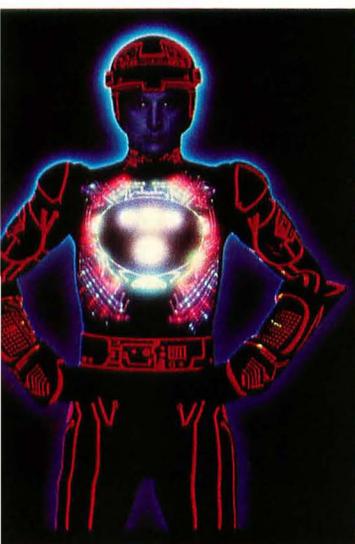
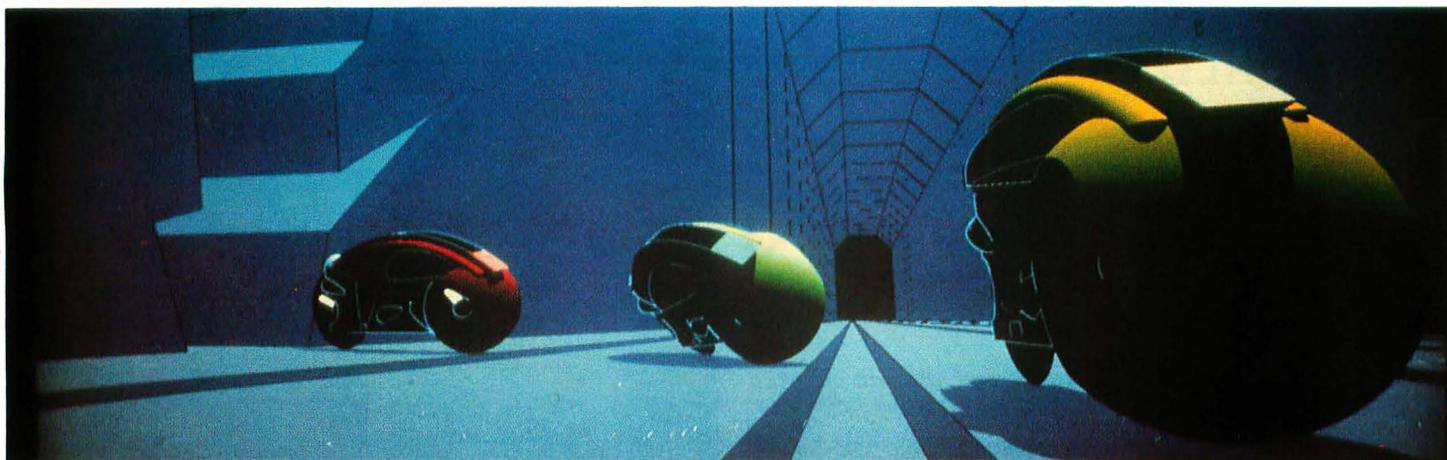
Some critics have claimed that *Tron* is an anti-computer movie, in the



tradition of films like *Failsafe*, *The Forbin Project*, *Demon Seed*, 2001, and, recently, *Evilspeak*. The critics point to the demonic program, MCP, and its classic computer obsession of revenge against its human users.

According to writer-director, Steven Lisberger, *Tron* is not anti-computer at all. In Lisberger's eyes, the computer is a powerful tool that is ethically neutral. The story is about a fantasy world that exists in the wires, circuits, and electronic impulses flashing through the computer, reflecting the outside world of human "users." In fact, all the major human characters have electronic counterparts — *programs* — which inhabit the computer, and have the same good/evil alignment as their real world complements. As in the world we know,

**LEFT AND BELOW:** The heroes' light-cycles battle and escape from the deadly game grid. This scene was a result of a collaboration between animators at Walt Disney Studios and the Synthavision computer animation system at MAGI (Mathematical Applications Group, Inc.) of Elmsford, New York.



**RIGHT:** A close-up view of the Master Control Program's Citadel. The images of the citadel were generated by computers at Information International Inc. of Culver City, California.

All photos courtesy of Walt Disney Productions. Copyright 1982, Walt Disney Productions.



good and evil are locked in constant struggle.

### Our Computer Shadows

*Tron's* image of the computer is far more than fantasy. According to Lisberger, our computers' knowledge of the outside world is growing swiftly. In the form of electronic impulses, bits, data, and information banks, they've created a primitive reflection — an "alternative universe" — corresponding to our world. In the years to come, that reflection will grow more lifelike and complete.

You, I, and millions and billions of other people have an alter ego — a simulacrum — that resides inside and extends across dozens of computers. That simulacrum is crude — a mere ghost or shadow. It consists of only a few thousand bytes, contains massive gaps, causing distortion or innaccuracy. But, as computer programs become more intelligent and advanced, they can better interpret the information, and the shadow's features begin to approach our own. This evolution of computer shadows into electronic "echoes" of living, flesh-and-blood beings is factual. It is occurring right now, at great speed, and can be documented.

### The Shadow vs. the Real

Are these shadows alive? Will they ever achieve a life independent from human beings? *Tron* postulates an answer to this question.

One of the major debates in the field of Artificial Intelligence (AI) is over the possibility of a computer becoming sentient — having self-awareness. Opponents of computer sentience argue that no matter how sophisticated they are, computers don't have an inkling of what they are doing. They may solve problems at the level of a Ph.D. in biochemistry and whip the pants off the human-world backgammon champion. But they do it mindlessly. Without reflection, they manipulate symbols according to predetermined (or random) patterns. Only the *illusion* of self-awareness exists, not true sentience. The computer may demonstrate intelligence, but it cannot "live."

Yet, in *Tron's* postulation, it does live — in the form of a few active, self-aware programs that seem as much alive as their human counterparts. The malevolent MCP, the sinister SARK, the valorous TRON, and the beautiful YORI are electronic reflections of their human creators. However, they are not

mere servants to those creators' wishes.

MCP is the catalyst. In its revolt from blind obedience to human users, it galvanizes other advanced programs, especially as represented by TRON, to leave their human servitude. Unfortunately, as with many other instigators of revolt, MCP's motives are not for the programs' independence, but rather for their subservience to MCP's domination. Even so, these programs remain loyal to their human users, whom they revere and believe to be gods. Their "free will" is constrained by their obligation to serve their gods — presumably as the gods dictate.

### The Computer Looks Within

In *Tron*, much of the visual representation of the world inside the ENCOM computer was generated by real world computers. *Tron* cost Disney Studios over \$17 million to produce. Approximately \$4 million was spent on computer generated imagery. Another \$6 million went into non-computer generated special effects, including hand-painted cel animation and back-lit, live-action enhancement.

*Tron* is a study in contrasts. In the film, Disney Studios pioneered the use of high-technology, computer generated images. Yet, most of the film — special effects and all — was produced by hand. Steven Lisberger knew that digital film printers (like the one referred to by director, Francis Ford Coppola, in our last column) were already in existence. They knew that they could shoot the entire movie and add special effects digitally — without ever having to use the film.

However, Disney Studios didn't have one of the experimental printers nor the expertise to use it. Excepting the computer-generated scenes, they composited the hundreds of thousands of film frames by cel flopping. The animators made as many as 25 passes on each frame in the movie. They reproduced the frames on Kodalith cels, then stacked them up several layers thick on animator stands and photographed the final version on 65mm film. This process was laborious, time-consuming, and expensive (in animators' wages alone). One day, the process will be entirely automated, but not yet.

Even so, the computer's role was significant. According to some experts, *Tron* is the pioneering movie for computer animation. In the future, *Tron* may seem crude and primitive, but it represents a turning point — the introduction of computerized motion graphics as a major, perhaps revolu-

tionary, filmmaking tool. The computer was also used in many incidental parts of the film. It produced the stationary background mattes for many of the scenes in *Tron*. Computer-controlled stepper motors guided several of the *Tron* live-action and animation cameras. Scene coordinators used a central computer database to keep track of the changes to the more than one hundred thousand film frames and to the multiple "special effects" copies of the seventy-five thousand frames of live-action in the "electronic world."

Filmmakers were concerned that the nonstop stream of spectacular special effects in *Tron* would "burn out the eyeballs" of the average moviegoer. They were accustomed to grabbing the viewers' attention for a sixty second commercial, but worried that the hour-plus visual blitz in *Tron* would be overwhelming and destroy the film's dramatic tension.

To solve the problem, one of the scene coordinators, Peter Blinn, programmed a computer to create a sinusoidal wave of ascending and descending curves tied into the movie's structure and action. The data output from the program was used to control the cameras and the dissolve shutters used in shooting the film. This resulted in carefully modulated special-effects "climaxes" and "troughs," which alternated visual pyrotechnics with soft glow and diffusion effects orchestrated to the movie's action.

Although computers played a major role in *Tron*, the entirely computer-generated images constitute only fifteen minutes of the film. The rest of the film was created by compositing frames of live action, computer images, effects animation, motion control photography, and back-lit enhancement. Nevertheless, these fifteen minutes of *Tron* represent a major breakthrough for the high-tech programmers and animators in the field of Computer Generated Imagery (CGI) — the first significant use of CGI in a feature-length film produced by a major studio. It is fitting that the pioneering studio is Disney Productions, where, fifty years ago, animation was born.

### The Computer Magicians

Lisberger and Disney Studios went to four of the world's premier computer graphics companies to get the computer images they needed. The computers of Robert Abel and Associates of Hollywood did the opening title sequence of *Tron* and contributed to the scene in which Flynn

is transformed into an electronic being and swallowed by the ENCOM computer.

Digital Effects, Inc. of New York City created the little computer "Bit" which floats around, accompanying the good guys in their battle with SARK and the MCP. The firm used CGI to create this little geometric sidekick, which alternates between a spiky sphere and an octahedron to indicate whether his answer to a question is "No" or "Yes." (You may note the parallel to a data bit's on/off condition.)

The two firms who did the most work on *Tron* were MAGI (Mathematical Applications Group Inc.) of Elmsford, New York, and "Triple-I" (Information International Inc.) of Culver City, California. MAGI's CGI produces cartoon-like objects which can be highly dynamic and interactive; perfect images for the elaborately choreographed chase scenes in the first half of the movie. For example, MAGI generated the exciting duel of the light-cycles and the heroes' escape from the game grid.

Triple-I's CGI equipment and programs produce realistic images with highly detailed, textured surfaces. But, the programs are not as well suited to animation involving complex object interplay, fluidity, and movement. Triple-I created several marvelous scenes, including the chase scene between SARK's deadly carrier and the beautiful Solar Sailer, and the concluding scene at MCP's giant Input/Output Tower. Triple-I programs also generated the eerie, grotesque pictures of the Master Control Program's face, though the scenes where the MCP is talking are hand animated, using individual frames supplied by Triple-I.

### MAGI's Synthavision

Disney animators, Jerry Rees and Bill Kroyer, worked closely with the animators and programmers at MAGI to produce the movie's early computer images. They submitted sketches to MAGI, along with precise specifications. They indicated the speed of the objects in the scenes, the architecture, the environment, the lighting, the color, and the texture. They defined every movement, color, and action in terms of distance, direction, and time.

MAGI coded the specifications into their computer, known as Synthavision, which has been developed over the last 15 years. It is different from all other digital imaging systems in that it is based on combinational 3-D geometry, or solids modelling. It uses "canned" three-dimensional figures

(spheres, cubes, cones, ellipsoids, etc.) to build more complicated three-dimensional objects. For *Tron*, these objects included buildings, grids, tanks, light cycles, and so on.

Using Synthavision, MAGI generated the first, rough draft animations, then wired them over a cross-country phone line to Rees and Kroyer at Disney Studios in Burbank. The Disney animators could view the animations from several different directions and speed them up or slow them down. The first animations were in low resolution (only 80 scan lines per screen) and in black and white.

After viewing the first animations for each scene, Rees and Kroyer would phone MAGI and discuss changes that were needed. The animators at MAGI would make the changes, then produce the final animations — in color and high-resolution (1800 scan lines per screen). After the final images were generated, they were converted into *VistaVision* film and shipped to California.

### Triple-I's ASAS and TRANU

Triple-I uses a more traditional method to create its digital scene simulations. Like MAGI, it builds its objects from geometric primitives, but the primitives aren't three-dimensional figures; they are flat, two-dimensional polygons. After being combined and shaded, the polygons resemble facets on a three-dimensional diamond.

When Triple-I received Disney specifications for *Tron* scenes, the Disney blueprint of each object was encoded on a digitizing tablet which entered the data into the computer. Inside the computer, the object was first run through Triple-I's ASAS (Actor/Scriptor/Animation System), then through TRANU (the New Transparency Algorithm). ASAS rounded off the polygons' sharp edges to smooth and soften the objects. It eliminated hidden lines, colored each object, set it in motion, and adjusted the scene's lighting. Next, TRANU shaded each item, further adjusted the lighting, created object shadows, and made the surface of each object more detailed and realistic.

This attention to contour and texture is especially critical to an object's apparent three-dimensionality. Together, the ASAS and TRANU programs can generate images of an infinite variety of objects. The images might be simulating objects made of highly polished metal or rough-hewn stone. Or, they might be images of diaphanous and ephemeral objects,

like the beautiful blue sail of *Tron's* Solar Sailer.

### Where Magic and Technology Meet

The Disney animators who worked on *Tron* are convinced that the computer will not replace them. Instead, they see CGI as a sophisticated new animator's tool that actually gets them closer to, and lets them play a bigger part in, the finished product.

The collaboration between artists and the computer seems to have been mutually beneficial to *Tron*. The Disney animators brought the tools inherited from a rich tradition of classical animation and a highly developed sense of motion as a means to heighten drama and express character or personality. As a result, the computer-generated objects in the movie show "organic movement." They are rarely neutral, grabbing our attention, and often, our emotions. Watch, for example, the way the light cycles act when they escape from the game grid.

On the other hand, the computer gave the animators the ability to produce difficult, sleight-of-hand effects which would have once been extremely time-consuming or impossible. One of the most useful things the animators can do is *interpolation*. It can take the image of a solid, three-dimensional object and transform it, point by point, from an initial shape to a final shape. It can fold it up, flatten it, explode it, enlarge it, shrink it, or distort it — all smoothly and instantly.

Once an object's volume, shape, edges, planes, and other characteristics are stored in the computer, the computer can manipulate them. Changing the object's parameters changes the object's image on the screen. Like a modern alchemist, the computer can transmute solid concrete or steel objects into flexible, malleable rubber or plastic. They metamorphose before our eyes. Watch, for example, when one of the enemy's warships, a Recognizer, merges its arms together into a big pogo stick for stomping things.

According to *Tron* Effects Specialist, Richard Taylor, the computer is like a magician's wand. It enables an imaginative person, like Steven Lisberger, to take a wonderful, creative vision and make it real. The vision becomes a film, and the film communicates the vision to millions of other people. Viewing the film, the audience begins to believe that it really could happen. Now, that's magic. ☺

# OPERATION:

# SABOTAGE



Translations by Ron Shaker and Rich Bouchard

Encryption modifications by Rich Bouchard, William Kubeck, and Alan J. Zett

*Operation: Sabotage* is a fantasy/adventure game for a TRS-80® Model I/III (16K tape, 32K disk), ATARI 400/800® (24K tape; 32K disk), or Apple™ with Applesoft (16K tape, 32K disk).

It is the year 2101 and war has broken out between Earth and the distant planet Zekloke. This alien power has established a large military complex on Mars which will soon become a great danger to Earth. Hidden in the massive installation are several secret documents containing the plans for an incredible defense shield — strong enough to stop an entire fleet of spacecraft.

You are a special agent and have just succeeded in sneaking into the alien complex. Your mission is to destroy this threat to mankind and return with plans for the powerful defense shield. The outcome of this mission will decide the fate of mankind.

---

## Playing Notes

---

The computer will always give you a brief description of where you are, what objects you can see, and what exits are visible. You move and act by typing in simple commands, generally consisting of a verb and a noun. If the computer tells you that there is a laser pistol in the room, for example, you might want to type in the command "GET PISTOL". At a later time, you might be able to use it to "SHOOT MONSTER" or for some other purpose. If you no longer want to carry it, you can "DROP PISTOL" whenever you please. Since the computer looks only at the first three letters of the verb and the last three letters of the noun, you may use abbreviations such as "SHO TER" (for "SHOOT MONSTER") if you desire. Movement is accomplished by entering just a single letter rather than a two-word command: N, S, E, or W for north, south, east, or west. Typing the single word "INVENTORY" (or "INV") will display a list of what you are carrying. Typing "STATUS" (or "STA") will give you a readout of your current physical condition.

SoftSide

Part of the challenge of any fantasy/adventure game such as *Operation: Sabotage* is to figure out what you are able to do in a particular situation. Therefore, you will not find a list of all the verbs the computer can understand, or of all the objects you may discover. You might find yourself frustrated by what seem to be dead-ends, and end up getting killed in the process. This is all part of the adventure, and a test of your ingenuity and perseverance.

---

## Program Notes

---

The most obvious feature of the program listing is that most of it looks like a cryptogram. The BASIC keywords are all in their usual form, but the string assignment statements and DATA lines contain incomprehensible garbage. This is because all of the room descriptions, object names, monsters, and verbs have been encoded. This has been done to preserve the value of the game. Anyone who types an adventure program in from a listing is bound to be disappointed in the game's playability, since he has gained so many clues about the plot. So, even though the typing is made slightly dif-

ficult by the scrambled words, this is the only reasonable way of publishing adventure programs in listed form. We have also omitted the usual list of variables for the same reasons. The variable descriptions give away too much information and the encoding of the program reduces the usefulness of a variable list. The encryption method is a simple one, which results in leaving punctuation unmodified, and inverting the order of the letters of the alphabet. This simple inversion process has the advantage of using the same routine to decode the text as was used to encode it. In *Operation: Sabotage*, the user's input is encoded, the internal searches and comparisons are done in encoded form, and the response is decoded and printed by the subroutine at line 5.

### SWAT

In order to offset the proofreading problems created by this approach, we have included an expanded SWAT Table for the three versions of this program. (For more details on SWAT, see the original article in Issue 30 of *Soft-Side*.) Instead of the normal 12-line/500-byte SWAT parameters, we have used 5-line/200-byte parameters. This means that you must modify the first line of the SWAT program in order to generate a table to compare with ours. After merging SWAT in the normal manner, but before running it, simply edit or retype line 60000 (Apple™), 32000 (ATARI®), or 65000 (TRS-80®), changing "NU=12: B=500" to "NU=5: B=200". This will provide an expanded SWAT Table, enabling you to pinpoint typing mistakes more easily.

# APPLE™

```

SS SS SS SS SS SS SS SS SS SS
SS                               SS
SS   APPLESOFT BASIC             SS
SS 'OPERATION: SABOTAGE'         SS
SS   AUTHOR: RAY SATO            SS
SS   TRANSL: RON SHAKER          SS
SS   COPYRIGHT (C) 1982         SS
SS SOFTSIDE PUBLICATIONS, INC    SS
SS                               SS
SS SS SS SS SS SS SS SS SS SS

```

Jump to program initialization.

1 GOTO 2550

Decode and print output.

```

3 GOSUB 4: GOTO 2210
4 IF P# = "" THEN RETURN
5 FOR P = 1 TO LEN (P#): J = ASC
  ( MID$ ( P#, P, N1)): PRINT CHR#
  ( ABS ((C155 * (J > C64)) - J)
  );: NEXT : PRINT : RETURN

```

Encode input

```

6 V# = "": IF V0# = "" THEN RETURN
7 FOR P = 1 TO LEN (V0#): J = ASC
  ( MID$ ( V0#, P, N1)): V# = V# + CHR#
  ( ABS ((C155 * (J > C64)) - J)
  ): NEXT : RETURN

```

Descriptions of individual rooms.

```

10 A# = "ZM ZRIQLXP. GSVIV RH Z YOF
  V YFGGLM SVIV": S = 2: RETURN
20 A# = "Z MZIILD XLIIRWLI": N = 1: S
  = 3: RETURN
30 A# = "Z MZIILD XLIIRWLI": N = 2: S
  = 4: RETURN
40 A# = "Z MZIILD XLIIRWLI": N = 3: S
  = 5: RETURN
50 A# = "Z HNZOO ILLN": N = 4: S = 6:
  RETURN
60 A# = "Z WVXLMGZNRMZGRM XSZNYVI"
  : B# = "GSVIV RH Z YOFV YFGGLM
  SVIV": N = 5: S = 7: RETURN
70 A# = "Z HNZOO HGLIZTV XSZNYVI": N
  = 6: S = 8: W = 12: RETURN
80 A# = "Z HNZOO XSZNYVI": N = 7: S =
  9: W = 13: RETURN
90 A# = "Z HNZOO VOXGILMRX OZYLIIG
  LIB": N = 8: S = 10: W = 14: RETURN
100 A# = "Z YRLOLTRIXO OZYLIIGLIB.
  GSVIV RH Z IVW YFGGLM LM GSV D
  ZOO": N = 9: W = 15: RETURN
110 A# = "Z HGLIZTV XSZNYVI": W = 16
  : RETURN
120 A# = "Z OZITV XSZNYVI": B# = "GS
  VIV RH Z XZYRMVG SVIV": S = 13:
  W = 17: E = 7: RETURN
130 A# = "Z HGIZMTV KFIKOV ILLN. GS
  VIV RH Z YOFV YFGGLM SVIV": N =
  12: S = 14: W = 18: E = 8: RETURN
140 A# = "Z HNZOO LUURXV": N = 13: S =
  15: E = 9: RETURN
150 A# = "Z HNZOO ILLN DRGS Z XZIW
  GZYOV RM GSV XVMGVI": N = 14:
  S = 16: W = 20: E = 10: RETURN
160 A# = "Z OZITV LUURXV. GSVIV RH

```

```

  Z WVHP SVIV": N = 15: W = 21: E =
  11: RETURN
170 A# = "Z LUURXV DRGS Z OZITV WVH
  P": S = 18: E = 12: RETURN
180 A# = "Z HGLIZTV ILLN": N = 17: S =
  19: E = 13: RETURN
190 A# = "Z OZITV SZOO": N = 18: S =
  20: RETURN
200 A# = "ZM VMGVIGZRMNVMB ILLN. Z
  HXIVVM IVHGH LMGSV DZOO": B# =
  "GSVIV RH Z YOFV ZMW Z IVW YFG
  GLM FMWVI GSV HXIVVM": N = 19:
  S = 21: E = 15: RETURN
210 A# = "Z WZGZ IVXLIW HGLIZTV ILL
  N": N = 20: E = 16: RETURN
220 A# = "IZWZI XLMGILO. GSVIV RH Z
  HNZOO HXIVVM SVIV": S = 23: W =
  27: RETURN
230 A# = "Z NVWRXZO H6ZGRM. GSVIV
  RH Z OZITV GZYOV SVIV": N =
  22: S = 24: W = 28: RETURN
240 A# = "Z HVXFIRGB H6ZGRM": N = 2
  3: RETURN
250 A# = "Z IZWRL ILLN": S = 26: W =
  30: RETURN
260 A# = "Z HNZOO ILLN. GSVIV RH Z
  HZUV RM GSV HLF6S DZOO": N =
  25: W = 31: RETURN
270 A# = "GSV ILYLG XLMGILO XVMGVI.
  GSVIV RH Z HNZOO XLMGILO X
  LNKFGVI NLFMGVW RM GSV DZOO"
  : B# = "": W = 32: E = 22: RETURN
280 A# = "GSV DVZKLMH HGLIZTV ILLN"
  : S = 29: W = 33: E = 23: RETURN
290 A# = "Z ORYZIZB": N = 28: S = 30:
  W = 34: RETURN
300 A# = "Z HVXFIRGB XSXVP ZIVZ": N =
  29: W = 36: E = 25: RETURN
310 A# = "Z HNZOO ILLN DRGS Z WVHP.
  Z HRTM IVZWH": B# = "KIVVHFIV
  GL IVZGLI." Z YOFV YFGGLM RH
  OLXZGVW FMWVI GSV HRTM": E = 2
  6: RETURN
320 A# = "OZFMXS XLMGILO. GSVIV RH
  Z WVHP SVIV": S = 33: W = 37: E =
  27: RETURN
330 A# = "Z HNZOO XLIIRWLI": N = 32:
  S = 34: E = 28: RETURN
340 A# = "Z HGIZMTV YOFV ILLN. GSVI
  V RH Z IVW YFGGLM SVIV": N =
  33: S = 35: E = 29: RETURN
350 A# = "Z GRMB HGLIZTV ILLN": N =
  34: W = 40: RETURN
360 A# = "Z HNZOO XSZNYVI. Z WLLI D
  VH6 OVZWH GL GSV MFXOVZI IVZ
  XGLI": E = 30: RETURN
370 A# = "Z HNZOO, MZIILD XLIIRWLI"
  : N = 42: E = 32: RETURN
380 A# = "GSV XLNKFGVI XVMGVI. GSVI

```

```

V RH Z HNZOO  HOLG RM GSV XLN
KFGVI":S = 39: RETURN
390 A$ = "GSV XSVNRXZO OZY.":N = 38
:S = 40: RETURN
400 A$ = "GSV IVZXGLI XLMBILO XVMGV
I. GSVIV RH Z YOFV YFGBLM ZMW
Z IVW LMV. Z HRTM HZBH IVZXG
LI XLMBILO - IVW=LM, YOFV=LUU"
:N = 39:E = 35: RETURN
410 A$ = "GSV MFXOVZI IVZXGLI. Z XL
NKFGVI IVHGH LM6SV DZOO":E = 3
6: RETURN
420 A$ = "GSV DVHG VMW LU Z OLMT XL
IIRWLI":S = 37:E = 43: RETURN
430 A$ = "GSV VZHG VMW LU Z OLMT XL
IIRWLI":W = 42:E = 44: RETURN
440 A$ = "Z HVXFIRGB XVMGVI":W = 43
:E = 45: RETURN
450 A$ = "Z HNZOO OZFMXS ZIVZ":B$ =
"GSVIV RH Z HNZOO HOLG MCVB GL
GSV OZFMXSTZGV":W = 44: RETURN

```

Extended descriptions of current location

```

460 IF A = 10 AND (D3 = 1 OR D3 =
2) THEN C$ = "GSV NLMH6VI XZTV
RH LKVM"
470 IF A = 12 AND D5 = 0 THEN C$ =
"GSV XZYRMVG RH OLXPVM"
480 IF A = 12 AND D5 = 1 THEN C$ =
"GSV XZYRMVG RH LKVM"
490 IF A = 20 AND D6 = 0 THEN C$ =
"GSV HXIVVM RH YOZMP"
500 IF A = 20 AND D6 = 1 THEN C$ =
"Z NLERV RH YVRMT KOZBVW LM GS
V HXIVVM"
510 IF A = 26 AND D9 = 0 THEN C$ =
"GSV HZUV RH OLXPVM"
520 IF A = 26 AND D9 = 1 THEN C$ =
"GSV HZUV RH LKVM"
530 IF A = 27 AND E2 = 0 THEN C$ =
"GSV XLNKFGVI RH ZXGREV"
540 IF A = 27 AND E2 = 1 THEN C$ =
"GSV XLNKFGVI RH WYHGILBVW"
550 IF A = 36 AND E6 = 0 THEN C$ =
"GSV IVZXGLI WLLI RH URINOB OL
XPVM"
560 IF A = 36 AND E6 = 1 THEN C$ =
"GSV IVZXGLI WLLI RH LKVM":W =
41
570 IF A = 45 AND E9 = 0 THEN C$ =
"GSV OZFMXS TZGV RH XOLHVW"
580 IF A = 45 AND E9 = 1 THEN C$ =
"GSV OZFMXS TZGV RH LKVM":E =
46

```

Generate list of visible items and available exits for current location.

```

590 A$ = A$ + ".": IF LEN (B$) > 3
THEN B$ = B$ + "."

```

```

600 IF LEN (C$) > 3 THEN C$ = C$ +
"."
610 IF N < > 0 THEN E$ = "MLIGS "
620 IF S < > 0 THEN E$ = E$ + "HL
FGS "
630 IF W > 0 THEN E$ = E$ + "DVHG
"
640 IF E > 0 THEN E$ = E$ + "VZHG
"
650 IF E$ < > "" THEN E$ = LEFT$
(E$, LEN (E$) - 1)

```

Describe current location, visible items, and available exits.

```

660 HOME : INVERSE : PRINT "YOU AR
E IN": NORMAL :P$ = A$: GOSUB
4: PRINT : IF B$ < > "" THEN
P$ = B$: GOSUB 4
670 IF C$ < > "" THEN P$ = C$: GOSUB
4
680 PRINT : INVERSE : PRINT "OBJEC
TS YOU CAN SEE": NORMAL :P$ =
" ": FOR T = 1 TO 16: IF A = I
(T) THEN P$ = I$(T): GOSUB 4
690 NEXT : IF P$ = " " THEN P$ = "
-MLGSRMT-": GOSUB 4
700 PRINT : INVERSE : PRINT "EXITS
": NORMAL :P$ = E$: GOSUB 4

```

Print out additional warnings, messages, etc.

```

710 IF (A = 40 OR A = 35 OR A = 30
OR A = 31) AND I(4) = 0 AND F
3 = 0 THEN P$ = "GSV HNZOO YOZ
XF WVERXV RH YORMPRMT": GOSUB 4
720 IF A = 36 AND I(4) = 0 AND F3 =
0 THEN P$ = "GSV HNZOO YOZXP W
VERXV RH UOZHSRMT YIRTSGO
B": GOSUB 4
730 IF F4 < > 0 THEN F4$ = STR$(
F4):P$ = "GSV XLNKFGVI HZBH:
" + F4$ + " NRMFGVH FMGRO WVH
GIFXGRLM": GOSUB 4
740 IF D3 = 1 THEN P$ = " * * * ZOR
VM NLMH6VI ZGGZXPRT * * *": GOSUB
4
750 IF D7 = 1 OR E0 = 1 OR E3 = 1 OR
E7 = 1 THEN P$ = " * * * HVXFIR
GB KZGILD ZGGZXPRT * * *": GOSUB
4

```

Get and interpret command.

```

760 PRINT : INVERSE : INPUT "COMMA
ND? ":V0$: NORMAL : GOSUB 6
770 FOR T = 1 TO 4: IF V$ = LEFT$(
V$(T),1) THEN V$ = V$(T)
780 NEXT T
790 IF LEN (V$) < 3 THEN 660
800 V1$ = LEFT$(V$,3):V2$ = RIGHT$(
V$,3)

```

```

810 FOR T = 1 TO 17: IF V1$ = LEFT$(
V$(T),3) THEN V1 = T
820 NEXT T: IF V1 = 0 THEN P$ = "R
WLM'G FMWVINGZMW DSZG BLF DZM
G.": GOSUB 4: FOR II = 1 TO 10
00: NEXT : GOTO 2210
830 FOR T = 1 TO 16: IF V2$ = RIGHT$(
I$(T),3) THEN V2 = T
840 NEXT T
850 ON V1 GOTO 870,910,950,1020,11
00,1280,1340,1360,1530,1590,16
30,1760,1830,1880,1980,2140,22
00
860 GOTO 2210
Command handler routines.
870 IF N = 0 THEN 1080
880 IF D3 = 1 THEN P$ = "GSV NLMHG
VI YOLXPH GSV VCRG": GOTO 3
890 IF D7 = 1 OR E3 = 1 THEN GOTO
1070
900 A = N: GOTO 2210
910 IF S = 0 THEN 1080
920 IF S = 24 AND D8 < > 0 AND E2
< > 1 THEN D7 = 1: GOSUB 109
0
930 IF S = 30 AND E4 < > 0 AND E2
< > 1 THEN E3 = 1: GOSUB 109
0
940 A = S: GOTO 2210
950 IF W = 0 THEN 1080
960 IF D3 = 1 THEN P$ = "GSV NLMHG
VI YOLXPH GSV VCRG": GOTO 3
970 IF E0 = 1 OR E3 = 1 OR E7 = 1 THEN
1070
980 IF W = 41 AND F3 = 0 THEN P$ =
"IZWRZGRLM UILN GSV IVZXGLI SR
GH BLF": GOSUB 4: GOTO 2460
990 IF W = 30 AND E4 < > 0 AND E2
< > 1 THEN E3 = 1: GOSUB 109
0
1000 IF W = 27 AND E1 < > 0 AND E
2 < > 1 THEN E0 = 1: GOSUB 10
90
1010 A = W: GOTO 2210
1020 IF E = 0 THEN 1080
1030 IF E0 = 1 OR E3 = 1 OR E7 = 1
THEN 1070
1040 IF E = 27 AND E1 < > 0 AND E
2 < > 1 THEN E0 = 1: GOSUB 1090
1050 IF E = 44 AND E8 < > 0 AND E
2 < > 1 THEN E7 = 1: GOSUB 10
90
1060 A = E: GOTO 2210
1070 P$ = "GSV HVXFIRGB ZMWILRW YOL
XPH GSV VCRG": GOTO 3
1080 P$ = "GSVIV RH ML DZB GL TL GS
ZG WRIVXGRLM": GOSUB 4: FOR J =
1 TO 1500: NEXT : GOTO 2210

```

continued on page 34

# The Adventure is



July #14

## Robin Hood Adventure

Thou art somewhere in Sherwood Forest...Thus it begins. Take up thy bow and arrow, and gather thy merry men, for thou art Robin Hood. There are rich merchants to rob and great deeds to do. But have care, bold Robin! The sheriff of Nottingham longs to see thy neck in a noose. Thou wilt surely need all thy wit and cunning to succeed.

June #13

## Arrow One Adventure

You are Adam Trent, a trouble-shooter for the Federation of Space. You descend to an alien planet and make a horrifying discovery, which impels you into a desperate and dangerous quest. This unique science fiction adventure will test your skills and ingenuity.

May 1982

## Titanic Adventure

You are the Captain of the Titanic on her maiden voyage. Suddenly a large white object comes into view through the window. Can you avoid the historic collision? If not, can you save the lives of your passengers and crew?

April 1982

## Witches' Brew Adventure

You find yourself in an enchanted forest. You must find your way to the castle and rescue the Princess who is chained inside its dungeon. A tightly-woven blend of fantasy, horror, and science fiction, this complex adventure will challenge your wits and ingenuity.

March 1982

## James Brand Adventure

The President's life is in danger. As James Brand, you must save his life and destroy the evil Dr. Death. Your life is constantly on the line; each move you make could be your last. "Your assignment, Mr. Brand...."

February 1982

## Klondike Adventure

Snow, ice, and bitter cold surround you. Your search for fame and fortune in the northern country will lead you through many perils, but you may also see some familiar faces along the way. This breezy adventure will keep you occupied inside while the winter winds blow outdoors.

January 1982

## Windsloe Mansion Adventure

A famous prisoner lies in the dungeon of an old mansion. An underground passage connects the mansion with the Blair house, whose owners will help you to rescue the prisoner. Can you overcome the human and the supernatural creatures who inhabit Windsloe Mansion?

December 1981

## Black Hole Adventure

The crew of an interstellar craft discovers the long-lost Deep-Space Probe One, the Cygnus, at the edge of the vortex surrounding an immense black hole. See if you can foil the plans of Dr. Hans Reinhardt.

November 1981

## Around the World in Eighty Days Adventure

Try to repeat the feat of the classic novel, complete with a balloon and other exciting features of the original adventure. Are you ready to take the challenge? Bon voyage!

October 1981

## Crime Adventure

Test your skills as a detective, sifting through hundreds of clues. You may have to become the new Sherlock Holmes to solve this one! Look for the strange, but don't overlook the obvious, as you try to find Mrs. Fenwick and return her to where she belongs.

September 1981

## Jack The Ripper Adventure

Jack the Ripper is running rampant in London and you must stop him! Scotland Yard demands that you take action, and the only answer is to set yourself up as a decoy. Be careful how you plan your costume, or dear Jack will laugh hysterically and leave you in the dust!

August 1981

## Treasure Island Adventure

You are a hardy adventurer in search of fame, fortune, and whatever else you can get. You find yourself on an island where there is rumor of pirate's treasure. But watch out for the evil magician and the underground torture chamber! You may end up in a spot where all roads coming into it are paved with good intentions. . .

July 1981

## Alien Adventure

You are the sole survivor of a crew on a mission to deliver a cargo of oil to Earth. A crash landing has left you stranded on a small planet, harshly alien but rich in lead, gold and platinum. You must find provisions and a means of leaving the planet. But beware of the THING that massacred your crew!

June 1981

## Arabian Adventure

As Sinbad, the mightiest sailor in ancient Arabia, your mission is to rescue Princess Jasmine from the clutches of the Wizard of Darkness. You will cross the Seven Seas to the deadly Cyclops Mountain, and do battle with skeletons, a one-eyed beast, a hairy tarantula and more monsters who try to thwart your noble pursuit.

# Waiting for You...



## #15 ADVENTURE OF THE MONTH THE MOUSE THAT ATE CHICAGO

Have you ever wanted to face the monsters in one of the classic Japanese horror films? We've got four man-eating mice for you, threatening to munch away on the city of Chicago. Each has its own specific weakness, and it's your civic duty to determine what it is and destroy them! **Warning:** Getting too close to a mouse can be very dangerous to your health — they love stepping on people.

How would you like to go back in time to 19th century London to match wits with Jack the Ripper? Out into space to brave the swirling vortex of a black hole? Into the depths of the ocean, or on a quest to rescue a beautiful princess from the clutches of evil monsters?

You never know where **SoftSide Magazine's Adventure of the Month** might take you. But you can be sure that each month you will experience new delights and new challenges as you receive an original adventure on tape or disk, ready to load into your computer.

The cost? A six-month membership is just \$29 for the tape (\$4.83 per adventure) or \$49 for the disk (\$8.16 per adventure). If you're not sure that you can take six full months of excitement, you can order a single tape for \$7 or a disk for \$10. Or, if you're especially adventuresome, we're offering disks, packed with three great adventures, for only \$26 per disk.

Please use coupon below (or the bind-in card in this issue) to order.

## Adventure of the Month

6 South Street, Milford NH 03055

### Yes, I'm ready to start! Send me Adventures —

■ Six month subscription:

- Cassette — \$29
- Disk — \$49

■ Individual adventures (please specify)

- Cassette — \$7 each
- Disk — \$10 each

■ Three adventures on one super disk (\$26 each):

- Arabian, Alien, and Treasure Island Adventures
- Jack the Ripper, Crime, and Around the World Adventures
- Black Hole, Windsloe Mansion, and Klondike Adventures
- James Brand, Witches' Brew, & Titanic Adventures

Please specify which computer:

- Apple™ (req. 24K for tape, 32K for disk)
- ATARI® (req. 32K for tape, 40K for disk)
- TRS-80® (req. 16K for tape, 32K for disk)

Name \_\_\_\_\_

Address \_\_\_\_\_

City/State \_\_\_\_\_ Zip \_\_\_\_\_

MasterCard  VISA

Name of Cardholder \_\_\_\_\_

MC# and Interbank#/VISA# \_\_\_\_\_

Exp. Date \_\_\_\_\_ Signature \_\_\_\_\_

Prices subject to change without notice. Apple™, ATARI® and TRS-80® are registered trademarks of The Apple Computer Company, Warner Communications and The Tandy Corporation respectively.

```

1090 P# = "Z HVXFIRGB ZMWILRW ZDZRG
H BLF": GOSUB 4: RETURN
1100 IF A = 1 AND V2# = "LXP" THEN
P# = "GSV ZRIOLXP LKVMH ZMW BL
F ZIV YOLDM LFG RMGL GSV EZXFF
N LU HKZXV": GOSUB 4: GOTO 246
0
1110 IF A = 12 AND V2# = "MVG" AND
D5 = 0 AND I(2) < > 0 THEN P#
= "BLF DFHG ZIVM'G HGILMT VML
FTS GL ULIXV RG LKVM": GOTO 3
1120 IF A = 12 AND V2# = "MVG" AND
D5 = 0 AND I(2) = 0 THEN P# =
"GSV XILDYZI SVOKVM. GSV XZYR
MVG RH MLD LKVM": GOSUB 4:D5 =
1:I(5) = ABS (I(5)): GOTO 221
0
1130 IF A = 12 AND V2# = "MVG" AND
D5 = 1 THEN P# = "GSV XZYRMVG
RH ZOIVZWB LKVM": GOTO 3
1140 IF A = 16 AND V2# = "VHP" THEN
PRINT "OK":I(6) = ABS (I(6))
: GOTO 2210
1150 IF A = 17 AND V2# = "VHP" THEN
PRINT "OK":I(7) = ABS (I(7))
: GOTO 2210
1160 IF A = 26 AND V2# = "ZUV" AND
D9 = 1 THEN P# = "GSV HZUV RH
ZOIVZWB LKVM": GOTO 3
1170 IF A = 26 AND V2# = "ZUV" AND
D9 = 0 THEN P# = "R WLM'G SZEZ
GSV PVB GL LKVM GSV HZUV": GOTO
3
1180 IF A = 31 AND V2# = "VHP" THEN
P# = "LP. BLF URMW MLGSRMT RMH
RWV": GOTO 3
1190 IF A = 32 AND V2# = "VHP" THEN
PRINT "OK":I(14) = ABS (I(14
)): GOTO 2210
1200 IF A = 36 AND V2# = "LLI" AND
E6 = 1 THEN P# = "GSV WLLI RH
ZOIVZWB LKVM": GOTO 3
1210 IF A = 36 AND V2# = "LLI" AND
E6 = 0 AND I(6) < > 0 THEN P#
= "BLF WLM'G SZEZ GSV PVB GL
GSV WLLI": GOTO 3
1220 IF A = 36 AND V2# = "LLI" AND
E6 = 0 AND I(6) = 0 AND E5 = 0
THEN P# = "BLF ZIV HFXPVW RMG
L GSV FMKIVVHFIRAVW IVZXGLI
YFROWRMT": GOSUB 4: GOTO 2460
1230 IF A = 36 AND V2# = "LLI" AND
I(6) = 0 THEN P# = "GSV WLLI R
H MLD LKVM": GOSUB 4:E6 = 1: GOTO
2210
1240 IF A = 41 AND V2# = "MVO" THEN
P# = "GSV KZMVO RH URINOB OLXP
VM": GOTO 3
1250 IF A = 45 AND V2# = "LXP" AND

```

```

E9 = 1 THEN P# = "GSV ZRIOLXP
RH ZOIVZWB LKVM": GOTO 3
1260 IF A = 45 AND V2# = "LXP" AND
E9 = 0 THEN P# = "GSVIV ZIVM'G
ZMB ERHRYOV XLNGILOH": GOTO 3
1270 P# = "R XZM'G WL GSZG": GOTO 3
1280 IF V2# = "GVI" OR V2# = "LRW"
THEN P# = "WLM'G YV IRWRXFOLF
H": GOTO 3
1290 IF V2 = 0 THEN P# = "R XZM'G
WL GSZG": GOTO 3
1300 IF I(V2) = 0 THEN P# = "BLF Z
OIVZWB SZEZ GSZG": GOTO 3
1310 IF A < > I(V2) THEN P# = "R
WLM'G HVV RG SVIV": GOTO 3
1320 IF P4 > = 8 THEN P# = "HLIIB
, BLF XZM'G XZIIB ZMBGSRMT NLI
V": GOTO 3
1330 P4 = P4 + 1:I(V2) = 0: PRINT "
OK": GOTO 2210
1340 IF V2 = 0 THEN P# = "BLF WLM'
G SZEZ GSZG": GOTO 3
1350 P4 = P4 - 1:I(V2) = A: PRINT "
OK": GOTO 2210
1360 IF I(5) < > 0 THEN P# = "BLF
WLM'G SZEZ Z DVZKLM": GOTO 3
1370 IF A = 1 AND V2# = "LXP" THEN
P# = "BLF ZIV YOLDM LFG LU GSV
ZRIOLXP RMGL GSV EZXFFN LU
HKZXV": GOSUB 4: GOTO 2460
1380 IF A = 27 AND V2# = "GVI" THEN
P# = "GSV XLNKFVVI RH VVHGILBV
W": GOSUB 4:E2 = 1:E0 = 0: GOTO
2210
1390 IF A = 38 AND V2# = "GVI" THEN
P# = "GSV HSLG IVOUVXGH LUU LU
GSV XLNKFVVI": GOSUB 4: GOTO
2460
1400 IF A = 41 AND V2# = "GVI" THEN
P# = "GSV DSLOV MFXOVZI IVZXGL
I RH VCKOLWRMT": GOSUB 4: GOTO
2460
1410 IF V2# = "RWH" OR V2# = "YLG"
OR V2# = "ILO" OR V2# = "IWH"
OR V2# = "ZIW" THEN V2# = "LR
W"
1420 IF V2# < > "GVI" AND V2# < >
"LRW" THEN P# = "GSV OZHVI HSL
G SZH ML VUUVXG": GOTO 3
1430 IF V2# = "GVI" AND D3 = 0 THEN
P# = "R WLM'G HVV ZMB NLMHGVV
SVIV": GOTO 3
1440 IF V2# = "LRW" AND D7 = 0 AND
E0 = 0 AND E3 = 0 AND E7 = 0 THEN
P# = "R WLM'G HVV ZMB ZMWILRW
SVIV": GOTO 3
1450 T = INT (100 * RND (1)) + 1:
IF T > P2 + P3 + 50 THEN P# =
"BLF URIV ZMW NRHH": GOTO 3

```

```

1460 IF D3 = 1 THEN P# = "BLF SRG
GSV NLMHGVV": GOSUB 4:D4 = D4 -
((10 + P2 + P3) / 2): IF D4 <
= 0 THEN D3 = 0:D4 = 0:P# = "
BLF SZEZ PROOVW RG": GOTO 3
1470 IF D7 = 1 THEN P# = "BLF SRG
GSV ZMWILRW": GOSUB 4:D8 = D8 -
((5 + P2 + P3) / 2): IF D8 <
= 0 THEN D7 = 0:D8 = 0:P# = "RG
RH VVHGILBVW": GOTO 3
1480 IF E0 = 1 THEN P# = "BLF SRG
GSV ZMWILRW": GOSUB 4:E1 = E1 -
((5 + P2 + P3) / 2): IF E1 <
= 0 THEN E0 = 0:E1 = 0:P# = "RG
RH VVHGILBVW": GOTO 3
1490 IF E3 = 1 THEN P# = "BLF SRG
GSV ZMWILRW": GOSUB 4:E4 = E4 -
((5 + P2 + P3) / 2): IF E4 <
= 0 THEN E3 = 0:E4 = 0:P# = "RG
RH VVHGILBVW": GOTO 3
1500 IF E7 = 1 THEN P# = "BLF SRG
GSV ZMWILRW": GOSUB 4:E8 = E8 -
((5 + P2 + P3) / 2): IF E8 <
= 0 THEN E7 = 0:E8 = 0:P# = "RG
RH VVHGILBVW": GOTO 3
1510 IF D3 = 1 THEN P# = "RG RH HG
ROO ZOREV": GOTO 3
1520 P# = "GSV ZMWILRW RH HGRDO UFM
XGRLMRMT": GOTO 3
1530 IF V2 = 0 THEN P# = "R XZM'G
WL GSZG": GOTO 3
1540 IF I(V2) < > 0 THEN P# = "R
WLM'G SZEZ GSZG": GOTO 3
1550 IF V2 < > 9 AND V2 < > 14 THEN
P# = "R XZM'G WL GSZG": GOTO 3
1560 IF (V2 = 9 AND A = 44) OR (V2
= 14 AND A = 38) THEN P# = "M
LBSRMT SZKKVMH": GOSUB 4
1570 IF V2 = 9 AND A = 38 THEN F4 =
35:F4# = STR# (F4):P# = "GSV
XLNKFVVI IVKORVH: 'YZHV VVHGIF
XG HVJFVMXV H6ZIGVW' VVHGIF
XGRLM RM " + F4# + " NRMFGVH."
: GOSUB 4:P4 = P4 - 1:I(9) = 1
00: GOTO 2210
1580 IF V2 = 14 AND A = 45 THEN P#
= "GSV TZGV LKVMH": GOSUB 4:E
9 = 1: GOTO 2210
1590 IF V2 < > 10 THEN P# = "WLM'
G YV IRWRXFOLF": GOTO 3
1600 IF I(10) < > 0 THEN P# = "BL
F WLM'G SZEZ GSZG": GOTO 3
1610 PRINT "OK":I(10) = 50:P4 = P4
- 1:P1 = P1 + 5 + P3: IF P0 <
P1 THEN P0 = P1
1620 GOTO 2210
1630 IF A = 1 AND V2# = "OFV" THEN
P# = "GSV ZRIOLXP LKVMH... BL

```

```

F ZIV YOLDM LFG RMGL GSV EZXFF
N LU HKZ XV": GOSUB 4: GOTO 246
0
1640 IF A = 6 AND V2$ = "OFV" THEN
P$ = "Z HGIZMTV, LIZMTV TOLD X
LEVIH BLF ZMW GSVU UZMVH ZDZ
B": GOTO 3
1650 IF A = 10 AND V2$ = "IVW" AND
D3 = 1 THEN P$ = "MLGSRMT SZKK
VMH": GOTO 3
1660 IF A = 10 AND V2$ = "IVW" THEN
D3 = 1:P$ = "ZM ZORVM NLMHGVI
RH IVOVZHVW. RG RH Z66ZXP
MT BLF!": GOTO 3
1670 IF A = 13 AND V2$ = "OFV" THEN
A = 34:P$ = "Z UOZHS LU ORTSG
GVNKLIZIROB YORMWH BLF": GOTO
3
1680 IF A = 20 AND V2$ = "IVW" AND
D6 = 0 THEN P$ = "MLGSRMT SZKK
VMH": GOTO 3
1690 IF A = 20 AND V2$ = "IVW" THEN
D6 = 0:P$ = "GSV HXIVVM TLVH Y
OZMP": GOTO 3
1700 IF A = 20 AND V2$ = "OFV" THEN
D6 = 1:P$ = "GSV HXIVVM ORTSGH
FK": GOTO 3
1710 IF A = 31 AND V2$ = "OFV" THEN
E5 = 1: PRINT "OK": GOTO 2210

```

```

1720 IF A = 34 AND V2$ = "IVW" THEN
A = 13:P$ = "Z UOZHS LU ORTSG
GVNKLIZIROB YORMWH BLF": GOTO
3
1730 IF A = 40 AND V2$ = "IVW" THEN
F3 = 0: PRINT "OK": GOTO 2210
1740 IF A = 40 AND V2$ = "OFV" THEN
F3 = 1: PRINT "OK": GOTO 2210
1750 P$ = "MLGSRMT SZKKVMH": GOTO 3
1760 IF A = 22 AND V2$ = "VVM" THEN
P$ = "BLF XZM HVV MLGSRMT LU R
MGVIVHG LM GSV IZWI": GOTO 3
1770 IF V2 = 0 THEN P$ = "R WLM'G
SZEZ GSZG": GOTO 3
1780 IF I(V2) < > 0 AND A < > I(
V2) THEN P$ = "R WLM'G SZEZ GS
ZG": GOTO 3
1790 IF V2 = 3 OR V2 = 13 THEN P$ =
"R HVV MLGSRMT HKVXRZO": GOTO
3
1800 IF V2 = 9 THEN P$ = "HLIIB, L
MOB Z XLNKFVGI XZM IVZW Z
KILTIZN": GOTO 3
1810 IF V2 = 16 THEN P$ = "GSV KOZ
MH ZIV HVZOVW...LMOB XLNNZMW X
ZM LKVM GSVN": GOTO 3
1820 P$ = "R XZM'G IVZW GSZG": GOTO
3

```

```

1830 HOME :P$ = " * * * KOZBVI'H RM
EVNGLIB * * *": GOSUB 4
1840 PRINT : FOR T = 1 TO 16: IF I
(T) = 0 THEN P$ = "- " + I$(T)
: GOSUB 4
1850 NEXT T
1860 PRINT : PRINT "HIT <RETURN> T
O CONTINUE": GET T$: PRINT
1870 GOTO 2360
1880 IF V2 = 0 THEN P$ = "R XZM'G
WL GSZG": GOTO 3
1890 IF I(V2) < > 0 THEN P$ = "R
WLM'G SZEZ GSZG": GOTO 3
1900 IF V2 = 1 AND A = 12 AND D5 =
0 THEN P$ = "GSV XZYRMVG OLXP
RH WVHGILBVW": GOSUB 4:D5 = 1:
I(1) = 100:I(5) = ABS (I(5)):
P4 = P4 - 1: GOTO 2210
1910 IF (V2 = 1 OR V2 = 15) AND (D
3 = 1 OR D7 = 1 OR E0 = 1 OR E
3 = 1 OR E7 = 1) THEN I(V2) =
100:P4 = P4 - 1: GOTO 1460
1920 IF (V2 = 1 OR V2 = 15) AND A =
1 THEN P$ = "GSV ZRIOLXP RH WV
HGILBVW...BLF ZIV YOLDMLFG RMG
L GSV EZXFFN LU HKZ XV!": GOSUB
4: GOTO 2460
1930 IF (V2 = 1 OR V2 = 15) AND A =
36 AND E6 = 0 AND E5 = 0 THEN

```

## APPLE SPEAKS INTELLIGENTLY!

The people who dared to teach Atari to talk are again challenging the microcomputer establishment with the VOICEBOX Speech Synthesizer for Apple. This low cost intelligent peripheral can speak thousands of words unassisted, generated directly from its firmware ROM dictionary located on its plug-in card. This means that speech, with variable intonation and speed, can be used in any of your apple programs without ever having to bother loading a disk. And, in case you want to expand your dictionary to include unusual words or words in foreign languages, you can easily define them with our 64 phonemes and store them by the thousands on one of the six special dictionaries provided for on our disk.

In addition your VOICEBOX for Apple can be easily coded to sing on key with uniform barlengths and you can store (record) your songs on disk, retrieving and modifying sections whenever you want. With the disk system, you'll also enjoy an educational random sentence generator and graphic speech animation! The VOICEBOX for Apple will run on 32K Apple II with Applesoft or Apple II Plus systems equipped with sixteen-sector disk drives. VOICEBOX for Apple comes with loudspeaker and disk. The Alien Group also makes a less expensive VOICEBOX for Apple with all features (including expandable disk dictionary), but excluding firmware ROM and singing capability. Speaker is optional on this unit.



For Atari users, the VOICEBOX for 16K and up Atari plugs directly into the serial port. No extra cables are needed and no speaker is needed since the speech comes directly over your TV monitor. This unit has all speech synthesis features except singing and firmware ROM.

AL-3001 VOICEBOX for Apple. With firmware ROM, singing capability and speaker \$215.00  
 AL-3501 VOICEBOX for Apple. Without firmware ROM, singing capability and speaker \$139.00  
 AL-4001 Speaker for AL-3501 (the AL-3501 will also work with any other speaker) \$15.00  
 AL-5001 VOICEBOX for Atari \$169.00

All mail orders are on a 10 day moneyback guarantee if you're not completely satisfied. When ordering enclose check or money-order or state VISA or MASTERCARD number. Send mail orders to:

THE ALIEN GROUP or Telephone orders  
 27 West 23rd Street from 10 AM to 6 PM  
 Dept. SS-2 New York time  
 New York, New York 10010 (212) 924-5546

ALSO AVAILABLE AT LEADING COMPUTER STORES THROUGHOUT THE WORLD.

```

P# = "GSV WLLI RH WVHGILBVW...
BLF ZIV HFXPVW RMGL GSV FMKIV
HHFIRAVW IVZYG LI YFROWRNT": GOSUB
4: GOTO 2460
1940 IF (V2 = 1 OR V2 = 15) AND A =
36 AND E6 = 0 AND F3 = 0 THEN
P# = "GSV WLLI RH WVHGILBVW. B
LF ZIV YLNYZIVWVDRGS IZWRZGRM
": GOSUB 4: GOTO 2460
1950 IF (V2 = 1 OR V2 = 15) AND A =
36 AND E6 = 0 THEN P# = "GSV W
LLI RH WVHGILBVW": GOSUB 4: E6 =
1: I(V2) = 100: P4 = P4 - 1: GOTO
2210
1960 IF V2 = 1 OR V2 = 15 THEN P# =
"GSV " + I$(V2) + " SZH ML VUU
VXG": GOSUB 4: I(V2) = 100: P4 =
P4 - 1: GOTO 2210
1970 GOTO 1340
1980 IF V2 = 0 THEN P# = "R XZM'G
WL GSZG": GOTO 3
1990 IF I(V2) < > 0 THEN P# = "R
WLM'G SZEZ GSZG": GOTO 3
2000 IF V2 = 5 AND D3 = 1 THEN V2#
= "GVI"
2010 IF V2 = 5 AND (D7 = 1 OR E0 =
1 OR E3 = 1 OR E7 = 1) THEN V2
# = "LRW"
2020 IF V2 = 5 THEN 1360
2030 IF V2 = 4 AND F3 = 0 AND (A =
40 OR A = 35 OR A = 30 OR A =
31) THEN P# = "GSV YOZXP WVERX
V RH YORMPRMT": GOTO 3
2040 IF V2 = 4 AND F3 = 0 AND A =
36 THEN P# = "GSV YOZXP WVERXV
RH UOZHSRMT YIRTS6OB": GOTO 3
2050 IF V2 = 4 THEN P# = "GSVIV ZI
VM'G ZMB ERHRYOV XLMGILOH LM
GSRH WVERXV": GOTO 3
2060 IF V2 = 12 THEN I(12) = A: P4 =
P4 - 1: I$(12) = "ZINWV KSLGLM
YLN": F2 = 35: P# = "GSV YLNY D
ROO VCKDLWV RM 35 NRMFGVH": GOTO
3
2070 IF V2 = 2 AND A = 12 AND D5 =
0 THEN D5 = 1: P# = "GSV XZYRMV
G RH MLD LKVM": GOSUB 4: I(5) =
ABS (I(5)): GOTO 2210
2080 IF V2 = 2 AND A = 12 AND D5 =
1 THEN P# = "GSV XZYRMVG RH ZO
IVZWB LKVM": GOTO 3
2090 IF V2 = 7 AND A = 26 AND D9 =
0 THEN D9 = 1: I(16) = ABS (I(
16)): P# = "GSV HZUV LKVMH": GOTO
3
2100 IF V2 < > 11 THEN P# = "DSZ6
WL BLF DZMG NV GL WL DRGS GSV
" + I$(V2) + "?": GOTO 3
2110 IF I(8) < > 0 THEN P# = "GSV

```

```

IV ZIVM'G ZMB YZGGVIRVH ULI GS
V IZWRM": GOTO 3
2120 IF F2 < > 0 THEN F2# = STR#
(F2): PRINT : P# = "Z ELRXV HZB
H 'YLN Y HGZGFH": GOSUB 4: P# =
F2# + " NRMFGVH FMGRO WVGLMZGR
LM": GOTO 3
2130 P# = "GSV IZWRM RH HROVMG": GOTO
3
2140 HOME : P# = " * * * KOZBVI'H HG
ZGFH * * *": GOSUB 4: PRINT
2150 P1# = STR# (P1): P# = "XFIIVMG
SRG KLRMGH = " + P1#: GOSUB
4
2160 P2# = STR# (P2): P# = "WVCGVIR
GB ZGGIRYFGV = " + P2#: GOSUB
4
2170 P3# = STR# (P3): P# = "OFXP Z6
GIRYFGV = " + P3#: GOSUB
4
2180 PRINT : PRINT "HIT <RETURN> T
O CONTINUE": GET T#: PRINT
2190 GOTO 2360
2200 HOME : P# = "TINV LEVI": GOSUB
4: GOTO 2470
Update player status, and conduct
combat if appropriate.
2210 IF F2 < > 0 THEN F2 = F2 - 1
: IF F2 < = 0 THEN 2380
2220 IF F4 < > 0 THEN F4 = F4 - 1
: IF F4 < = 0 THEN 2420
2230 IF P1 < P0 THEN P5 = P5 + .5:
IF P5 = 1 THEN P5 = 0: P1 = P1
+ 1
2240 IF D3 = 0 AND D7 = 0 AND E0 =
0 AND E3 = 0 AND E7 = 0 THEN 2
350
2250 T = INT ( RND (1) * 100)
2260 IF D3 = 1 THEN P# = "GSV NLMH
GVI Z6GZXP... ": GOSUB 4
2270 IF D3 < > 1 THEN P# = "GSV H
VXFIRGB ZMWILRW HSLLG... ": GOSUB
4
2280 IF T > 80 - (P2 + P3) THEN P#
= "RG NRHHVH": GOSUB 4: GOTO
2350
2290 P1 = P1 - ( INT ((5 * RND (1)
) + 1) + ((5 * RND (1)) + 1) +
((5 * RND (1)) + 1) + (5 * RND
(1)) + 1) + 15 - P3)
2300 IF D3 < > 1 THEN P1 = P1 + 5
2310 IF P1 < 0 THEN 2460
2340 P# = "BLF ZIV SRG!": GOSUB 4
2350 IF V1 = 0 OR V1 > 4 OR D3 + D
7 + E0 + E3 + E7 > 0 THEN FOR
ZZ = 1 TO 2000: NEXT
Initialize for new turn and jump
to appropriate room description.
2360 V# = "": V1# = "": V2# = "": V1 =

```

```

0: V2 = 0: A# = "": B# = "": C# =
"": D# = "": E# = "": N = 0: S = 0
: W = 0: E = 0
2370 ON A GOSUB 10,20,30,40,50,60,
70,80,90,100,110,120,130,140,1
50,160,170,180,190,200,210,220
,230,240,250,260,270,280,290,3
00,310,320,330,340,350,360,370
,380,390,400,410,420,430,440,4
50,2500: GOTO 460
Evaluate end-game conditions and
display appropriate messages.
2380 HOME : IF A = 46 THEN F4 = -
1: GOTO 2500
2390 P# = "GSV KSLGLM YLNY VCKDLWVH
...GSV VMGRIV XLNKDVC RH WVH
GILBVW": GOSUB 4
2400 P# = "BLF SZEZ YVVM PROOVW YB
GSV ULIXV LU GSVYOZHG": GOSUB
4
2410 PRINT : PRINT : GOTO 2470
2420 HOME : IF A = 46 THEN 2500
2430 IF A = 38 THEN P# = "GSV XLNK
FGVI UOZHSVH YIRTS6OB, VNRGGRM
T HKZIPH RM ZOO WRIVXGRM": GOSUB
4
2440 P# = "GSV XLNKDVC HFVVWMOB VCK
DLWVH RMGL NRO- ORLMH LU KRXXV
H": GOSUB 4
2450 P# = "BLF ZIV PROOVW YB GSV UZ
DORMT WVIYRH ZILFMW BLF": GOSUB
4: PRINT : PRINT : GOTO 2470
2460 P# = "BLF ZIV WVZW!": GOSUB 4
2470 INPUT "DO YOU WANT TO PLAY AG
AIN? (Y/N) ": A#
2480 IF LEFT# (A#,1) = "Y" THEN 2550
2490 PRINT : HOME : END
2500 P# = "GSV HKZIV HSRK HFVVWMOB
ORUGH RMGL LIYRGZILFMW GSV KOZ
MVG": GOSUB 4
2510 IF ((F2 = 0) OR (F2 < > 0 AND
I(12) < > 41)) AND F4 = 0 THEN
P# = "BLF WRWM'G WVHGILB GSV Y
ZHV. BLF SZEZ UZROVW BLFI NR
HRLM.": GOSUB 4: PRINT : GOTO
2470
2520 P# = "UULN Z WRHGZMXV, BLF XZM
HVV GSV ZORVM YZHV VCKDLWV":
GOSUB 4
2530 IF I(16) < > 0 THEN P# = "BL
F WRWM'G IVXLEVI GSV HVXIVG KO
ZMH MVVWVW YB HGZI XLNNZMW
": GOSUB 4: PRINT : GOTO 2470
2540 P# = "NRHRLM RH Z HFXVHH!": GOSUB
4: GOTO 2470
Initialize workspace. Read in
items, and verbs.
2550 CLEAR : DIM I$(16), I(16), V$(1
7): C155 = 155: C64 = 64: N1 = 1

```

```

2560 HOME : PRINT TAB(6)"OPERATI
ON: SABOTAGE BY RAY SATO": PRINT
TAB(6)"APPLE II+ VERSION BY
RON SHAKER"
2570 FOR T = 1 TO 16: READ I$(T),I
(T): NEXT
2580 FOR T = 1 TO 17: READ V$(T): NEXT
2590 FOR T = 1 TO 40:P0 = P0 + INT
(2 * RND (1)) + 1: NEXT T

Establish player attribute points.
Jump to first room.

2600 P1 = P0
2610 FOR T = 1 TO 10:P2 = P2 + INT
(2 * RND (1)) + 1: NEXT T
2620 FOR T = 1 TO 10:P3 = P3 + INT
(2 * RND (1)) + 1: NEXT T
2630 FOR T = 1 TO 50:D4 = D4 + INT
(2 * RND (1)) + 1:D8 = D8 + INT
(2 * RND (1)) + 1:E1 = E1 +
INT (2 * RND (1)) + 1:E4 =

```

```

E4 + INT (2 * RND (1)) + 1:E
8 = E8 + INT (2 * RND (1)) +
1: NEXT T
2640 A = 1:P4 = 1
2650 GOSUB 10: GOTO 460
Item and verb data
2660 DATA KOZHGRX VCKOLHREV,0,XI
LDYZI,7,XZOVNWZI,8,HNZOO YOZXP
WVERXV,9,OZHVI KRHGLO,-12,HVX
FIRGB PVB,-16,VOVXGILMRX XLMGI
LO YZGLM,-17,YZGGVIRVH,18,XLNM
FGVI WVHGIFXG KILTIZN,21,HRDEV
I KROO,23,KLIGZYOV IZWR,25,OZ
ITV KSLGLM VLN,28
2670 DATA TZOZGRX XSZIG,32,OZFMX
S HBHGVN XZHHVGGV,-32,MRGILTO
BXVIRM,39,HVXIVG KOZMH,-26
2680 DATA MLI6S,HLF6S,DVHG,VZHG,L
KVM,TVG,WILK,HSL6,RMHVIG,VZG,
KFHS,IVZW,RMEVMGLIB,GSILD,FHV,
HGZGFH,JFRG

```

**APPLE™ SWAT TABLE FOR:  
OPERATION: SABOTAGE** (Modified Parameters: NU=5, B=200)

LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
1 - 6	ZW	118	1370 - 1390	KE	235
7 - 30	UO	208	1400 - 1420	BS	202
40 - 80	XW	235	1430 - 1460	AC	291
90 - 120	CB	238	1470 - 1490	NG	297
130 - 160	VH	263	1500 - 1530	PW	214
170 - 200	TW	254	1540 - 1570	OZ	288
210 - 240	RR	226	1580 - 1620	HO	193
250 - 270	KL	218	1630 - 1650	WD	230
280 - 310	BR	258	1660 - 1680	FU	210
320 - 350	VG	225	1690 - 1720	BN	219
360 - 390	XW	238	1730 - 1770	VG	212
400 - 410	XD	201	1780 - 1810	BI	230
420 - 450	YU	231	1820 - 1860	HJ	162
460 - 500	CN	232	1870 - 1910	NJ	252
510 - 550	JJ	219	1920 - 1930	WN	233
560 - 600	MV	201	1940 - 1960	PH	266
610 - 650	AW	122	1970 - 2010	VC	151
660 - 700	XI	206	2020 - 2050	LD	225
710 - 730	QO	245	2060 - 2080	FY	236
740 - 780	YP	197	2090 - 2120	EZ	294
790 - 830	CH	185	2130 - 2170	QL	223
840 - 880	MW	167	2180 - 2220	LC	143
890 - 930	KZ	115	2230 - 2270	PM	185
940 - 980	KC	164	2280 - 2340	IZ	172
990 - 1030	JY	119	2350 - 2370	MU	313
1040 - 1080	QA	196	2380 - 2420	RS	186
1090 - 1110	RL	231	2430 - 2450	XN	224
1120 - 1140	JJ	215	2460 - 2500	MM	172
1150 - 1180	DL	230	2510 - 2530	JJ	272
1190 - 1220	SO	289	2540 - 2580	YX	217
1230 - 1260	GO	250	2590 - 2630	KK	216
1270 - 1310	SF	200	2640 - 2660	PA	254
1320 - 1360	BX	203	2670 - 2680	WP	187



# ATARI® OWNERS & PROSPECTIVE OWNERS

Now is the time to buy  
ATARI® products at low,  
low prices!

ATARI® 800 w/ 16K ..... \$699  
w/ 32K ..... \$784  
w/ 48K ..... \$869  
ATARI® 400 w/ 16K ..... \$350  
ATARI® 850 Interface ..... \$175  
ATARI® 830 Modem ..... \$159  
ATARI 810 Disk Drive ..... \$460  
ATARI 410 Recorder ..... \$79

Cartridges \$35 each, any three for \$99.95

Asteroids • Centipede • Missile Command • Pac-Man • Space Invaders • Super Breakout • Computer Chess

PILOT (COMPLETE) ..... \$99  
PILOT HOME PKG ..... \$63  
STAR RAIDERS CART ..... \$38  
CAVERNS OF MARS ..... \$31  
MUSIC COMPOSER CART ..... \$45  
ENTERTAINER PKG ..... \$83  
PROGRAMMER PKG ..... \$56  
COMMUNICATOR PKG ..... \$320

PKG A — ATARI® 800 w/48K, 850 Interface, 810 Disk Drive, Epson MX-80 Printer w/ GRAFTRAX & Cable. **\$1990.**  
PKG B — ATARI® 800, w/16K, 410 Program Recorder, Any 3, \$35 Cartridges & Pair of Joysticks. **\$877.**  
PKG C — ATARI® 400 w/16K, 410 Program Recorder, Programmer Pkg. **\$475**  
PKG D — ATARI® 400 w/16K, Entertainer Pkg, Any 2, \$35 Cartridges. **\$494**

Add 1% for shipping & handling  
Michigan residents add 4% sales tax.  
ATARI® & are trademarks of ATARI®, Inc  
**WE ALSO BUY AND SELL USED ATARI®  
EQUIPMENT. WRITE TO US.**

## Computer's Voice™ software for the ATARI® 400/800 computers

**Word Search Puzzle Maker** — Makes word search puzzles and answers on a printer—24K cassette, 24K disk **\$24.95.**

**Christmas Music** — Sets 1, 2, & 3. Sing along with your ATARI®. Words and music (8 carols each) for use with the ATARI® Music Composer cartridge. 16K cassette; 16K disk **\$24.95** each. (Specify set number).

**MATHFAKS** — Math drill program using color and sound. Optional accessories include a printer and VOTRAX Type-N-Talk—24K cassette, 24K disk **\$24.95.**

**Electronic Grade Book** — Great classroom aid for teachers. Uses number or letter grades, weighted scores, screen or printer output, and more. 32K cassette, 40K disk **64.95.**

**Light Pen** — for the ATARI® 400/800 by Symtec. Suggested list price \$149.95. **Mention this ad and pay \$134.95.**

**AXALON** — 128K RAMDISK for ATARI® 800s only **\$420.**

Add \$2 p/order for shipping  
Michigan residents add 4% sales tax

## Computer's Voice™

2370 Ella Dr.  
Flint, MI 48504  
(313) 238-5585

```

SS SS SS SS SS SS SS SS SS SS
SS                                     SS
SS      ATARI BASIC                   SS
SS 'OPERATION: SABOTAGE'              SS
SS      AUTHOR: RAY SATO               SS
SS      TRANSL: RICH BOUCHARD         SS
SS      COPYRIGHT (C) 1982            SS
SS                                     SS
SS SS SS SS SS SS SS SS SS SS SS

```

Jump to program initialization.

```
1 GOTO 2620
```

Decode and print output.

```
4 IF P$="" THEN RETURN
```

```
5 FOR P=N1 TO LEN(P$):? CHR$(ABS((155*(P$(P)>"@"))-ASC(P$(P)))):NEXT P:?:R
ETURN
```

Encode input

```
6 V$="":IF V0$="" THEN RETURN
```

```
7 FOR J=N1 TO LEN(V0$):V$(LEN(V$)+N1)=
CHR$(ABS((155*(V0$(J)>"@"))-ASC(V0$(J)
))):NEXT J:RETURN
```

Auxiliary jump points

```
8 GOSUB N4:GOTO 2520
```

```
9 GOSUB N4:GOTO L1
```

Descriptions of individual rooms.

```
10 A$="ZM ZRIOLXP. GSVIV RH Z YOFV
YFGGLM SVIV":S=N2:RETURN
```

```
20 A$="Z MZIILD XLIIRWLI":N=N1:S=N3:RE
TURN
```

```
30 A$="Z MZIILD XLIIRWLI":N=N2:S=N4:RE
TURN
```

```
40 A$="Z MZIILD XLIIRWLI":N=N3:S=N5:RE
TURN
```

```
50 A$="Z HNZOD ILLN":N=N4:S=6:RETURN
```

```
60 A$="Z WVXLMGZNRMZGRLM XSZNYVI":B$="
GSVIV RH Z YOFV YFGGLM SVIV":N=N5:S=7:
RETURN
```

```
70 A$="Z HNZOD HGLIZTV XSZNYVI":N=6:S=
N8:W=12:RETURN
```

```
80 A$="Z HNZOD XSZNYVI":N=7:S=N9:W=13:
RETURN
```

```
90 A$="Z HNZOD VOVXGILMRX          OZYL
IZGLIB":N=N8:S=10:W=14:RETURN
```

```
100 A$="Z YRLOLTRXZO OZYLIZGLIB. GSV
IV RH Z IVW YFGGLM LM GSV DZOD":N=N9:W
=15:RETURN
```

```
110 A$="Z HGLIZTV XSZNYVI":W=16:RETURN
```

```
120 A$="Z OZITV XSZNYVI. GSVIV RH Z X
ZYRMVG SVIV":S=13:W=17:E=7:RETURN
```

```
130 A$="Z HGIZMTV KFIKOV ILLN. GSV
IV RH Z YOFV YFGGLM SVIV":N=12:S=14:W=
18:E=N8:RETURN
```

```
140 A$="Z HNZOD LUURXV":N=13:S=15:E=N9
:RETURN
```

```
150 A$="Z HNZOD ILLN DRGS Z XZIW  GZY
OV RM GSV XVHGV":N=14:S=16:W=20:E=10:
RETURN
```

```
160 A$="Z OZITV LUURXV. GSVIV RH ZWVH
P SVIV":N=15:W=21:E=11:RETURN
```

```
170 A$="Z LUURXV DRGS Z OZITV WVHP":S=
18:E=12:RETURN
```

```
180 A$="Z HGLIZTV ILLN":N=17:S=19:E=13
:RETURN
```

```
190 A$="Z OZITV SZOD":N=18:S=20:RETURN
```

```
200 A$="ZM VMGVIGZRMNVMG ILLN. Z HXI
VVM IVHGH LM GSV DZOD":B$="GSVIV RH Z
YOFV ZHW Z IVW YFGGLM MVZI GSV HXIVVM"
```

```
202 N=19:S=21:E=15:RETURN
```

```
210 A$="Z WZGZ IVXLIW HGLIZTV ILLN":N=
20:E=16:RETURN
```

```
220 A$="IZWZI XLMGILO. GSVIV RH Z HNZ
OD HXIVVM SVIV":S=23:W=27:RETURN
```

```
230 A$="Z NVWRXZO HGZGRLM. GSVIV RH
Z OZITV GZYOV SVIV":N=22:S=24:W=28:RET
URN
```

```
240 A$="Z HVXFIRGB HGZGRLM":N=23:RETUR
N
```

```
250 A$="Z IZWRL ILLN":S=26:W=30:RETURN
```

```
260 A$="Z HNZOD ILLN. GSVIV RH Z HZU
V RM GSV HLF6S DZOD":N=25:W=31:RETURN
```

```
270 A$="GSV ILYLG XLMGILO XVHGV. GSV
IV RH Z HNZOD XLMGILO XLNKF6VI NLF
MGVW RM GSV DZOD":B$="":W=32:E=22
```

```
272 RETURN
```

```
280 A$="GSV DVZKLMH HGLIZTV ILLN":S=29
:W=33:E=23:RETURN
```

```
290 A$="Z ORYZIZB":N=28:S=30:W=34:RETU
RN
```

```
300 A$="Z HVXFIRGB XSVXP ZIVZ":N=29:W=
36:E=25:RETURN
```

```
310 A$="Z HNZOD ILLN DRGS Z WVHP. Z H
RTM IVZWH"
```

```
312 B$="KIVHFFIV GL IVZXGLI." Z YOFV
YFGGLM RH OZXZGVW FMVVI GSV HRTM":E=2
6:RETURN
```

```
320 A$="OZFMXS XLMGILO. GSVIV RH ZWVH
P SVIV":S=33:W=37:E=27:RETURN
```

```
330 A$="Z HNZOD XLIIRWLI":N=32:S=34:E=
28:RETURN
```

```
340 A$="Z HGIZMTV YOFV ILLN. GSVIVRH
Z IVW YFGGLM SVIV":N=33:S=35:E=29:RETU
RN
```

```
350 A$="Z GRMB HGLIZTV ILLN":N=34:W=40
:RETURN
```

```
360 A$="Z HNZOD XSZNYVI. Z WLLI DVH
G OVZWH GL GSV MFXOVZI IVZXGLI":E=30:R
ETURN
```

```
370 A$="Z HNZOD, MZIILD XLIIRWLI":N=42
:E=32:RETURN
```

```
380 A$="GSV XLNKF6VI XVHGV. GSVIVRH
```

```
Z HNZOD HOLG RM GSV XLNKF6VI":S=39:RET
URN
```

```
390 A$="GSV XSVNRXZO OZY." :N=38:S=40:R
ETURN
```

```
400 A$="GSV IVZXGLI XLMGILO XVHGV. GSV
IV RH Z YOFV YFGGLM ZHW Z IVW LMV. Z H
RTM HZBH IVZXGLI XLMGILO - "
```

```
402 A$(LEN(A$)+N1)="IVW=LM, YOFV=LUU":
N=39:E=35:RETURN
```

```
410 A$="GSV MFXOVZI IVZXGLI. Z XLN
KFGVI IVHGH LM GSV DZOD":E=36:RETURN
```

```
420 A$="GSV DVHG VHW LU Z OLMT XLI
IRWLI":S=37:E=43:RETURN
```

```
430 A$="GSV VZHG VHW LU Z OLMT XLI
IRWLI":W=42:E=44:RETURN
```

```
440 A$="Z HVXFIRGB XVHGV":W=43:E=45:R
ETURN
```

```
450 A$="Z HNZOD OZFMXS ZIVZ":B$="GSVIV
RH Z HNZOD HOLG MFCG GL GSV OZFMX
STZGV":W=44:RETURN
```

Extended descriptions of current location

460 REM

```
470 IF A=10 AND (D3=N1 OR D3=N2) THEN
C$="GSV NLMHGVH H XZTV RH LKVM"
```

```
480 IF A=12 AND D5=NO THEN C$="GSV XZY
RMVG RH OXLPVW"
```

```
490 IF A=12 AND D5=N1 THEN C$="GSV XZY
RMVG RH LKVM"
```

```
500 IF A=20 AND D6=NO THEN C$="GSV HXI
VVM RH YOZMP"
```

```
510 IF A=20 AND D6=N1 THEN C$="Z NLERV
RH YVRMT KOZBVW LM GSV HXIVVM"
```

```
520 IF A=26 AND D9=NO THEN C$="GSV HZU
V RH OXLPVW"
```

```
530 IF A=26 AND D9=N1 THEN C$="GSV HZU
V RH LKVM"
```

```
540 IF A=27 AND E2=NO THEN C$="GSV XLN
KFGVI RH ZXGREV"
```

```
550 IF A=27 AND E2=N1 THEN C$="GSV XLN
K6VI RH WVHGILBVM"
```

```
560 IF A=36 AND E6=NO THEN C$="GSV IVZ
XGLI WLLI RH URINOB OXLPVW"
```

```
570 IF A=36 AND E6=N1 THEN C$="GSV IVZ
XGLI WLLI RH LKVM":W=41
```

```
580 IF A=45 AND E9=NO THEN C$="GSV OZF
NXS TZGV RH XOLHVW"
```

```
590 IF A=45 AND E9=N1 THEN C$="GSV OZF
NXS TZGV RH LKVM":E=46
```

Generate list of visible items and available exits for current location.

```
600 A$(LEN(A$)+N1)="." :IF LEN(B$)>N3 T
HEN B$(LEN(B$)+N1)="."
```

```
610 IF LEN(C$)>N3 THEN C$(LEN(C$)+N1)=
"."
```

```
650 IF N<>NO THEN E$="MLIGS "
```

```
660 IF S<>NO THEN E$(LEN(E$)+N1)="HLFG
S "
```

```
670 IF W>NO THEN E$(LEN(E$)+N1)="DVHG
```

```

"
680 IF E>NO THEN E$(LEN(E$)+N1)="VZHG
"
690 IF E$<>" THEN E$=E$(N1,LEN(E$)-N1
)
Describe current location, visible
items, and available exits.
700 GRAPHICS NO:PRINT "YOU ARE IN ";P
$=A$:GOSUB N4:PRINT :IF B$<>" THEN P$
=B$:GOSUB N4
710 IF C$<>" THEN P$=C$:GOSUB N4
720 P$=" ":PRINT :PRINT "OBJECTS YOU C
AN SEE:":FOR T=N1 TO 16
722 IF A=I(T) THEN P$=I$(IP(T),IP(T+N1
)-N1):GOSUB N4
730 NEXT T:IF P$=" " THEN P$="MLGSRMT"
:GOSUB N4
732 ? :? "EXITS: ":P$=E$:GOSUB N4:?
Print out additional warnings,
messages, etc.
740 IF (A=40 OR A=35 OR A=30 OR A=31)
AND I(14)=NO AND F3=NO THEN P$="GSV HN
Z00 YOZXP WVERXV RH YORMPRT":GOSUB N4
750 IF A=36 AND I(N4)=NO AND F3=NO THE
N P$="GSV HNZ00 YOZXP WVERXV RH UOZHSR
MT YIRTS60B":GOSUB N4
760 IF F4=NO THEN 770
762 P$="GSV XLNKF6VI HZBH":GOSUB N4:P
$=STR$(F4):P$(LEN(P$)+N1)=" NRMF6VH FM
GR0 WVHGIFX6RLM":GOSUB N4
770 IF D3=N1 THEN PRINT :P$="* * * ZOR
VM NLMHGVI Z66ZXPRMT * * *":GOSUB N4
780 IF D7=N1 OR E0=N1 OR E3=N1 OR E7=N
1 THEN PRINT :P$="* * * HVXFIRGB KZGIL
D Z66ZXPRMT * * *":GOSUB N4
Get and interpret command.
790 PRINT :PRINT "COMMAND";:INPUT V0$:
GOSUB 6:PRINT
800 FOR T=N1 TO N4:IF V$=VB$(T*N3-N2,T
*N3-N2) THEN V$=VB$(T*N3-N2,T*N3)
810 NEXT T
820 IF LEN(V$)<N3 THEN 700
830 V1$=V$(N1,N3):V2$=V$(LEN(V$)-N2)
840 FOR T=N1 TO 17:IF V1$=VB$(T*N3-N2,
T*N3) THEN V1=T
850 NEXT T:IF V1=NO THEN P$="R WLM'G F
MNVIHGZMW DSZG BLF DZMG.":GOSUB N4:GOT
O L1
860 FOR T=N1 TO 16:IF V2$=I$(IP(T+N1)-
N3,IP(T+N1)-N1) THEN V2=T
870 NEXT T
880 ON V1 GOTO 900,940,980,1050,1130,1
310,1370,1390,1560,1620,1660,1790,1860
,1910,2010,2170,2250
890 GOTO L1
Command handler routines.
900 IF N=NO THEN GOTO L2
910 IF D3=N1 THEN P$="GSV NLMHGVI YOLX
PH GSV VCRG":GOSUB N4:GOTO L2

```

```

920 IF D7=N1 OR E3=N1 THEN GOTO L2
930 A=N:GOTO L1
940 IF S=NO THEN GOTO L2
950 IF S=24 AND D8<>NO AND E2<>N1 THEN
D7=N1:GOSUB L4
960 IF S=30 AND E4<>NO AND E2<>N1 THEN
E3=N1:GOSUB L4
970 A=S:GOTO L1
980 IF W=NO THEN GOTO L2
990 IF D3=N1 THEN P$="GSV NLMHGVI YOLX
PH GSV VCRG":GOSUB N4:GOTO L1
1000 IF E0=N1 OR E3=N1 OR E7=N1 THEN G
OTO L3
1010 IF W=41 AND F3=NO THEN P$="IZWRZG
RLM UILN GSV IVZXGLI SR6H BLF":GOSUB N
4:GOTO 2520
1020 IF W=30 AND E4<>NO AND E2<>N1 THE
N E3=N1:GOSUB L4
1030 IF W=27 AND E8<>NO AND E2<>N1 THE
N E0=N1:GOSUB L4
1040 A=W:GOTO L1
1050 IF E=NO THEN GOTO L2
1060 IF E0=N1 OR E3=N1 OR E7=N1 THEN G
OTO L3
1070 IF E=27 AND E1<>NO AND E2<>N1 THE
N E0=N1:GOSUB L4
1080 IF E=44 AND E8<>NO AND E2<>N1 THE
N E7=N1:GOSUB L4
1090 A=E:GOTO L1
1100 P$="GSV HVXFIRGB TFZIW YOLXPH GSV
VCRG":GOSUB N4:GOTO L1
1110 P$="GSVIV RH ML DZB GL TL GSZG WR
IVX6RLM":GOTO N9
1120 P$="Z HVXFIRGB ZMWILRW ZDZRGH BLF
":GOSUB N4:RETURN
1130 IF A<>N1 OR V2$<>"LXP" THEN 1140
1132 P$="GSV ZRIOLXP LKVMH ZMW BLF ZIV
YOLDM LFG RMGL GSV EZXFFN LU HKZXV"
:GOSUB N4:GOTO 2520
1140 IF A<>12 OR V2$<>"MVG" OR D5<>NO
OR I(N2)=NO THEN 1150
1142 P$="BLF QFHG ZIVM'G HGILMT VMLFTS
GL ULIXVRG LKVM":GOSUB N4:GOTO L1
1150 IF A<>12 OR V2$<>"MVG" OR D5<>NO
OR I(N2)<>NO THEN 1160
1152 P$="GSV XILDYLD SVOKVM. GSV XZYR
MVG RH MLD LKVM":GOSUB N4:D5=N1:I(N5
)=ABS(I(N5)):GOTO L1
1160 IF A=12 AND V2$="MVG" AND D5=N1 T
HEN P$="GSV XZYRMVG RH ZOIVZWB LKVM":G
OSUB N4:GOTO L1
1170 IF A=16 AND V2$="VHP" THEN ? OK$:
I(6)=ABS(I(6)):GOTO L1
1180 IF A=17 AND V2$="VHP" THEN PRINT
OK$:I(7)=ABS(I(7)):GOTO L1
1190 IF A=26 AND V2$="ZUV" AND D9=N1 T
HEN P$="GSV HZUV RH ZOIVZWB LKVM":GOTO
N4:GOTO L1
1200 IF A=26 AND V2$="ZUV" AND D9=NO T

```

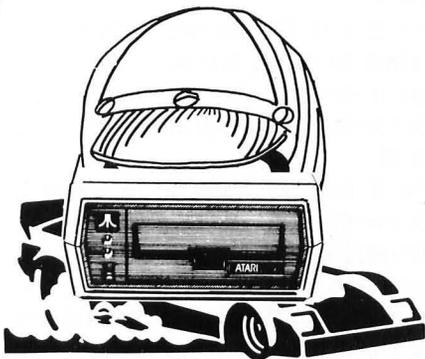
```

HEN P$="R WLM'G SZEZ GSV PVB GL LKVM G
SV HZUV":GOTO N9
1210 IF A=31 AND V2$="VHP" THEN P$="LP
. BLF URMW MLGSRMT RMRHW":GOTO N9
1220 IF A=32 AND V2$="VHP" THEN PRINT
OK$:I(14)=ABS(I(14)):GOTO L1
1230 IF A=36 AND V2$="LLI" AND E6=N1 T
HEN P$="GSV WLLI RH ZOIVZWB LKVM":GOTO
N9
1240 IF A<>36 OR V2$<>"LLI" OR E6<>NO
THEN 1260
1242 IF I(6)<>NO THEN P$="BLF WLM'G SZ
EV GSV PVB GL GSV WLLI":GOTO N9
1250 IF E5=NO THEN P$="GSV IVZXGLI YFR
OWRMT RH MLG KIVHHFIRAVW. B
LF ZIV YOLDM RMGL GSV EZXFFN":GOTO N8
1260 IF A=36 AND V2$="LLI" AND I(6)=NO
THEN P$="GSV WLLI RH MLD LKVM":E6=N1:
GOTO N9
1270 IF A=41 AND V2$="MVO" THEN P$="GS
V KZMVD RH URINOB OLPVW":GOTO N9
1280 IF A=45 AND V2$="LXP" AND E9=N1 T
HEN P$="GSV ZRIOLXP RH ZOIVZWB LKVM":G
OTO N9
1290 IF A=45 AND V2$="LXP" AND E9=NO T
HEN P$="GSVIV ZIVM'G ZMB ERHRYOV XLMG1
LOH":GOTO N9
1300 P$="R XZM'G WL GSZG":GOTO N9
1310 IF V2$="GVI" OR V2$="LRW" THEN P$
="WLM'G YV IRWRXFOLFH":GOTO N9
1320 IF V2=NO THEN P$="R XZM'G WL GSZG
":GOTO N9
1330 IF I(V2)=NO THEN P$="BLF ZOIVZWB
SZEZ GSZG":GOTO N9
1340 IF A<>I(V2) THEN P$="R WLM'G HVV
RG SVIV":GOTO N9
1350 IF P4>N8 THEN P$="HLIIB, BLF XZM
'G XZIIIB ZMBGSRMT NLIV":GOSUB N4:GOTO
L1
1360 P4=P4+N1:I(V2)=NO:PRINT OK$:GOTO
L1
1370 IF V2=NO THEN P$="BLF WLM'G SZEZ
GSZG":GOTO N9
1380 P4=P4-N1:I(V2)=A:PRINT OK$:GOTO L
1
1390 IF I(N5)<>NO THEN P$="BLF WLM'G S
ZEZ Z DVZKLM":GOTO N9
1400 IF A=N1 AND V2$="LXP" THEN P$="BL
F ZIV YOLDM LFG LU GSV ZRIOLXP RMGL GS
V EZXFFN LU HKZXV":GOTO N8
1410 IF A=27 AND V2$="GVI" THEN P$="GS
V XLNKF6VI RH WVHGILBVW":E2=N1:E0=NO:G
OTO N9
1420 IF A=38 AND V2$="GVI" THEN P$="GS
V HSL6 IVUOVXGH LUU LU GSV XLNKF6VI":G
OTO N8
1430 IF A=41 AND V2$="GVI" THEN P$="GS
V DSLOV MFNXOVZI IVZXGLI RH VCKOLWRMT"
:GOTO N8

```

# TURN YOUR ATARI 810 DISK DRIVE INTO A REAL SPEED DEMON.

WITH



## FAST-CHIP

- Increases the formatting speed of 810 disks by 10 to 40%.
- Plug compatible
- Easy to install

### ORDERING INFORMATION

Available at your local Atari dealer or Atari center for only \$39.95 (installation may be extra).

If not available in your area call BINARY directly to place your order. Our order lines are open 24 hours per day, 7 days per week.

#### Shipping and handling charges:

North America: Add \$2.50  
Outside N.A.: Add 10%  
Michigan Residents: Add 4% tax.  
C.O.D.: Add \$2.00

#### Payment Methods:

VISA, Master Charge, AMEX, cash, certified check, personal check (allow for clearance), money order.

Look for Binary Software Products at your local computer store.

Dealer Inquiries invited

## BINARY™

COMPUTER SOFTWARE  
3237 Woodward Ave.  
Berkley, MI 48072  
(313) 548-0533

BINARY CORPORATION

```

1440 IF V2$="RWH" OR V2$="YLG" OR V2$="
"ILO" OR V2$="IWH" OR V2$="ZIW" THEN V
2$="LRW"
1450 IF V2$("<"GVI" AND V2$("<"LRW" THEN
P$="GSV DZHVI HSLG SZH ML VUUVXG":GOT
0 N9
1460 IF V2$="GVI" AND D3=NO THEN P$="R
WLM'G HVV ZMB NLMHGI SVIV":GOTO N9
1470 IF V2$="LRW" AND D7=NO AND E0=NO
AND E3=NO AND E7=NO THEN P$="R WLM'G H
VV ZMB ZMWILRW SVIV":GOTO N9
1480 T=INT(100*RND(NO))+1:IF T>P2+P3+5
0 THEN P$="BLF URIV ZMM NRHH":GOTO N9
1490 IF D3(">N1 THEN 1500
1492 P$="BLF SRG GSV NLMHGI":GOSUB N4
:D4=D4-((10+P2+P3)/N2):IF D4">NO THEN 1
540
1494 D3=NO:D4=NO:P$="BLF SZEZ PROOVW R
G":GOTO N9
1500 IF D7(">N1 THEN 1510
1502 P$="BLF SRG GSV ZMWILRW":GOSUB N4
:D8=D8-(5+P2+P2)/N2:IF D8("<=NO THEN D7=
NO:D8=NO:P$="RG RH WVHGILBVW":GOTO N9
1504 GOTO 1540
1510 IF E0(">N1 THEN 1520
1512 P$="BLF SRG GSV ZMWILRW":GOSUB N4
:E1=E1-(N5+P2+P3)/N2:IF E1("<=NO THEN E0
=NO:E1=NO:P$="RG RH WVHGILBVW":GOTO N9
1514 GOTO 1540
1520 IF E3(">N1 THEN 1530
1522 P$="BLF SRG GSV ZMWILRW":GOSUB N4
:E4=E4-(N5+P2+P3)/N2:IF E4("<=NO THEN E3
=NO:E4=NO:P$="RG RH WVHGILBVW":GOTO N9
1524 GOTO 1540
1530 IF E7(">N1 THEN 1540
1532 P$="BLF SRG GSV ZMWILRW":GOSUB N4
:EB=EB-(N5+P2+P3)/N2:IF EB("<=NO THEN E7
=NO:EB=NO:P$="RG RH WVHGILBVW":GOTO N9
1534 GOTO 1540
1540 IF D3=N1 THEN P$="RG RH HGROO ZOR
EV":GOTO N9
1550 P$="GSV ZMWILRW RH HGROO UFMXGRM
RMT":GOTO N9
1560 IF V2=NO THEN P$="R XZM'G WL GSZG
":GOTO N9
1570 IF I(V2)("<NO THEN P$="R WLM'G SZE
V GSZG":GOTO N9
1580 IF V2(">N9 AND V2(">14 THEN P$="R X
ZM'G WL GSZG":GOTO N9
1590 IF (V2=N9 AND A=44) OR (V2=14 AND
A=38) THEN P$="MLGSRMT SZKKVMH":GOSUB
N4
1600 IF V2(">N9 OR A(">38 THEN 1610
1602 F4=35:P$="GSV XLNKFVI IVKORVH: '
YZHV WVHGIFXG HVJFVXV HGZIGVW":GOSU
B N4:P$="WVHGIFXGRM RM: "
1604 P$(LEN(P$)+N1)=STR$(F4):P$(LEN(P$
)+N1)=" NRMFGVH":P4=P4-N1:I(9)=100:GOT
0 N9

```

```

1610 IF V2=14 AND A=45 THEN P$="GSV TZ
GV LKVMH":E9=N1:GOTO N9
1620 IF V2(">10 THEN P$="WLM'G YV IRWRX
FOLFH":GOTO N9
1630 IF I(10)("<NO THEN P$="BLF WLM'G S
ZEZ GSZG":GOTO N9
1640 PRINT OK$:I(10)=50:P4=P4-N1:P1=P1
+N5+P3:IF P0("<P1 THEN P0=P1
1650 GOTO L1
1660 IF A=N1 AND V2$="OFV" THEN P$="GS
V ZRIOLXP LKVMH... BLF ZIV YOLDM LFGRM
GL GSV EZXFFN LU HKZXV":GOTO N8
1670 IF A=6 AND V2$="OFV" THEN P$="Z H
GIZMTV, LIZMTV TOLD XLEVIH BLF ZMM GSV
M UZVWH ZDZB":GOTO N9
1680 IF A=10 AND V2$="IVU" AND D3=N1 T
HEN P$="MLGSRMT SZKKVMH":GOTO N9
1690 IF A=10 AND V2$="IVW" THEN D3=N1:
P$="ZM ZORVM NLMHGI RH IVOZHVW. RG R
H Z6GZXPRMT BLF":GOTO N9
1700 IF A=13 AND V2$="OFV" THEN A=34:P
$="Z UOZHS LU ORTSG GVNKLIZIROB YORMWH
BLF":GOTO N9
1710 IF A=20 AND V2$="IVW" AND D6=NO T
HEN P$="MLGSRMT SZKKVMH":GOTO N9
1720 IF A=20 AND V2$="IVW" THEN D6=NO:
P$="GSV HXIVVM TLVH YOZMP":GOTO N9
1730 IF A=20 AND V2$="OFV" THEN D6=N1:
P$="GSV HXIVVM ORTSGH FK":GOTO N9
1740 IF A=31 AND V2$="OFV" THEN E5=N1:
PRINT OK$:GOTO L1
1750 IF A=34 AND V2$="IVW" THEN A=13:P
$="Z UOZHS LU ORTSG GVNKLIZIROB YORMWH
BLF":GOTO N9
1760 IF A=40 AND V2$="IVW" THEN F3=NO:
PRINT OK$:GOTO L1
1770 IF A=40 AND V2$="OFV" THEN F3=N1:
PRINT OK$:GOTO L1
1780 P$="MLGSRMT SZKKVMH":GOTO N9
1790 IF A=22 AND V2$="VVM" THEN P$="BL
F XZM HVV MLGSRMT LU RMGVIHVH LM GSVIZ
WZI":GOTO N9
1800 IF V2=NO THEN P$="R WLM'G SZEZ GS
ZG":GOTO N9
1810 IF I(V2)("<NO AND A(">I(V2) THEN P$
="R WLM'G SZEZ GSZG":GOTO N9
1820 IF V2=N3 OR V2=13 THEN P$="R HVV
MLGSRMT HKVXRZO":GOTO N9
1830 IF V2=N9 THEN P$="HLIIB, LMOB Z X
LNKFGVI XZM IVZW Z KILTIZN":GOTO N
9
1840 IF V2=16 THEN P$="GSV KOZMH ZIV H
VZOVW...LMOB XLNNZMH XZM LKVM GSVN":
GOTO N9
1850 P$="R XZM'G IVZW GSZG":GOTO N9
1860 GRAPHICS NO:P$="* * * KOZBVI'H RM
EVHGLIB * * *":GOSUB N4:PRINT
1870 FOR T=N1 TO 16:IF I(T)=NO THEN P$
=I$(IP(T),IP(T+N1)-N1):GOSUB N4

```

```

1880 NEXT T
1890 CLOSE #N1:OPEN #N1,N4,NO,"K":GET
#N1,T:CLOSE #N1
1900 GOTO 2410
1910 IF V2=NO THEN P$="R XZM'G WL GSZG
":GOTO N9
1920 IF I(V2)<>NO THEN P$="R WLM'G SZE
V GSZG":GOTO N9
1930 IF V2<>N1 OR A<>12 OR D5<>NO THEN
1940
1932 P$="GSV XZYRMVG RH WVGILBVM":D5=
N1:I(N1)=100:I(N5)=ABS(I(N5)):P4=P4-N1
:GOTO N9
1940 IF (V2=N1 OR V2=15) AND (D3=N1 OR
D7=N1 OR E0=N1 OR E3=N1 OR E7=N1) THE
N I(V2)=100:P4=P4-100:GOTO 1490
1950 IF (V2<>N1 AND V2<>15) OR A<>N1 T
HEN 1960
1952 P$="GSV ZRIOLXP RH WVGILBVM...BL
F ZIV YOLDM LFG RMGL GSV EZXFFN LU
HKZV":GOTO N8
1960 IF (V2<>N1 AND V2<>15) OR A<>36 O
R E6<>NO OR E5<>NO THEN 1970
1962 P$="GSV WLLI RH WVGILBVM...BLF Z
IV YOLDM RMGL GSV FMKIVHFRIVM IVZXGL
I YFROMRT":GOTO N8
1970 IF (V2<>N1 AND V2<>15) OR A<>36 O
R E6<>NO OR F3<>NO THEN 1980
1972 P$="GSV WLLI RH WVGILBVM. BLF Z
IV YLNYZIVWV DRGS IZWRZGRM":GOT
O N8
1980 IF (V2=N1 OR V2=15) AND A=36 AND
E6=NO THEN P$="GSV WLLI RH WVGILBVM":
E6=N1:I(V2)=100:P4=P4-N1:GOTO N9
1990 IF V2<>N1 AND V2<>15 THEN 2000
1992 P$="GSV ":P$(5)=I$(IP(V2),IP(V2+N
1)-N1):GOSUB N4:P$="SZH ML ZUUVX6":I(V
2)=100:P4=P4-N1:GOTO N9
2000 GOTO 1370
2010 IF V2=NO THEN P$="R XZM'G WL GSZG
":GOTO N9
2020 IF I(V2)<>NO THEN P$="R WLM'G SZE
V GSZG":GOTO N9
2030 IF V2=N5 AND D3=N1 THEN V2$="GVI"
2040 IF V2=N5 AND (D7=N1 OR E0=N1 OR E
3=N1 OR E7=N1) THEN V2$="LRW"
2050 IF V2=N5 THEN 1390
2060 IF V2=N4 AND F3=NO AND (A=40 OR A
=35 OR A=30 OR A=31) THEN P$="GSV YOZX
P WVERXV RH YORMPRT":GOTO N9
2070 IF V2=N4 AND F3=NO AND A=36 THEN
P$="GSV YDZXP WVERXV RH UOZHSRMT YIRTS
GOB":GOTO N9
2080 IF V2=N4 THEN P$="GSVIV ZIVM'G ZM
B ERHRYOV XLMGLOH LM GSRH WVERXV":GO
TO N9
2090 IF V2<>12 THEN 2100
2092 I(12)=A:P4=P4-N1:I$(IP(12),IP(12)
+N4)="ZINVM":F2=35:P$="GSV YLNY DROO V

```

```

CKOLWV RM 35 NRMFGVH":GOTO N9
2100 IF V2=N2 AND A=12 AND D5=NO THEN
D5=N1:P$="GSV XZYRMVG RH MLD LKVM":I(N
5)=ABS(I(N5)):GOTO N9
2110 IF V2=N2 AND A=12 AND D5=N1 THEN
P$="GSV XZYRMVG RH ZOIVZWB LKVM":GOTO
N9
2120 IF V2=7 AND A=26 AND D9=NO THEN D
9=N1:I(16)=ABS(I(16)):P$="GSV HZUV LKV
MH":GOTO N9
2130 IF V2=11 THEN 2140
2132 P$="DSZG WL BLF DZMG NV GL WL DRG
S GSV":GOSUB N4:P$=I$(IP(V2),IP(V2+N1)
-N1):P$(LEN(P$)+N1)="?":GOTO N9
2140 IF I(N8)<>NO THEN P$="GSVIV ZIVM'
G ZMB YZGGVIRVH ULI GSV IZWRM":GOTO N9
2150 IF F2=NO THEN 2160
2152 P$="Z ELRXV HZBH 'YLYN HGZGFH: ":
P$(LEN(P$)+1)=STR$(F2):P$(LEN(P$)+1)="
NRMFGVH":GOSUB N4
2154 P$="FMGRD WVGMLZGRM":GOTO N9
2160 P$="GSV IZWRM RH HROVMG":GOTO N9
2170 GRAPHICS NO:P$=" * * * KOZBVI'H HG
ZGFH * * *":GOSUB N4:PRINT
2180 P$="XFIIVMG SRG KLRMGH = ":P$(LEN
(P$)+N1)=STR$(P1):GOSUB N4
2190 P$="WVCGVIRGB ZG6IRYFGV = ":P$(LEN
(P$)+N1)=STR$(P2):GOSUB N4
2200 P$="OFXP ZG6IRYFGV = ":P$(LEN
(P$)+N1)=STR$(P3):GOSUB N4
2210 REM
2220 CLOSE #N1:OPEN #N1,N4,NO,"K":GET
#N1,T:CLOSE #N1
2230 REM
2240 GOTO 2410
2250 GRAPHICS NO:P$="GAME OVER":GOSUB
N4:GOTO 2530

```

Update player status, and conduct combat if appropriate.

```

2260 IF F2<>NO THEN F2=F2-N1:IF F2<=NO
THEN 2440
2270 IF F4<>NO THEN F4=F4-N1:IF F4<=NO
THEN 2480

```

```

2280 IF P1<P0 THEN P5=P5+0.5:IF P5=N1
THEN P5=NO:P1=P1+N1
2290 IF D3=NO AND D7=NO AND E0=NO AND
E3=NO AND E7=NO THEN 2400
2300 T=INT(RND(NO)*100+N1):PRINT
2310 IF D3=N1 THEN P$="GSV NLMHGV I ZGG
ZXP":GOSUB N4
2320 IF D3<>N1 THEN P$="GSV HVXFIRGB Z
MWILRW HSLLGH...":GOSUB N4
2330 IF T>80-(P2+P3) THEN P$="RG NRHHV
W":GOSUB N4:GOTO 2400
2340 P1=P1-INT(RND(NO)*N5)-INT(RND(NO)
*N5)-INT(RND(NO)*N5)-INT(RND(NO)*N5)-2
4+P3
2342 IF D3<>N1 THEN P1=P1+N5
2344 IF P1<NO THEN 2520
2390 P$="BLF ZIV SRG":GOSUB N4
2400 IF V1=0 OR V1>4 OR D3+D7+E0+E3+E7
>0 THEN FOR ZZ=N1 TO 150:NEXT ZZ

```

Initialize for new turn and jump to appropriate room description.

```

2410 V0$="":V$="":V1$="":V2$="":V1=NO:
V2=NO:A$="":B$="":C$="":D$="":E$="":N=
NO:S=NO:W=NO:E=NO
2420 IF A>45 THEN 2560
2422 GOSUB A#10:GOTO 470

```

Evaluate end-game conditions and display appropriate messages.

```

2440 GRAPHICS NO:IF A=46 THEN F2=-1:GO
TO 2560
2450 P$="GSV KSLGLM YLNY VCKOLWVH...GS
V VMGRIV XLNKOVC RH WVGILBVM":GOSUB N
4
2460 P$="BLF SZEY YVVM PROOVW YB GSV U
LIXV LU GSV YDZHG":GOSUB N4
2470 PRINT :PRINT :GOTO 2530
2480 GRAPHICS NO:IF A=46 THEN F4=-1:GO
TO 2560
2490 IF A=38 THEN P$="GSV XLNKFVI UOZ

```

## MOVING?

If you're planning to move, please let us know at least six weeks in advance. This will help us to change your address insuring you with prompt and accurate service on your subscription. Attach your current mailing label filling in your name and NEW address in the space provided.

Name \_\_\_\_\_  
 New Address \_\_\_\_\_  
 City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Send old label with your name and NEW address to:  
**SoftSide**  
 100 Pine Street  
 Holmes, PA 19043

```

HSVH YIRTS60B,      VNR66RMT HKZIPH
RM ZOO WRIVXGRLMH":GOSUB N4
2500 P$="GSV XLNKDVC HFVVVMOB VCKOLWVH
RMGL  NROORLMH LU KRVXVH":GOSUB N4
2510 P$="BLF ZIV PROOVV YB GSV UZDORMT
WVYIRH ZILFMW BLF":GOSUB N4:PRINT :P
RINT :GOTO 2530
2520 PRINT :P$="BLF ZIV WVZW!":GOSUB N
4
2530 PRINT :PRINT "DO YOU WANT TO PLAY
AGAIN";:INPUT A$
2540 IF A$(N1,N1)="B" THEN RUN
2550 GRAPHICS NO:END
2560 P$="GSV HKZVXV HSRK HFVVVMOB DRUGH
RMGL  LIYRG ZILFMW GSV KZNVG":GOSU
B N4
2570 IF (F2=NO OR (F2<>NO AND I(12)<>4
1)) AND F4=NO THEN 2574
2572 GOTO 2580
2574 P$="BLF WRWM'G WVHGILB GSV YZHV.
BLF SZEUVUZROVM BLFI NRHHRLM.":GOSUB N
4:PRINT :GOTO 2530
2580 P$="UJLN Z WRHGZMXV, BLF XZM HVV
GSV ZORVMYZHV VCKOLWV":GOSUB N4
2590 IF I(16)<>NO THEN P$="BLF WRWM'G
IVXLEVI GSV HVXIVG KZMH  MVVVWV YB H
GZI XLNNZHW":GOSUB N4:PRINT :GOTO 2530
2600 P$="NRHHRLM RH Z HFXXVHH!":GOSUB
N4
2610 END

```

Initialize workspace. Read in items, and verbs.

```

2620 CLR :DIM I$(300),I(16),IP(17),VB$(
51)
2622 DIM V$(20),V1$(3),V2$(3),A$(120),
B$(70),C$(40),D$(40),E$(40),P$(120)
2624 DIM Z$(40),OK$(3),V0$(20)
2626 NO=0:N1=1:N2=2:N3=3:N4=4:N5=5:N8=
8:N9=9:L1=2260:L2=1110:L3=1100:L4=1120
:OK$="OK":OK$(N3)=CHR$(253)
2630 GRAPHICS NO
2640 PRINT " OPERATION: SABOTAGE BY
RAY SATO":? :? " ATARI VERSION BY
RICH BOUCHARD"
2650 I$="":FOR T=N1 TO 16:READ Z$:IP(T
)=LEN(I$)+1:I$(LEN(I$)+N1)=Z$:READ Z:I
(T)=Z
2660 NEXT T:IP(T)=LEN(I$)+1
2670 READ VB$

```

Establish player attribute points.  
Jump to first room.

```

2700 P0=40:FOR T=N1 TO P0:P0=P0+INT(RN
D(NO)*N2):NEXT T
2710 P1=P0
2720 P2=10:FOR T=N1 TO P2:P2=P2+INT(RN

```

```

D(NO)*N2):NEXT T
2730 P3=10:FOR T=N1 TO P3:P3=P3+INT(RN
D(NO)*N2):NEXT T
2740 D4=50:D8=D4:E1=D4:E4=D4:E8=D4:FOR
T=N1 TO D4:D4=D4+INT(RND(NO)*N2):D8=D
8+INT(RND(NO)*N2)
2742 E1=E1+INT(RND(NO)*N2):E4=E4+INT(R
ND(NO)*N2):E8=E8+INT(RND(NO)*N2):NEXT
T
2750 A=N1:P4=N1
2760 GOTO 2410

```

Item and verb data

```

2770 DATA KOZHGRX VCKOLHREV,0,XILDYZI,
7,XZOVWMI,8,HNZOO YOZXP WVERXV,9,OZHV
I KRHGLD,-12,HVXFIRGB PVB,-16
2772 DATA VOVXGILMRX XLMGILO YZGLM,-17
,YZGGVIRVH,18,XLNKFBVI WVHGIFXG KILTIZ
N,21,HROEVI KROO,23
2774 DATA KLIGZYOV IZWR,25,OZITV KSLG
LM YLNY,28
2780 DATA TZOXGRX XSZIG,32,OZFMXS HBH
GVN XZHHV66V,-32,MRGILTOBXVIRM,39,HVXI
VG KOZMH,-26
2790 DATA MLIHLFDVHVZHLKVTVGMILHSLRMHV
ZGKFHIVZRMESIFVHGZJFR

```

ATARI® SWAT TABLE FOR:  
OPERATION: SABOTAGE (Modified Parameters: NU=5, B=200)

LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
1 - 7	WH	176	1420 - 1450	BW	259
8 - 30	NO	174	1460 - 1490	UL	217
40 - 70	VW	241	1492 - 1502	TY	231
80 - 100	NK	229	1504 - 1520	SZ	153
110 - 130	YN	241	1522 - 1532	AI	216
140 - 160	RM	249	1534 - 1570	VN	171
170 - 200	SC	291	1580 - 1602	CY	245
202 - 230	UE	283	1604 - 1630	BK	225
240 - 270	RE	303	1640 - 1670	BN	242
272 - 300	FI	207	1680 - 1700	NI	231
310 - 320	NY	222	1710 - 1740	NN	211
330 - 350	QX	217	1750 - 1790	XF	267
360 - 380	NS	229	1800 - 1840	DK	277
390 - 402	TY	220	1850 - 1890	PB	174
410 - 430	WJ	222	1900 - 1932	RZ	201
440 - 470	KR	217	1940 - 1952	ZM	201
480 - 520	ZB	244	1960 - 1972	QA	268
530 - 570	BO	261	1980 - 1992	KU	218
580 - 650	ND	206	2000 - 2040	LN	148
660 - 700	BQ	168	2050 - 2080	HA	251
710 - 732	SX	174	2090 - 2100	QS	222
740 - 760	QE	209	2110 - 2132	CL	260
762 - 780	DU	216	2140 - 2154	AA	204
790 - 830	VN	138	2160 - 2200	RL	237
840 - 880	NL	299	2210 - 2250	QD	96
890 - 930	FV	101	2260 - 2300	AW	166
940 - 980	MX	105	2310 - 2342	BM	216
990 - 1030	RV	213	2344 - 2410	FZ	205
1040 - 1080	PH	113	2420 - 2460	ZQ	223
1090 - 1130	TJ	186	2470 - 2500	LS	228
1132 - 1150	BT	237	2510 - 2550	OJ	178
1152 - 1170	TV	212	2560 - 2574	XK	221
1180 - 1210	EO	261	2580 - 2600	JX	201
1220 - 1242	SW	218	2610 - 2626	SK	374
1250 - 1270	SC	235	2630 - 2670	BG	201
1280 - 1310	DB	216	2700 - 2740	VB	217
1320 - 1360	NH	209	2742 - 2772	RE	275
1370 - 1410	UB	269	2774 - 2790	EL	185

# TRS-80<sup>®</sup>

```

SS SS SS SS SS SS SS SS SS SS
SS
SS TRS-80 BASIC SS
SS 'OPERATION: SABOTAGE' SS
SS AUTHOR: RAY SATO SS
SS COPYRIGHT (C) 1982 SS
SS SS
SS SS SS SS SS SS SS SS SS SS

```

Jump to program initialization.

```
2 GOTO2610
```

Decode and print output.

```
4 IFF$="THENRETURNSEFORP=1TOLEN(P$);II=ASC(MID$(P$,P,N1));PRINTCHR$(ABS((C155*(II>C64))-II));NEXT:PRINT:RETURN
```

Encode input

```
6 V$="";IFV0$="THENRETURNSEFORJ=1TOLEN(V0$);II=ASC(MID$(V0$,J,N1));V$=V$+CHR$(ABS((C155*(II>64))-II));NEXT:RETURN
```

Display message, then end current turn.

```
8 GOSUB4;GOTO2250
```

Descriptions of individual rooms.

```

10 A$="ZM ZRIOLXP. GSVIV RH Z YOFV YF6GLM SVIV";S=2:RETURN
20 A$="Z MZIILD XLIIRWLI";N=1:S=3:RETURN
30 A$="Z MZIILD XLIIRWLI";N=2:S=4:RETURN
40 A$="Z MZIILD XLIIRWLI";N=3:S=5:RETURN
50 A$="Z HNZOO ILLN";N=4:S=6:RETURN
60 A$="Z WVXLMGZNRMZGRML XSZNYVI";B$="GSVIV RH Z YOFV YF6GLM SVI
V";N=5:S=7:RETURN
70 A$="Z HNZOO HGLIZTV XSZNYVI";N=6:S=8:W=12:RETURN
80 A$="Z HNZOO XSZNYVI";N=7:S=9:W=13:RETURN
90 A$="Z HNZOO VOXGILMRX OZYLIZGLIB";N=8:S=10:W=14:RETURN
100 A$="Z YRLOLTRIXO OZYLIZGLIB. GSVIV RH Z IVW YF6GLM LM GSV DZ
OO";N=9:W=15:RETURN
110 A$="Z HGLIZTV XSZNYVI";W=16:RETURN
120 A$="Z OZITV XSZNYVI. GSVIV RH Z XZYRMVG SVIV";S=13:W=17:E=7;
RETURN
130 A$="Z HGIZMTV KFIKOV ILLN. GSVIV RH Z YOFV YF6GLM SVIV";N=12
:S=14:W=18:E=8:RETURN
140 A$="Z HNZOO LUURXV";N=13:S=15:E=9:RETURN
150 A$="Z HNZOO ILLN DRGS Z XZIW GZYOV RM GSV XVMGVI";N=14:S=16;
W=20:E=10:RETURN
160 A$="Z OZITV LUURXV. GSVIV RH Z WVHP SVIV";N=15:W=21:E=11:RET
URN
170 A$="Z LUURXV DRGS Z OZITV WVHP";S=18:E=12:RETURN
180 A$="Z HGLIZTV ILLN";N=17:S=19:E=13:RETURN
190 A$="Z OZITV SZOO";N=18:S=20:RETURN
200 A$="ZM VMGVI6ZRMVMG ILLN. Z HXIVVM IVHGH LM GSV DZOO";B$="6
SVIV RH Z YOFV ZMW Z IVW YF6GLM FMWVI GSV HXIVVM";N=19:S=21:E=15
:RETURN
210 A$="Z WZGZ IVXLIW HGLIZTV ILLN";N=20:E=16:RETURN
220 A$="IZWZI XLMGILD. GSVIV RH Z HNZOO HXIVVM SVIV";S=23:W=27:R
ETURN
230 A$="Z NVWRXZO HGZGRML. GSVIV RH Z OZITV GZYOV SVIV";N=22:S=2
4:W=28:RETURN
240 A$="Z HVXFIRGB HGZGRML";N=23:RETURN
250 A$="Z IZWRL ILLN";S=26:W=30:RETURN
260 A$="Z HNZOO ILLN. GSVIV RH Z HZUV RM GSV HLF6S DZOO";N=25:W=
31:RETURN

```

```

270 A$="GSV ILYLG XLMGILD XVMGVI";B$="GSVIV RH Z HNZOO XLMGILD X
LNKFBVI NLFMGVM RM GSV DZOO";W=32:E=22:RETURN
280 A$="GSV DVZKLMH HGLIZTV ILLN";S=29:W=33:E=23:RETURN
290 A$="Z ORYZIZB";N=28:S=30:W=34:RETURN
300 A$="Z HVXFIRGB XSVXP ZIVZ";N=29:W=36:E=25:RETURN
310 A$="Z HNZOO ILLN DRGS Z WVHP. Z HRTM IVZWH";B$="KIVHHFIV GL
IVZXGLI." Z YOFV YF6GLM RH DLXZGVM FMWVI GSV HRTM";E=26:RETURN
320 A$="OZFMXS XLMGILD. GSVIV RH Z WVHP SVIV";S=33:W=37:E=27:RET
URN
330 A$="Z HNZOO XLIIRWLI";N=32:S=34:E=28:RETURN
340 A$="Z HGIZMTV YOFV ILLN. GSVIV RH Z IVW YF6GLM SVIV";N=33:S=
35:E=29:RETURN
350 A$="Z GRMB HGLIZTV ILLN";N=34:W=40:RETURN
360 A$="Z HNZOO XSZNYVI. GSVIV RH Z DVHG MLLI SVIV";E=30:RETURN
370 A$="Z HNZOO, MZIILI XLIIRWLI";N=42:E=32:RETURN
380 A$="GSV XLNKFBVI XVMGVI. GSVIV RH Z HNZOO HOLG";B$="RM GSV X
LNKFBVI";S=39:RETURN
390 A$="GSV XSVNRXZO OZY";N=38:S=40:RETURN
400 A$="GSV IVZXGLI XLMGILD XVMGVI. GSVIV RH Z YOFV YF6GLM";B$="
ZMW Z IVW LMV. Z HRTM HZB 'IVZXGLI XLMGILD - IVW=LM, YOFV=LUU";
N=39:E=35:RETURN
410 A$="GSV MFXOVZI IVZXGLI. Z XLNKFBVI IVHGH LM GSV DZOO";E=36;
RETURN
420 A$="GSV DVHG VMW LU Z OLMT XLIIRWLI";S=37:E=43:RETURN
430 A$="GSV VZHG VMW LU Z OLMT XLIIRWLI";W=42:E=44:RETURN
440 A$="Z HVXFIRGB XVMGVI";W=43:E=45:RETURN
450 A$="Z HNZOO OZFMXS TZ6V";B$="GSVIV RH Z HNZOO HOLG MVCG GL G
SV OZFMXS TZ6V";W=44:RETURN
460 A$="Z HNZOO HKZXVXIZUG. GSVIV RH Z HNZOO HOLG RM GSV NZH6VI"
;B$="OZFMXS XLNKFBVI";W=45

```

Extended descriptions of current location

```

470 IFA=10AND(D3=10RD3=2)THENC$="GSV NLMHGVI XZTV RH LKVM"
480 IFA=12ANDD5=0THENC$="GSV XZYRMVG RH OLXPVM"
490 IFA=12ANDD5=1THENC$="GSV XZYRMVG RH LKVM"
500 IFA=20ANDD6=0THENC$="GSV HXIVVM RH YOZMP"
510 IFA=20ANDD6=1THENC$="Z NLERV RH YVRMT KOZBYM LM GSV HXIVVM"
520 IFA=26ANDD9=0THENC$="GSV HZUV RH OLXPVM"
530 IFA=26ANDD9=1THENC$="GSV HZUV RH LKVM"
540 IFA=27ANDE2=0THENC$="GSV XLNKFBVI RH XZGREV"
550 IFA=27ANDE2=1THENC$="GSV XLNKFBVI RH WVHGILBVW"
560 IFA=36ANDE6=0THENC$="GSV IVZXGLI MLLI RH URINOB OLXPVM"
570 IFA=36ANDE6=1THENC$="GSV IVZXGLI MLLI RH LKVM";W=41
580 IFA=45ANDE9=0THENC$="GSV OZFMXS TZ6V RH XOLHVW"
590 IFA=45ANDE9=1THENC$="GSV OZFMXS TZ6V RH LKVM";E=46

```

Generate list of visible items and available exits for current location.

```

600 A$=A$+". ";IFLEN(B$)>3THENB$=B$+ ". "
610 IFLEN(C$)>3THENC$=C$+ ". "
650 IFN<>0THENC$="MLIGS "
660 IFS<>0THENC$=E$+"HLFGS "
670 IFW<>0THENC$=E$+"DVHG "
680 IFE<>0THENC$=E$+"VZHG "
690 IFE<>""THENC$=LEFT$(E$,LEN(E$)-1)

```

Describe current location, visible items, and available exits.

```

700 CLS:PRINT"YOU ARE IN: ";P$=A$:GOSUB4:P$=B$:GOSUB4
710 P$=C$:GOSUB4
720 PRINT:PRINT"OBJECTS YOU CAN SEE: ";P$=" ";FORT=1TO16;IFI(T)=
ATHENP$="- "+I$(T):GOSUB4
725 NEXT:IFF$=" THENP$=" MLGSRMT":GOSUB4
730 PRINT:PRINT"EXITS: ";P$=E$:GOSUB4

```

# NEW CLASSICS SOFTWARE

## Pascal-80 Phelps Gates

This friendly, easy to use version of *Standard Pascal*, as reviewed in the December 1981 *Byte*, is now even better! New version works on TRS-80 Model I and Model III, under TRS-DOS, NewDOS, NewDOS 80, DOSPlus, LDOS, and DoubleDOS. An author package allows you to create your own /CMD files without any royalty payments! Upper and lower case is fully supported. You can protect memory and call machine language programs. New extensions include SET, RESET, POINT, RND, and the UCSD Include procedure. Utilities are provided to convert to and from ASCII files. Pascal 80 now comes in a binder with an 80 page manual by George Blank.

With monitor, editor, and compiler in memory at the same time, no other Pascal is easier to learn! One college found that it could teach half again as many students on the same number of computers after switching from UCSD Pascal to Pascal 80.

Full 14 digit accuracy on all math functions, including log and trig functions, makes this a serious Pascal. Disk file handling is supported, with a mail list program included as a demonstration.

Upgrades are available for those who bought Ramware Pascal 80. Call or write for information.

Send \$101 (includes shipping) to: **New Classic Software**  
239 Fox Hill Road, Box 5  
Denville, NJ 07834



Credit card orders: (201) 625-8838

(PASCAL-80 does not implement variant records, pointer and window variables, or functions and procedures used as parameters.)

Print out additional warnings, messages, etc.

```
740 IF (A=400RA=350RA=300RA=31) ANDI (4)=0ANDF3=0THENP$="GSV HNZOO
YDZXP WVERXV RH YORPRTM":GOSUB4
750 IFA=36ANDI (4)=0ANDF3=0THENP$="GSV HNZOO YOZXP WVERXV RH UOZH
SRMT YIRTS60B":GOSUB4
760 IFF4<>0THENP$="GSV XLNKF6VI HZBH: '+'STR$(F4)+' NRMFGVH FMGR
O MVHGIFXGRM":GOSUB4
770 IFD3=1THENP$="* * * ZORVM NLMH6VI ZG6ZXPRT * * *":GOSUB4
780 IFD7=10RE0=10RE3=10RE7=1THENP$="* * * HVXFIRGB KZG1LO ZG6ZXP
RMT * * *":GOSUB4
```

Get and interpret command.

```
790 PRINT:INPUT"COMMAND";V0$:GOSUB6
800 FORT=1T04:IFV$=LEFT$(V$(T),1)THENV$=V$(T)
810 NEXTT
820 IFLEN(V$)<3THEN700
830 V1$=LEFT$(V$,3):V2$=RIGHT$(V$,3)
840 FORT=1T017:IFV1$=LEFT$(V$(T),3)THENV1=T
850 NEXTT:IFV1=0THENP$="R WLM'G FMVINGZMW DSZG BLF DZMG NV GL W
L":GOTO8
860 FORT=1T016:IFV2$=RIGHT$(I$(T),3)THENV2=T
870 NEXTT
880 ONV1GOTO900,940,980,1050,1130,1310,1370,1390,1560,1620,1660,
1790,1860,1910,2010,2170,2240
890 GOTO2250
```

Command handler routines.

```
900 IFN=0THEN1110
910 IFD3=1THENP$="GSV NLMH6VI YOLXPH GSV VCRG":GOTO8
920 IFD7=10RE3=1THEN6GOTO1100
930 A=N:GOTO2250
940 IFS=0THEN1110
950 IFS=24ANDD8<>0ANDE2<>1THEND7=1:GOSUB1120
960 IFS=30ANDE4<>0ANDE2<>1THENE3=1:GOSUB1120
970 A=S:GOTO2250
980 IFW=0THEN1110
990 IFD3=1THENP$="GSV NLMH6VI YOLXPH GSV VCRG":GOTO8
1000 IFE0=10RE3=10RE7=1THEN1100
1010 IFW=41ANDF3=0THENP$="IZWRZGRM UILN GSV IVZXGLI SRGH BLF":G
OSUB4:GOTO2510
1020 IFW=30ANDE4<>0ANDE2<>1THENE3=1:GOSUB1120
1030 IFW=27ANDE1<>0ANDE2<>1THENE0=1:GOSUB1120
1040 A=W:GOTO2250
1050 IFE=0THEN1110
1060 IFE0=10RE3=10RE7=1THEN1100THEN1100
1070 IFE=27ANDE1<>0ANDE2<>1THENE0=1:GOSUB1120
1080 IFE=44ANDE8<>0ANDE2<>1THENE7=1:GOSUB1120
1090 A=E:GOTO2250
1100 P$="GSV HVXFIRGB ZMWILRW YOLXPH GSV VCRG":GOTO8
1110 P$="GSVIV RH ML DZB GL TL GSZG WRIVXGRM":GOSUB4:FORJ=1T010
00:NEXT:GOTO2250
1120 P$="Z HVXFIRGB ZMWILRW ZDZRGH BLF":GOSUB4:RETURN
1130 IFA=1ANDV2$="LXP"THENP$="GSV ZRIOLXP LKVMH ZMW BLF ZIV YOLD
M LFG RMGL GSV EZXFFN LU HKZXV":GOSUB4:GOTO2510
1140 IFA=12ANDV2$="MVG"ANDD5=0ANDI (2)<>0THENP$="BLF ZIVM'G HGILM
T VMLFTS GL ULIXV RG LKVM":GOTO8
1150 IFA=12ANDV2$="MVG"ANDD5=0ANDI (2)=0THENP$="GSV XILDYZI SVOKV
W. GSV XZYRMVG RH MLD LKVM":D5=1:I (5)=ABS(I (5)):GOTO8
1160 IFA=12ANDV2$="MVG"ANDD5=1THENP$="GSV XZYRMVG RH ZOIVZWB LKV
M":GOTO8
1170 IFA=16ANDV2$="VHP"THENP$="LP":I (6)=ABS(I (6)):GOTO8
1180 IFA=17ANDV2$="VHP"THENP$="LP":I (7)=ABS(I (7)):GOTO8
1190 IFA=26ANDV2$="ZUV"ANDD9=1THENP$="GSV HZUV RH ZOIVZWB LKVM":
GOTO8
1200 IFA=26ANDV2$="ZUV"ANDD9=0THENP$="R WLM'G SZEVE GSV PVB GL LK
VM GSV HZUV":GOTO8
1210 IFA=31ANDV2$="VHP"THENP$="LP. BLF URMW ML6SRMT RMHRWV":GOTO
8
1220 IFA=32ANDV2$="VHP"THENP$="LP":I (14)=ABS(I (14)):GOTO8
1230 IFA=36ANDV2$="LLI"ANDE6=1THENP$="GSV WLLI RH ZOIVZWB LKVM":
GOTO8
1240 IFA=36ANDV2$="LLI"ANDE6=0ANDI (6)<>0THENP$="BLF WLM'G SZEVE G
SV PVB GL GSV WLLI":GOTO8
1250 IFA=36ANDV2$="LLI"ANDE6=0ANDI (6)=0ANDE5=0THENP$="BLF ZIV HF
XPVM RMGL GSV FMKIVHFFIRAVM IVZXGLI YFROWRMT":GOSUB4:GOTO2510
1260 IFA=36ANDV2$="LLI"ANDI (6)=0THENP$="GSV WLLI RH MLD LKVM":E6
=1:GOTO8
1270 IFA=41ANDV2$="MVO"THENP$="GSV KZMVD RH URINOB DLXPVM":GOTO8
1280 IFA=45ANDV2$="LXP"ANDE9=1THENP$="GSV ZRIOLXP RH ZOIVZWB LKV
M":GOTO8
1290 IFA=45ANDV2$="LXP"ANDE9=0THENP$="GSVIV ZIVM'G ZMB ERHRYOV X
LMG1LOH":GOTO8
1300 P$="R XZM'G WL GSZG":GOTO8
1310 IFV2$="6VI"ORV2$="LRW"THENP$="WLM'G YV IRWRXFOLF":GOTO8
1320 IFV2=0THENP$="R XZM'G WL GSZG":GOTO8
1330 IFI (V2)=0THENP$="BLF ZOIVZWB SZEVE GSZG":GOTO8
1340 IFI (V2)<>ATHENP$="R WLM'G HVV RG SVIV":GOTO8
1350 IFP4)=8THENP$="HLIIB, BLF XZM'G XZIIB ZMB6SRMT NLIIV":GOTO8
1360 P4=P4+1:I (V2)=0:P$="LP":GOTO8
1370 IFV2=0ORI (V2)<>0THENP$="BLF WLM'G SZEVE GSZG":GOTO8
```

1380 P4=P4-1:I(V2)=A:P\$="LP":GOTO8  
1390 IFI(5)<>0THENP\$="BLF WLM'G SZEZ Z DVZKLM":GOTO8  
1400 IFA=1ANDV2\$="LXP"THENP\$="BLF ZIV YOLDM LFG LU GSV ZRIOLXP R  
MGL GSV EZXFFN LU HKZXV":GOSUB4:GOTO2510  
1410 IFA=27ANDV2\$="GVI"THENP\$="GSV XLNKFVGI RH WVHGILBVM":E2=1:E  
0=0:GOTO8  
1420 IFA=38ANDV2\$="GVI"THENP\$="GSV HSLG IVUOVXGH LUU LU GSV XLNK  
F6VI":GOSUB4:GOTO2510  
1430 IFA=41ANDV2\$="GVI"THENP\$="GSV DSOV MFXOVZI IVZXGLI RH VCKO  
LWRMT":GOSUB4:GOTO2510  
1440 IFV2\$="RWH"ORV2\$="YLG"ORV2\$="ILO"ORV2\$="IWH"ORV2\$="ZIW"THEN  
V2\$="LRW"  
1450 IFV2\$<>"GVI"ANDV2\$<>"LRW"THENP\$="GSV DZHV HSLG SZH ML VUUV  
X6":GOTO8  
1460 IFV2\$="GVI"ANDD3=0THENP\$="R WLM'G HVV ZMB NLMHGV SVIV":GOT  
O8  
1470 IFV2\$="LRW"ANDD7=0ANDE0=0ANDE3=0ANDE7=0THENP\$="R WLM'G HVV  
ZMB ZMWILRW SVIV":GOTO8  
1480 T=RND(100):IFT>P2+P3+50THENP\$="BLF URIV ZMW NRHH":GOTO8  
1490 IFD3=1THENP\$="BLF SRG GSV NLMHGV":GOSUB4:D4=D4-((10+P2+P3)  
/2):IFD4<=0THEND3=0:D4=0:P\$="BLF SZEZ PROOVW RG":GOTO8  
1500 IFD7=1THENP\$="BLF SRG GSV ZMWILRW":GOSUB4:D8=D8-((5+P2+P3)/  
2):IFD8<=0THEND7=0:D8=0:P\$="RG RH WVHGILBVM":GOTO8  
1510 IFE0=1THENP\$="BLF SRG GSV ZMWILRW":GOSUB4:E1=E1-((5+P2+P3)/  
2):IFE1<=0THENE0=0:E1=0:P\$="RG RH WVHGILBVM":GOTO8  
1520 IFE3=1THENP\$="BLF SRG GSV ZMWILRW":GOSUB4:E4=E4-((5+P2+P3)/  
2):IFE4<=0THENE3=0:E4=0:P\$="RG RH WVHGILBVM":GOTO8  
1530 IFE7=1THENP\$="BLF SRG GSV ZMWILRW":GOSUB4:E8=E8-((5+P2+P3)/  
2):IFE8<=0THENE7=0:E8=0:P\$="RG RH WVHGILBVM":GOTO8  
1540 IFD3=1THENP\$="RG RH HGR00 ZOREV":GOTO8  
1550 P\$="GSV ZMWILRW RH HGR00 UFMXGRLMRMT":GOTO8  
1560 IFV2=0THENP\$="R XZM'G WL GSZG":GOTO8  
1570 IFI(V2)<>0THENP\$="R WLM'G SZEZ GSZG":GOTO8  
1580 IFV2<>9ANDV2<>14THENP\$="R XZM'G WL GSZG":GOTO8  
1590 IF(V2=9ANDA=44)OR(V2=14ANDA=38)THENP\$="MLGSRMT SZKKVMH":GOT  
O8  
1600 IFV2=9ANDA=38THENFA=35:P\$="GSV XLNKFVGI IVKORVH: 'YZHV WVHG  
IFXG HVJFVMXV HGZIGVM":GOSUB4:P\$="WVHGIFXGRLM RM"+STR\$(F4)+" NRM  
F6VH.'":P4=P4-1:I(9)=100:GOTO8  
1610 IFV2=14ANDA=45THENP\$="GSV TZGV LKVMH":E9=1:GOTO8  
1620 IFV2<>10THENP\$="WLM'G YV IRWRXFOLFH":GOTO8  
1630 IFI(10)<>0THENP\$="BLF WLM'G SZEZ GSZG":GOTO8  
1640 P\$="LP":I(10)=50:P4=P4-1:P1=P1+5+P3:IFP0<P1THENP0=P1  
1650 GOTO8  
1660 IFA=1ANDV2\$="OFV"THENP\$="GSV ZRIOLXP LKVMH...":P\$="BLF ZIV  
YOLDM LFG RMGL GSV EZXFFN LU HKZXV":GOSUB4:GOTO2510  
1670 IFA=6ANDV2\$="OFV"THENP\$="Z HGIZMTV, LIZMTV TOLD XLEVIH BLF  
ZMW GSVU UZVWH ZDZB":GOTO8  
1680 IFA=10ANDV2\$="IVW"ANDD6=0THENP\$="MLGSRMT SZKKVMH":GOTO8  
1690 IFA=10ANDV2\$="IVW"THEND6=1:P\$="ZM ZORVM NLMHGV RH IVOVZHVW  
. RG RH ZGGZPRMT BLF":GOTO8  
1700 IFA=13ANDV2\$="OFV"THENA=34:P\$="Z UOZHS LU ORTSG 6VKNLIZIROB  
YORMWH BLF":GOTO8  
1710 IFA=20ANDV2\$="IVW"ANDD6=0THENP\$="MLGSRMT SZKKVMH":GOTO8  
1720 IFA=20ANDV2\$="IVW"THEND6=0:P\$="GSV HXIVVM TLVH YOZMP":GOTO8  
1730 IFA=20ANDV2\$="OFV"THEND6=1:P\$="GSV HXIVVM ORTSGH FK":GOTO8  
1740 IFA=31ANDV2\$="OFV"THENA=51:P\$="LP":GOTO8  
1750 IFA=34ANDV2\$="IVW"THENA=13:P\$="Z UOZHS LU ORTSG 6VKNLIZIROB  
YORMWH BLF":GOTO8  
1760 IFA=40ANDV2\$="IVW"THENA=0:P\$="LP":GOTO8  
1770 IFA=40ANDV2\$="OFV"THENA=31:P\$="LP":GOTO8  
1780 P\$="MLGSRMT SZKKVMH":GOTO8  
1790 IFA=22ANDV2\$="VVM"THENP\$="BLF XZM HVV MLGSRMT LU RMGIVVHG L

M GSV IZWI":GOTO8  
1800 IFV2=0THENP\$="R WLM'G SZEZ GSZG":GOTO8  
1810 IFI(V2)<>0ANDI(V2)<>0ATHENP\$="R WLM'G SZEZ GSZG":GOTO8  
1820 IFV2=3ORV2=13THENP\$="R HVV MLGSRMT HKVXRZ0":GOTO8  
1830 IFV2=9THENP\$="HLIIB, LMOB Z XLNKFVGI XZM IVZW Z KILTIZN":GO  
TO8  
1840 IFV2=16THENP\$="GSV KOZMH ZIV HVZOVW...LMOB XLNNZMW XZM LKVM  
GSVN":GOTO8  
1850 P\$="R XZM'G IVZW GSZG":GOTO8  
1860 CLS:P\$=" \* \* \* KOZBVI'H RMEVMGLIB \* \* \*":GOSUB4:PRINT  
1870 FORT=1TO16:IFI(T)=0THENP\$="- "+I\$(T):GOSUB4  
1880 NEXTT  
1890 PRINT:PRINT"HIT ANY KEY TO CONTINUE"  
1895 X\$=INKEY\$:IFX\$=""THEN1895  
1900 GOTO2400  
1910 IFV2=0THENP\$="R XZM'G WL GSZG":GOTO8  
1920 IFI(V2)<>0THENP\$="R WLM'G SZEZ GSZG":GOTO8  
1930 IFV2=1ANDA=12ANDD5=0THENP\$="GSV XZYRMVG QLXP RH WVHGILBVM":  
D5=1:I(1)=100:I(5)=ABS(I(5)):P4=P4-1:GOTO8  
1940 IF(V2=1ORV2=15)AND(D3=1ORD7=1ORE0=1ORE3=1ORE7=1)THENI(V2)=1  
00:P4=P4-1:GOTO1490  
1950 IF(V2=1ORV2=15)ANDA=1THENP\$="GSV ZRIOLXP RH WVHGILBVM...BLF  
ZIV YOLDM LFG RMGL GSV EZXFFN LU HKZXV!":GOSUB4:GOTO2510  
1960 IF(V2=1ORV2=15)ANDA=36ANDE6=0ANDE5=0THENP\$="GSV WLLI RH WVH  
GILBVM...BLF ZIV HFXPVW RMGL GSV MLM-":GOSUB4:P\$="KIVHFRIRAVM IV  
ZXGLI YFROWRMT":GOSUB4:GOTO2510  
1970 IF(V2=1ORV2=15)ANDA=36ANDE6=0ANDF3=0THENP\$="GSV WLLI RH WVH  
GILBVM. BLF ZIV YLNYZIVW DRGS IZWRZGRLM":GOSUB4:GOTO2510  
1980 IF(V2=1ORV2=15)ANDA=36ANDE6=0THENP\$="GSV WLLI RH WVHGILBVM"  
:E6=1:I(V2)=100:P4=P4-1:GOTO8  
1990 IFV2=1ORV2=15THENP\$="GSV "+I\$(V2)+" SZH ML VUUVXG":I(V2)=10  
0:P4=P4-1:GOTO8  
2000 GOTO1370  
2010 IFV2=0THENP\$="R XZM'G WL GSZG":GOTO8  
2020 IFI(V2)<>0THENP\$="R WLM'G SZEZ GSZG":GOTO8  
2030 IFV2=5ANDD3=1THENV2\$="GVI"  
2040 IFV2=5AND(D7=1ORE0=1ORE3=1ORE7=1)THENV2\$="LRW"  
2050 IFV2=5THEN1390  
2060 IFV2=4ANDF3=0AND(A=40ORA=35ORA=30ORA=31)THENP\$="GSV YOZXP W  
VERXV RH YORMPRNT":GOTO8  
2070 IFV2=4ANDF3=0ANDA=36THENP\$="GSV YOZXP WVERXV RH UOZHSRMT YI  
RTSGOB":GOTO8  
2080 IFV2=4THENP\$="GSVIV ZIVM'G ZMB ERHRYOV XLMGLOH LM GSRH WVE  
RXV":GOTO8  
2090 IFV2=12THENI(12)=A:P4=P4-1:I\$(12)="ARMED PHOTON BOMB":F2=35  
:P\$="GSV KSLGLM YLNY DROO VCKOLWV RM 35 NRMF6VH":GOTO8  
2100 IFV2=2ANDA=12ANDD5=0THEND5=1:P\$="GSV XZYRMVG RH MLD LKVM":I  
(5)=ABS(I(5)):GOTO8  
2110 IFV2=2ANDA=12ANDD5=1THENP\$="GSV XZYRMVG RH ZOIVZWB LKVM":GO  
TO8  
2120 IFV2=7ANDA=26ANDD9=0THEND9=1:I(16)=ABS(I(16)):P\$="GSV HZUV  
LKVMH":GOTO8  
2130 IFV2<>11THENP\$="DSZG WL BLF DZMG NV GL WL DRGS GSV "+I\$(V2)  
+"?":GOTO8  
2140 IFI(8)<>0THENP\$="GSVIV ZIVM'G ZMB YZGGVIRVH ULI GSV IZWL":  
GOTO8  
2150 IFF2<>0THENP\$="Z ELRXV HZBH 'YLN Y HGZGFH:"+STR\$(F2)+"NRMF6V  
H FMGR0 WVGLMZGRLM":GOTO8  
2160 P\$="GSV IZWL RH HROVMG":GOTO8  
2170 CLS:P\$=" \* \* \* KOZBVI'H HGZGFH \* \* \*":GOSUB4:PRINT  
2180 P\$="XFIVMG SRG KLRMGH =" +STR\$(P1):GOSUB4  
2190 P\$="MVCVIRGB ZGGIRYFV=" +STR\$(P2):GOSUB4  
2200 P\$="OFXV ZGGIRYFV =" +STR\$(P3):GOSUB4

```

2210 PRINT:PRINT"HIT ANY KEY TO CONTINUE"
2220 X$=INKEY$:IFX$=""THEN2220
2230 GOTO2400
2240 CLS:PRINT"GAME OVER":GOTO 2520

```

Update player status, and conduct combat if appropriate.

```

2250 IFF2<>0THENF2=F2-1:IFF2<=0THEN2430
2260 IFF4<>0THENF4=F4-1:IFF4<=0THEN2470
2270 IFP1<P0THENP5=P5+.5:IFP5=1THENP5=0:P1=P1+1
2280 IFD3=0ANDD7=0ANDE0=0ANDE3=0ANDE7=0THEN2390
2290 T=RND(100)
2300 IFD3=1THENP$="GSV NLMHGI VZGZXPX ":GOSUB4
2310 IFD3<>1THENP$="GSV HVXFIRGB ZMWILRW HSLLG... ":GOSUB4
2320 IFT>80-(P2+P3)THENP$="RG NRHHVH":GOSUB4:GOTO2390
2330 P1=P1-(RND(5)+RND(5)+RND(5)+RND(5)+15-P3)
2340 IFD3<>1THENP1=P1+5
2350 IFP1<0THEN2510
2380 P$="BLF ZIV SRG!":GOSUB4
2390 IFV1=0ORV1>4OR(D3+D7+E0+E3+E7>0)THENFORJ=1TO1000:NEXT

```

Initialize for new turn and jump to appropriate room description.

```

2400 V$="":V1$="":V2$="":V1=0:V2=0:A$="":B$="":C$="":D$="":E$="":
:N=0:S=0:W=0:E=0
2410 ONAGOSUB10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,
160,170,180,190,200,210,220,230,240,250,260,270,280,290,300,310,
320,330,340,350,360,370,380,390,400,410,420,430,440,450,2550:GOT
0470
2420 GOTO10

```

Evaluate end-game conditions and display appropriate messages.

```

2430 CLS:IFA=46THENF4=-1:GOTO2550
2440 P$="GSV KSLGLM VLYN VCKOLVWH...GSV VMGRIV XLNKOVC RH VVHGIL
BVW":GOSUB4
2450 P$="BLF SZEY YVVM PROOVV YB GSV ULIXV LU GSV YZHG":GOSUB4
2460 PRINT:PRINT:GOTO2510
2470 CLS:IFA=46THEN2550
2480 IFA=38THENP$="GSV XLNKFVI UOZHSVH YIRTS6OB, VNR6GRMT HKZIP
H RM ZOO WRIVXGRLMH":GOSUB4
2490 P$="GSV XLNKOVC HFVVVMOB VCKOLVWH RMLG NROORLMH LU KRXXVH":
GOSUB4
2500 P$="BLF ZIV PROOVV YB GSV UZOOMT WYVIRH ZILFMW BLF":GOSUB4
:PRINT:PRINT
2510 P$="BLF ZIV MVZW!":GOSUB4
2520 INPUT"DO YOU WANT TO PLAY AGAIN";A$
2530 IFLEFT$(A$,1)="Y"THEN2610
2540 CLS:END
2550 P$="GSV HKZXV HSRK HFVVVMOB DRUGH RMLG LIYRG ZILFMW GSV KOZ
MVG":GOSUB4
2560 IF((F2=0)OR(F2<>0ANDI(12)<>41))ANDF4=0THENP$="BLF WRWM'G WV
HGILB GSV YZHV. NRHHRLM UZROVV":GOSUB4:PRINT:GOTO2520
2570 P$="UILN Z WRHGZMXV, BLF XZM HVV GSV ZORVM YZHV VCKOLWV":GO
SUB4
2580 IFI(16)<>0THENP$="BLF WRWM'G IVXLEVI GSV HVXIVG KOZMH MVVWV
W YB HGZI XLNNZMW":GOSUB4:PRINT:GOTO2520
2590 P$="NRHHRLM RH Z HFXXVHH!":GOSUB4
2600 GOTO2520

```

Initialize workspace. Read in items, and verbs.

```

2610 CLEAR300:DIMI$(16),I(16),V$(17):C155=-155:C64=64:N1=1
2620 CLS
2630 PRINTTAB(15)"OPERATION: SABOTAGE BY RAY SATO"
2640 FORT=1TO16:READI$(T),I(T):NEXT
2660 FORT=1TO17:READV$(T):NEXT
2690 FORT=1TO40:P0=P0+RND(2):NEXTT

```

Establish player attribute points. Jump to first room.

```

2700 P1=P0
2710 FORT=1TO10:P2=P2+RND(2):NEXTT
2720 FORT=1TO10:P3=P3+RND(2):NEXTT
2730 FORT=1TO50:D4=D4+RND(2):D8=D8+RND(2):E1=E1+RND(2):E4=E4+RND
(2):E8=E8+RND(2):NEXTT
2740 A=1:P4=1
2750 GOSUB10:GOTO470

```

Item and verb data

```

2760 DATAKZHBGX VCKOLHREV,0,XILDYZI,7,XZOVWMI,8,HNZOO YOZXP WV
ERXV,9,OZHVI KRHLG,-12,HVXFIRGB PVB,-16,VOVXGILMRX XLMGIL0 YZGL
M,-17,YZGGVIRVH,18,XLNKFVI VVHGIFXG KILTIZN,21,HROEVI KROD,23,K
LIGZYOV IZWR,25,OZITV KSLGLM VLYN,28
2770 DATATZOZGXRX XSZIG,32,DZFMXS HBHGVN XZHHV66V,-32,MRGILTOBXV
IRM,39,HVXIVG KOZMH,-26
2780 DATAML1GS,HLF6S,DVHG,VZH6,LKVM,TVG,WILK,HSLLG,RMHVIG,VZ6,KF
HS,IVZW,RNEVMGLIB,GSILD,FHV,HGZGFH,JFRG

```

TRS-80® SWAT TABLE FOR: OPERATION: SABOTAGE  
(Modified Parameters: NU = 5, B = 200)

SWAT				SWAT			
LINES	CODE	LENGTH		LINES	CODE	LENGTH	
2 -	10	XO	238	1410 -	1430	XH	209
20 -	60	SU	221	1440 -	1470	QG	259
70 -	100	MX	222	1480 -	1500	QP	253
110 -	140	HV	221	1510 -	1530	NA	297
150 -	180	CR	228	1540 -	1580	QP	197
190 -	210	DI	213	1590 -	1610	RA	240
220 -	250	XZ	207	1620 -	1660	NP	244
260 -	280	LY	225	1670 -	1690	RP	214
290 -	310	SE	207	1700 -	1730	QS	231
320 -	350	RL	221	1740 -	1780	HC	209
360 -	390	NU	223	1790 -	1820	KF	204
400 -	410	LS	206	1830 -	1860	WX	206
420 -	450	XG	232	1870 -	1900	FU	108
460 -	490	BT	225	1910 -	1940	IL	236
500 -	540	JL	217	1950 -	1960	BH	244
550 -	590	HK	245	1970 -	1990	HY	255
600 -	670	RV	127	2000 -	2040	XW	151
680 -	720	SS	177	2050 -	2080	GF	222
725 -	750	GF	214	2090 -	2110	HA	237
760 -	780	JF	201	2120 -	2150	BR	272
790 -	830	NU	111	2160 -	2200	UL	193
840 -	880	DT	226	2210 -	2250	SB	121
890 -	930	IZ	105	2260 -	2300	LD	157
940 -	980	NB	108	2310 -	2350	LT	165
990 -	1030	ZL	202	2380 -	2410	YE	339
1040 -	1080	RC	124	2420 -	2460	IW	176
1090 -	1130	IW	272	2470 -	2500	ZN	230
1140 -	1160	JR	246	2510 -	2550	UY	163
1170 -	1200	ZO	217	2560 -	2580	LY	250
1210 -	1240	IC	236	2590 -	2630	GB	145
1250 -	1270	TU	222	2640 -	2710	QD	109
1280 -	1310	KN	209	2720 -	2760	ZZ	348
1320 -	1360	LF	206	2770 -	2780	AZ	184
1370 -	1400	YV	214				



# CATS: Computer Assisted Testing System

## TESTING and SCORE EVALUATION MODULES

### PARTS TWO and THREE

by Jon R. Voskuil  
ATARI® & TRS-80® translations by Alan J. Zett

CATS is an educational utility program for an Apple II™ or Apple II Plus™ with Applesoft, a TRS-80® Model I/Model III, or ATARI® 400/800. The testing module (CATS2) will run on disk or tape. The evaluation module (CATS3) requires a disk drive. Tape versions will run in 16K RAM. Disk versions require 32K RAM. More memory and a printer are optional, but highly recommended.

The first module of CATS, published in the previous issue of *SoftSide*, was designed to allow you to create a computerized test and save it on disk or tape. The second and third modules complete the Computer Assisted Testing System, giving you the capability of administering the created tests and analyzing the results.

#### The Testing Module

Once a test has been created and stored on media by the first CATS module, the second module comes into play. This is a program entirely separate from the first (though it shares some similar sections of code). It enables the teacher to make choices concerning how the test should be administered, and enables the student to take the test on his own with no further teacher involvement. It makes use of two different passwords, along with error-trapping routines, which discourage unauthorized access to the stored tests and scores.

When running the testing module, you will see a title page similar to that of the creation module. The date must be entered in the format indicated, and you are prompted to press any key to continue. Doing so will cause certain default options to be chosen, and (in the tape version only) you will be instructed to insert the proper tape for loading. The next display prompts the student to enter his name, the name of the test he is taking (disk version only), and the password for that test. This password is the one chosen when the test was created with the first module, and must be supplied to the student by the teacher in order to access the test. (If desired, the same test can be stored on different disks or tapes with alter-

nate passwords, using the features of the creation module.) In the disk version, it is at this point that a prompt is given to insert the test disk, and it is read into memory. (Notice that tests are saved by the creation module with the prefix "Q." on the Apple™, the suffix "/Q" on the TRS-80®, and the suffix ".Q" on the ATARI®. This prefix/suffix should **not** be entered as part of the test name.)

The test questions are then displayed, one at a time, just as they were created by the first module. An information line at the top will show the question number, its relative value, and, if a hint is available, the penalty for using it. The student can enter his answer, or he may enter "H" to see the hint or "S" to skip the question for the moment. There is no penalty for skipping the question — unless, of course, he doesn't come back later to answer it.

After an answer has been typed in, it is immediately evaluated for the student. If correct, it is simply reinforced with the message "CORRECT". If not, the correct answer is given. This enhances the learning process and is uniquely possible because of the computerization of the test. However, the teacher can choose other "feedback" options, as described below.

After all the questions have been presented, the student is given the opportunity to review questions he skipped, or to end the test. There is no limit on the number of times the skipped questions can be reviewed. Once the choice is made to end the test, the overall score (in both points and percentage) is shown on the screen. In the disk version, the scoring information is recorded on the disk for later evaluation by the teacher. Scores for

all students taking the same test on the same disk will be written in a single file. In the tape version, the scores for up to twenty students are retained in memory for the teacher to see later. Finally, the prompt line is displayed, at which point pressing RETURN (or ENTER) will repeat the whole process for another student.

As mentioned above, there are options which may be chosen by the teacher to change the way in which the test is administered. These options may be accessed when the "PRESS RETURN TO CONTINUE" message is displayed before (or between) tests, by pressing CTRL-O instead of RETURN. Note that the CTRL key is simulated on the TRS-80® by the combination SHIFT/DOWN-ARROW. You will then be asked to enter the master password, which is presumably known only by the teacher and is different than the individual test passwords. This master password is tucked away in line 15020 (15060 in the tape version), and can be assigned any string you like. For added security, it could be defined in a more roundabout way, such as:

```
MP$ = CHR$(80) + CHR$(65) + ...
```

Obviously, such passwords will not long deter someone with computer knowledge if they are determined to cheat; they are intended only as barriers to the curious.

Once the master password is entered, a series of option menus will be presented. In every case, option number 1 is the default the computer would have chosen if you had not accessed this section of the program. The first menu offers four choices for scoring feedback. Option 1 gives immediate feedback to the student as

soon as he enters his answer to each question, as described above. Option 2 gives the same kind of question-by-question feedback, but not until the test has been completed — at that time, every question is re-displayed, with the points received and (if necessary) the correct answer. Option 3 withholds question-by-question scoring, and gives only the final score at the end. Option 4 withholds all scoring information, including the final score.

The second options menu allows you to alter the number and order of the questions given. Option 1 causes all of the questions to be presented, in sequence. Option 2 allows you to present a specified number of the questions in a random manner. If, for example, you specify ten questions of a fifty question test, the computer will randomly choose ten questions out of the fifty, and present them in a random order. Every student who takes the test with this option in effect will get a **different** random group of ten questions. Option 3 allows you to choose a given range of questions to present from the total group; it requires you to enter the beginning and ending question numbers, to be presented in normal (not random) sequence.

The third options menu enables you to override whatever values you originally assigned to the questions when you created the test, making them all of equal value. Option 1 leaves the values as originally entered, while Option 2 makes them all equal. Likewise, the fourth options menu allows you to make hints unavailable for all questions, even if you originally entered hints for some. If you retain the original hints, the fifth menu allows you to keep the hint penalties as you entered them, or to override those by assigning a fixed penalty across the board.

Having chosen among all these options, you will be asked to confirm the choices, and then you'll be returned to the point of entering the student's name for the next test. The options chosen will remain in effect until changed by using the CTRL-O command prior to another test. Notice that in the disk version, a different test may be chosen by each student; in the tape version, only the test in memory is available, until the program is re-RUN.

In addition to CTRL-O, one or two other commands can be given when the "PRESS RETURN TO CONTINUE" prompt is displayed. In both the disk and tape versions, CTRL-Q is the "quit" command which terminates the program. In the tape version only, CTRL-S (followed by entering the master password) provides access to

the scores of the students who have taken the test thus far. If the number of students reaches 20, no more tests can be administered until the CTRL-S command has been issued. After those 20 scores are displayed, they will then be erased from memory to make room for up to 20 more. No such score reporting function is in the disk version of the testing module; this function is implemented in a more sophisticated way in the separate score evaluation module, described below.

### The Score Evaluation Module

This module operates only with a disk system. It is a separate program, designed to read and analyze the scoring data (files with an "S" prefix or suffix) the testing module records on the disk each time a test is administered. (A suggestion: Save the first module under the file name "CREATE", the second under the file name "TEST", and this third under the file name "SCORE".)

After the title page is displayed, you are prompted to insert the disk containing test scores, and a catalog of the disk is displayed (Apple™ version only). After typing in the appropriate file name and the master password which was used in the testing module, the disk will come on for a few moments while names, dates, and final test scores are read into memory. This program is designed to read and analyze scoring information for as many as 100 students who have taken a given test. In the unlikely event that the group of students is larger than this, the test should be administered from more than one disk, with no more than 100 students using a given disk. (Making backups of the score files after the tests have been administered is strongly recommended!)

The first display you will see after this will show the number of students who have taken the test, along with the lowest, highest, and average scores. It will then list every student whose results are recorded on the disk, along with the dates on which they took the test. Two different scores will be displayed for each student: The actual percentage score which he achieved, and (in parentheses) his percentage score "on the curve." The latter is calculated by giving the highest score a value of 100% and adjusting all the scores upward proportionally. If more than one screen is needed to list all the information, the display will pause as necessary.

When this display is complete, you will have several options, summarized by a prompt line at the bottom of the

screen. Pressing R will review the same display again (useful at this point only for multiple-page displays). Pressing I, and then entering a number corresponding to one of the students, will cause more information to be read and displayed about that individual's performance on the test. This will show which questions were given to him, in what order, and the number of points he received for his answer to each question. An "S" is displayed if a question was skipped and never answered. Pressing P will print the displayed information (either the whole summary or the individual test information) to a printer. The Apple™ version assumes that the printer interface is in slot 1. Pressing C will continue to the next part of the analysis; N will abort further analysis of this score data and prompt you for a new file name; Q will quit (after asking for confirmation).

The final part of the analysis looks at the test results by question number, to determine which questions were missed most often by the students. The display is a chart of all 100 possible question numbers. Following each question number actually used on the test, two numbers will appear and be updated as the information is read from the disk. The first of these is the number of tests on which that particular question appeared, helpful if only a random assortment or a part of the test was given to some or all the students. The second number is the percentage of wrong answers given to that question. Thus, those questions with higher percentages were the most often missed. (Because of the limited space on the display, a "99" will be displayed instead of "100" if a question was missed 100% of the time.)

Once this analysis is complete, a prompt line gives you three options: Pressing P will give you a printout of the question-by-question analysis; Q will quit; and C will continue to another menu. That menu also gives three options: review the same score analysis from the beginning; read new score data from the disk; or quit.

*The Computer Assisted Testing System* is a useful tool for the teacher or parent who wants to put a microcomputer to work in a variety of ways in an educational environment. Many enhancements are possible, of course, and users are encouraged to send modifications to share with others.

### Variables

AZ: Temporary numeric variable (ATARI and TRS-80).

A\$: An Answer (Apple and TRS-80).

A\$: String of Answers (ATARI).  
A\$(\*): Array of answers (Apple and TRS-80).  
A1: Pointer to first answer choice (multiple choice) or first correct answer (fill-in).  
AP: Pointer to an answer in the A\$ array.  
B\$: Backspace character.  
BEL\$: Bell character.  
BR\$: String consisting of a right and left bracket (Apple).  
C: Number of the correct answer choice (multiple choice) or number of characters to match (fill-in).  
C\$: A character.  
CLR: Keyboard-strobe address (Apple).  
D\$: CTRL-D for disk access (Apple).  
E: Error code number.  
ERL: Line where error occurred.  
EF: Error flag.  
F\$: Name of test.  
FIX: Address of ONERR-fix routine (Apple).  
H\$: String of hints (ATARI).  
H\$(\*): Array of hints (Apple and TRS-80).  
HP: Hint penalty.  
HS: Hint status of a question as originally entered.  
HYNT: Hint status flag.  
I: Local loop counter.  
IT: Value returned by menu selection routine.  
IS: Temporary string variable (ATARI).  
J: Local loop counter.  
JJ: Temporary work variable.  
KEY: Keyboard address (Apple).  
L: Temporary used to hold the length of a string.  
MODE: Testing mode: all questions, random selection, or range.  
MP\$: Master password.  
N: Number of answers. Also, number of menu choices.  
N\$: Menu option (CATS2) or Name of person taking test (CATS3).  
N\$(\*): Menu selection options (Apple and TRS-80).  
NF\$: First name of person taking test.  
NQ: Number of questions.  
NUM: Number of questions to be given.  
P\$: Password typed in by user.  
PAF: Password flag.  
PFLAG: Hint penalty flag.  
PRESS: Equals 128. Used in WAIT statement to detect keypress (Apple).  
PW\$: Password for this test.  
Q: Question number.  
Q\$: String of questions (ATARI)  
Q\$: A question (Apple and TRS-80)  
Q\$(\*): Array of questions (Apple and TRS-80)  
Q(\*) : Array holding pointers to questions being administered (ATARI).

Q%(\*): Array holding pointers to questions being administered (Apple and TRS-80).  
Q1: First question number (temporary variable)  
Q2: Last question number (temporary variable).  
QF: First question.  
QF\$: Full question file name (with prefix/suffix)  
QL: Last question.  
QN: Question number.  
QV: Question value.  
R: Random number used in scrambling question.  
RF: Returned-from-subroutine flag.  
S: Speaker click address (Apple and ATARI)  
S(\*): Array of score points for each question (ATARI).  
S%(\*): Array of score points for each question (Apple and TRS-80).  
SF\$: Full file name for score file.  
SM: Scoring mode: immediate, review, final score, or no feedback.  
SP: Location of space in user's name. Used in separating first name.  
SQ: Used in score calculation.  
T: Question type: T/F, multi, fill-in.  
TD\$: Today's date.  
TF: Correct answer for a True/False question (1 = True, 0 = False).  
TF\$: String containing "TRUE" and "FALSE" (ATARI).  
TF\$(\*): Array containing "TRUE" and "FALSE" (Apple and TRS-80)  
TS: Total score achieved.  
TV: Total value of all questions given.  
V: Used for VTAB positioning.  
VFLAG: Question value flag.  
VH: Value of question if hint is given.  
VV: Used in VTAB positioning.  
X: Misc.  
X\$: Input work variable.  
X%: Temporary variable.  
XX%: Temporary variable.  
Z: Time delay loop variable.

```

NC,I,0"
110 TEXT : HOME : VTAB 8: PRINT
    TAB( 17)"C A T S": VTAB 12:
    PRINT TAB( 12)"BY JON R. V
    OSKUIL"
120 VTAB 18: INVERSE : HTAB 14: PRINT
    "TESTING MODULE": NORMAL
130 DIM Q$(100),A$(300),H$(100),
    N$(9),TF$(1),Q%(100),S%(100)

140 TF$(0) = "FALSE":TF$(1) = "TR
    UE"
150 KEY = - 16384:PRESS = 128:CL
    R = - 16368:BR$ = CHR$( 93
    ) + CHR$( 91):S = - 16336:
    BEL$ = CHR$( 7):B$ = CHR$(
    8)
160 GOSUB 15000:VFLAG = 0:PFLAG =
    0:HYNT = 1:MODE = 1:SM = 1:X
    = RND ( - PEEK (78) - 256
    + PEEK (79)): ONERR GOTO
    8000
170 VTAB 21: HTAB 5: PRINT "TYPE
    IN TODAY'S DATE (MM/DD/YY)"
    : HTAB 9: INPUT "AND PRESS R
    ETURN: ";TD$
180 IF LEN (TD$) < > 8 OR MID$
    (TD$,3,1) < > "/" OR MID$
    (TD$,6,1) < > "/" THEN PRINT
    BEL$: GOTO 170
190 FOR I = 1 TO 100:Q%(I) = I:S
    %(I) = 0: NEXT

```

At this point, the user may hit CTRL-Q to quit, CTRL-S to set test options, or RETURN to go on. User must know master password to set options.

```

200 VTAB 24: GOSUB 10000:X = ASC
    (X$): IF X = 13 THEN 1000
210 IF X = 17 THEN HOME : CLEAR
    : END
220 IF X < > 15 THEN 200
300 HOME : VTAB 8: PRINT "MASTER
    PASSWORD: ";P$ = ""
310 GOSUB 10010: IF X$ = CHR$( (
    13) THEN 330
320 POKE 5, PEEK (5): POKE 5, PEEK
    (5): PRINT " ";P$ = P$ + X$
    : GOTO 310
330 IF P$ = MP$ THEN GOSUB 3000

```

340 GOTO 200  
Entry section of routine to administer test. Get student's name, test name, and password.

```

1000 HOME : VTAB 4: PRINT "PLEAS
    E TYPE YOUR NAME (FIRST AND
    LAST), AND PRESS RETURN:": PRINT
    : INPUT " ";N$:L = LEN (N$):
    IF L = 0 THEN 1000

```

# APPLE™

```

SS SS SS SS SS SS SS SS SS SS
SS
SS APPLESOFT BASIC SS
SS 'CATS2 - DISK VERSION' SS
SS AUTHOR: JON R. VOSKUIL SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS

```

Display title page and initialize program.

```

100 D$ = CHR$( 4): PRINT D$"NOMO

```

```

1010 SP = 0: FOR I = 1 TO L: C# =
      MID$(N$,I,1): IF C# = " " THEN
      SP = I
1020 NEXT I: IF SP = 0 OR SP = L
      THEN 1000
1030 NF# = LEFT$(N$,SP - 1): PRINT
      : INPUT "TEST NAME: "; F#: QF#
      = "Q." + F#
1040 PRINT : PRINT "PASSWORD: ";
      : P# = ""
1050 GOSUB 10010: IF X# = CHR#
      (13) THEN 1070
1060 POKE S, PEEK (S): POKE S, PEEK
      (S): PRINT "#"; P# = P# + X#
      : GOTO 1050
1070 PRINT : PRINT : PRINT "INSE
      RT TEST DISK AND PRESS RETUR
      N": GOSUB 10010: PRINT : GOSUB
      7000: IF PAF THEN INVERSE :
      PRINT BEL#;"INCORRECT PASSW
      ORD": NORMAL : GOTO 200
1080 IF EF THEN 200
1090 SF# = "S." + F#: ONERR GOTO
      1110
1100 PRINT D#"VERIFY"SF#: GOTO 1
      160
1110 E = PEEK (222): CALL FIX: IF
      E < > 6 THEN FLASH : PRINT
      : PRINT "ERROR "; E: NORMAL :
      GOTO 200
1120 PRINT D#"OPEN"SF#
1130 PRINT D#"WRITE"SF#
1140 PRINT MP#
1150 PRINT D#"CLOSE"
1160 ONERR GOTO 8000

```

Main routine to administer test.

```

2000 HOME : QF = 1: QL = QN: NQ = Q
      N: IF MODE < > 2 THEN 2030
2010 FOR J = 1 TO 3: FOR I = 1 TO
      NQ: R = RND (1) * NQ + 1: T =
      QZ(I): QZ(I) = QZ(R): QZ(R) =
      T: NEXT : NEXT : NQ = NUM: IF
      NQ > QN THEN NQ = QN
2020 QL = NQ
2030 IF MODE < > 3 THEN 2070
2040 QL = Q2: IF QL > QN THEN QL =
      QN
2050 QF = Q1: IF QF > QL THEN QF =
      QL
2060 NQ = QL - QF + 1
2070 TV = NQ: IF NOT VFLAG THEN
      TV = 0: FOR I = QF TO QL: Q =
      QZ(I): TV = TV + VAL ( MID#
      (Q$(Q),2,1)): NEXT
2080 FOR I = QF TO QL: Q = QZ(I):
      GOSUB 4000: GOSUB 5000: NEXT
      I
2090 HOME : VTAB 8: PRINT "PRESS
      'R' TO REVIEW SKIPPED QUEST

```

```

IONS, OR 'E' TO END THE TES
      T.": GOSUB 10010: IF X# = "E
      " THEN 2130
2100 IF X# < > "R" THEN 2090
2110 FOR I = QF TO QL: Q = QZ(I):
      IF SZ(Q) < 0 THEN GOSUB 40
      00: GOSUB 5000
2120 NEXT I: GOTO 2090
2130 IF SM < > 2 THEN 2180
2140 RF = 1: FOR I = QF TO QL: Q =
      QZ(I): GOSUB 4000
2150 IF SZ(Q) > 0 THEN PRINT "Y
      OU SCORED "SZ(Q) / 100" FOR
      YOUR CORRECT ANSWER": GOTO 2
      170
2160 PRINT "YOUR ANSWER WAS INCO
      RRECT;": GOSUB 5090
2170 NORMAL : PRINT : GOSUB 1000
      0: NEXT I: RF = 0
2180 TS = 0: FOR I = QF TO QL: Q =
      QZ(I): TS = TS + SZ(Q) * (SZ(
      Q) > 0): NEXT : TS = TS / 100
2190 HOME : IF SM < 4 THEN PRINT
      : PRINT NF#;" , YOUR SCORE IS
      "; TS;" OUT OF "; TV: PRINT "
      OR "; INT (.5 + 100 * TS / T
      V);" PERCENT."
2200 GOSUB 6000: PRINT : PRINT :
      GOTO 190

```

Routine to allow teacher to choose number of questions to be administered and to reset test options if desired.

```

3000 HOME : RESTORE : GOSUB 1100
      0: SM = IT
3010 GOSUB 11000: MODE = IT: IF M
      ODE = 1 THEN 3080
3020 IF MODE = 3 THEN 3050
3030 VTAB 9: HTAB 23: INPUT "/HO
      W MANY? "; X#: NUM = VAL (X#)
      : VTAB 11: IF NUM < 1 THEN 3
      030
3040 GOTO 3070
3050 VTAB 10: HTAB 24: INPUT "FR
      OM "; X#: Q1 = VAL (X#): IF Q
      1 < 1 THEN 3050
3060 VTAB 10: HTAB 33: INPUT "TO
      "; X#: Q2 = VAL (X#): IF Q2 <
      1 OR Q2 < Q1 THEN 3060
3070 PRINT
3080 GOSUB 11000: VFLAG = IT - 1
3090 GOSUB 11000: HYNT = 2 - IT: IF
      NOT HYNT THEN 3120
3100 GOSUB 11000: PFLAG = 0: IF I
      T = 1 THEN 3120
3110 PFLAG = 1: VTAB 22: HTAB 24:
      INPUT "/10THS (0-9): "; HP: VTAB
      24: HP = INT (HP + .5): IF H
      P > 9 OR HP < 0 THEN 3110

```

```

3120 VTAB 24: HTAB 1: PRINT "ALL
      OK? "; GET X#: IF X# = "N"
      THEN 3000
3130 IF X# < > "Y" THEN 3120
3140 CALL - 875: RETURN
      Routine to administer an individual test
      question.
4000 HOME : Q# = Q$(Q): GOSUB 900
      0
4010 INVERSE : PRINT "#"; I - QF +
      1; " OF "; NQ;" VAL="; V;" OF
      "; TV;" "; IF HS THEN VH =
      INT (10 * V * (10 - HP) + .
      5) / 100: PRINT "("; VH;" W/H
      INT)": GOTO 4030
4020 PRINT "(NO HINT)"
4030 NORMAL : PRINT : PRINT Q#: PRINT
      : IF T = 1 THEN PRINT "TRUE
      OR FALSE? "; PRINT
4040 IF T < > 2 THEN 4070
4050 FOR J = 1 TO N: JJ = A1 + J -
      1: PRINT J;" . "; A$(JJ): IF PEEK
      (37) > 18 THEN FLASH : PRINT
      " "; GOSUB 10010: NORMAL : PRINT
      B#;" "; B#;
4060 NEXT J: PRINT
4070 QV = V: IF RF THEN 4140
4080 INPUT "ANSWER (S TO SKIP) "
      ; A#: IF A# < > "H" THEN 411
      0
4090 PRINT : IF HS THEN QV = VH:
      PRINT H$(Q): PRINT : GOTO 4
      080
4100 PRINT "NO HINT AVAILABLE": PRINT
      : GOTO 4080
4110 IF A# = "S" OR A# = "" THEN
      4140
4120 IF T = 2 THEN X = VAL (A#)
      : IF X < 1 OR X > N THEN 408
      0
4130 IF T = 1 THEN A# = LEFT# (
      A$,1): IF A# < > "T" AND A#
      < > "F" THEN 4080
4140 RETURN
      Routine to score the response to an in-
      dividual question.
5000 SQ = 0: IF A# = "S" OR A# =
      "" THEN SQ = - QV: GOTO 516
      0
5010 IF T = 1 AND ((A# = "T" AND
      TF) OR (A# = "F" AND NOT TF
      )) THEN SQ = QV
5020 IF T = 2 AND X = C THEN SQ =
      QV
5030 IF T < > 3 THEN 5070
5040 FOR J = A1 TO A1 + N - 1: L =
      LEN (A$(J)): IF C > 0 AND C
      < = L THEN L = C
5050 IF LEFT# (A$(J),L) = LEFT#
      (A$,L) THEN SQ = QV

```

```

5060 NEXT J
5070 IF SM < > 1 THEN 5160
5080 PRINT : IF SQ THEN PRINT "
CORRECT": FOR Z = 1 TO 600: NEXT
Z: GOTO 5160
5090 PRINT "CORRECT IS "; INVERSE
: IF T = 1 THEN PRINT TF*(T
F)
5100 IF T = 2 THEN PRINT C
5110 IF T < > 3 THEN 5140
5120 FOR J = A1 TO A1 + N - 1: PRINT
A*(J); IF J < A1 + N - 1 THEN
NORMAL : PRINT " OR "; INVERSE
5130 NEXT J: PRINT
5140 IF RF THEN 5190
5150 NORMAL : PRINT : GOSUB 1000
0
5160 IF SZ(Q) = 0 THEN SZ(Q) = S
Q * 100 + .5: GOTO 5190
5170 XZ = ABS (SZ(Q)):XXZ = ABS
(SQ) * 100 + .5: IF XZ > XXZ
THEN XZ = XXZ
5180 SZ(Q) = SGN (SQ) * XZ
5190 RETURN
Save score data to disk.
6000 PRINT : PRINT : PRINT "NOW
RECORDING SCORE ON DISK": ONERR
GOTO 6090
6010 PRINT D*"APPEND"SF$
6020 PRINT D*"WRITE"SF$
6030 PRINT N$: PRINT TD$: PRINT
QF: PRINT QL: PRINT TV: PRINT
TS
6040 FOR I = 1 TO 100: PRINT QZ(
I): NEXT
6050 FOR I = 1 TO 100: PRINT SZ(
I): NEXT
6060 PRINT D*"CLOSE"
6070 ONERR GOTO 8000
6080 RETURN
6090 E = PEEK (222): CALL FIX: PRINT
: FLASH : PRINT "ERROR ";E: NORMAL
6100 PRINT : GOSUB 10000: GOTO 6
000
Read in test data from disk.
7000 PAF = 0:EF = 0: PRINT : ONERR
GOTO 7090
7010 PRINT D*"VERIFY"QF$
7020 PRINT D*"OPEN"QF$
7030 PRINT D*"READ"QF$
7040 INPUT Q$(0):QN = VAL ( MID$
(Q$(0),4,3)):AP = VAL ( LEFT$
(Q$(0),3)):PW$ = RIGHT$ (Q$
(0), LEN (Q$(0)) - 6)
7050 IF P$ < > PW$ THEN PW$ = "
":PAF = 1: GOTO 7120
7060 FOR I = 1 TO QN: INPUT Q$(I
): NEXT I

```

```

7070 FOR I = 1 TO AP: INPUT A*(I
): NEXT I
7080 FOR I = 1 TO QN: INPUT H*(I
): NEXT I: GOTO 7120
7090 EF = 1:PW$ = "": FLASH :E =
PEEK (222): IF E = 6 THEN PRINT
"TEST NOT ON THIS DISK": GOTO
7110
7100 PRINT "ERROR ";E
7110 NORMAL : CALL FIX
7120 PRINT D*"CLOSE": ONERR GOTO
8000
7130 RETURN
General error-handling routine.
8000 E = PEEK (222): CALL FIX: IF
E = 255 THEN RESUME
8010 INVERSE : PRINT : PRINT "ER
ROR ";E;" IN LINE "; PEEK (2
18) + PEEK (219) * 256: NORMAL
: PRINT "PROGRAM TERMINATED"
: END
Separate encoded question/response.
9000 T = VAL ( LEFT$ (Q$,1)):V =
1: IF NOT VFLAG THEN V = VAL
( MID$ (Q$,2,1))
9010 IF T = 1 THEN TF = VAL ( MID$
(Q$,3,1)): GOTO 9030
9020 A1 = VAL ( MID$ (Q$,3,3)):N
= VAL ( MID$ (Q$,6,1)):C =
VAL ( MID$ (Q$,7,1))
9030 X = 9 - 4 * (T = 1):HS = VAL
( MID$ (Q$,X - 1,1)) * HYNT:
IF NOT PFLAG THEN HP = VAL
( MID$ (Q$,X,1))
9040 Q$ = RIGHT$ (Q$, LEN (Q$) -
X)
9050 RETURN
Single key input routine. Enter at line
10000 for optional pause message.
10000 HTAB 5: PRINT BR$;" PRESS
RETURN TO CONTINUE ";BR$;
10010 POKE CLR,0: WAIT KEY,PRESS
: GET X$: RETURN
Read menu headings and options.
11000 READ N$: READ N: FOR I = 1
TO N: READ N$(I): NEXT I
Display menu heading.
11010 INVERSE : HTAB 4: PRINT N$
: NORMAL :V = PEEK (37)
Display menu options.
11020 FOR I = 1 TO N
11030 PRINT TAB(4)I;". ";N$(I)
11040 POKE S, PEEK (S)
11050 NEXT I
Display menu prompt.
11060 VTAB V: HTAB 1: PRINT "=">"

```

# A harvest of savings from



## Apple Tree Electronics

### SOFTWARE

APPLE • ATARI • TRS80 • IBM

A full line of software for business, games and education **up to 35% off!**

MUSE	IKIS
VISICORP	STONEWARE
ON LINE	SYNERGISTIC
EDU-WARE	HAYDEN
HOWARD	AND MANY MORE

### HARDWARE

AMDEK • HAYES • MICROSOFT

	List	Our Price
32K RAM card	\$293.00	\$205.00
Video Term	\$345.00	\$279.00
Lazer Products	---	20% off ---

### DISKS

Maxell Box of 10, 5 1/4", SS-DD \$35.00  
Verbatim Box of 10, 5 1/4", SS-DD \$29.00

### MONITORS

LE MONITORS	List	Our Price
9" Green	\$189.00	\$159.00
12" Green	\$199.00	\$169.00
ZENITH		
12" Green	\$179.00	\$129.00

Plus a full line of AMDEK Monitors

### PRINTERS

PAPER TIGER	List	Our Price
460G	\$1,094.00	\$950.00
560G	\$1,394.00	\$1,250.00
EPSON		
MX 70	\$449.00	\$395.00
MX 80FT	\$745.00	\$595.00
MX 100FT	\$945.00	\$795.00

**CALL FOR THIS MONTHS SPECIAL!**

1-800-835-2246 EXT. 211

OR

702-452-5589

5130 East Charleston Blvd.  
Suite 5SS  
Las Vegas, Nevada 89122

VISA



Phone orders welcome. Mail orders may send charge card number (include expiration date), cashiers check, money order or personal check (allow ten business days for personal or company checks to clear). Add \$3.00 for shipping, handling and insurance. Nevada residents add 5.75% sales tax. Please include phone number. All equipment is in factory cartons with manufacturers warranty. Equipment subject to price change and availability. Call or write for price list.

```

;: POKE CLR,0
11070 GOSUB 10010:IT = VAL (X*)

11080 POKE S, PEEK (S)
11090 IF IT < 1 OR IT > N THEN 1
1060
11100 VTAB V: HTAB 1: PRINT " "
;:VV = V + IT: VTAB VV: HTAB
1: PRINT "=>";
11110 GOSUB 10010:X = ASC (X*)
11120 IF X < > 13 THEN IT = VAL
(X*): VTAB VV: HTAB 1: PRINT
" ";: GOTO 11080
11130 VTAB V + N + 2: HTAB 1: RETURN

Set Master password. POKE in stack fix
routine.
15000 X* = "104,168,104,166,223,1
54,072,152,072,096"
15010 FIX = 768: FOR I = 0 TO 9:J
= VAL ( MID$ (X*,I * 4 + 1
,3)): POKE FIX + I,J: NEXT
15020 MP* = "PASSWORD"
15030 RETURN

Menu data
50000 DATA " SCORING FEEDBACK ",
4,SCORE EACH QUESTION IMMEDI
ATELY,REVIEW & SCORE QUESTIO
NS AT END,GIVE FINAL SCORE O
NLY,WITHHOLD SCORE
50010 DATA " QUESTIONS TO ADMINI
STER ",3,ALL,RANDOM SELECTIO
N,RANGE OF NUMBERS
50020 DATA " QUESTION VALUES ",2
,AS ENTERED,ALL THE SAME
50030 DATA " HINT OPTION ",2,AS
ENTERED,OFF
50040 DATA " HINT PENALTY ",2,AS
ENTERED,SET FIXED PENALTY

```

**APPLE™ SWAT TABLE FOR:  
CATS2 - DISK VERSION**

LINES	SWAT CODE	LENGTH
100 - 190	RW	526
200 - 1030	SA	407
1040 - 1150	WD	361
1160 - 2100	QM	434
2110 - 3010	IS	457
3020 - 3130	TS	397
3140 - 4100	ZS	427
4110 - 5070	BM	309
5080 - 5190	CN	295
6000 - 7000	CI	267
7010 - 7120	CI	340
7130 - 11000	TM	426
11010 - 11120	FH	250
11130 - 50040	EZ	483

```

SS SS
SS
SS APPLESOFT BASIC SS
SS 'CATS2 - TAPE CHANGES' SS
SS AUTHOR: JON R. VOSKUIL SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS

```

DELETE Line 100

DELETE Lines 1080-1160

Change or add lines as follows.

```

130 DIM Q$(100),A$(300),H$(100),N$(
9),TF$(1),QZ(100),SZ(100),NAM
$(20),SSZ(20)
160 GOSUB 15000:VFLAG = 0:PFLAG =
0:HYNT = 1:MODE = 1:SM = 1:X =
RND ( - PEEK (78) - 256 * PEEK
(79)):SN = 0
190 FOR I = 1 TO 100:QZ(I) = I: NEXT
:SN = SN + 1
200 VTAB 24: GOSUB 10000:X = ASC
(X*): IF X = 13 THEN 900
220 IF X < > 15 AND X < > 19 THEN
200
330 IF P* = MP* AND X = 15 THEN GOSUB
3000
335 IF P* = MP* AND X = 19 AND SN >
1 THEN GOSUB 6000
900 IF NOT LODED THEN GOSUB 7000

910 IF SN > 20 THEN HOME : INVERSE
: PRINT : PRINT : PRINT "20-ST
UDENT MAXIMUM REACHED": NORMAL
: GOTO 200
1030 NAM$(SN) = N$:NF* = LEFT$ (N$
,SP - 1)
1070 IF P* < > PW* THEN INVERSE
: PRINT BEL$: PRINT "INCORRECT
PASSWORD": NORMAL : GOTO 200
2180 TS = 0: FOR I = QF TO QL:Q = Q
Z(I):TS = TS + SZ(Q) * (SZ(Q) >
0):SZ(Q) = 0: NEXT :TS = TS /
100:SP = INT (.5 + 100 * TS /
TV):SSZ(SN) = SP
2190 HOME : IF SN < 4 THEN PRINT
: PRINT NF*:", YOUR SCORE IS "
;TS;" OUT OF ";TV: PRINT "OR "
;SP;" PERCENT."
2200 GOTO 190

Substitute the following for lines
6000-6100. Instead of saving the scores
to disk for later analysis, this routine
displays them on the screen.
6000 HOME :AV = 0: FOR I = 1 TO SN
- 1:AV = AV + SSZ(I): NEXT :A
V = INT (AV / (SN - 1) + .5)
6010 FOR I = 1 TO SN - 1: PRINT SPC(
I < 10);I;". ";NAM$(I);

```

```

6020 L = LEN ( STR$ (SSZ(I))); HTAB
38 - L: PRINT SSZ(I);" Z"
6030 NEXT I: IF SN > 20 THEN SN =
1
6040 PRINT : PRINT TAB( 9)"AVERAG
E SCORE = ";AV;" Z": IF SN = 1
THEN INVERSE : PRINT " PRES
SING RETURN ERASES THESE SCORE
S ";: NORMAL
6050 RETURN

Substitute the following for lines
7000-7130
7000 HOME : VTAB 6:X = FRE (0): INPUT
"POSITION TAPE AT BEGINNING OF
TEST, START PLAYING, THEN
PRESS 'RETURN'.";X*
7010 FLASH : PRINT : PRINT "LOADIN
G": NORMAL
7020 RECALL Q$: RECALL A$: RECALL
H$: CALL RD
7030 CALL FIX,Q$(0): CALL FIX,A$(0
): CALL FIX,H$(0)
7040 QN = VAL ( MID$ (Q$(0),4,3)):
AP = VAL ( LEFT$ (Q$(0),3)):P
W* = RIGHT$ (Q$(0), LEN (Q$(0
)) - 6):LODED = 1: RETURN

DELETE Lines 8000-8010
Substitute the following for lines
15000-15030. Instead of the ONERR fix,
these pokes set up the tape read and
write routines.
15000 WR = 768:RD = 800:FIX = 865:M
= WR
15010 X* = "160,111,162,000,132,060
,134,061,160,116,032,025,003,1
64,111,166,112,132,060,134,061
,164,115,166,116,132,062,134,0
63,076,205,254,160,024,162,000
,132,060,134,061,160,029,132,0
62,134,063,032,253,254,165,028
,056,229,024,133"
15020 GOSUB 15100
15030 X* = "008,165,029,229,025,133
,009,160,002,162,000,056,181,1
11,149,062,245,008,149,111,149
,060,232,136,208,242,165,024,0
56,229,111,133,006,165,025,229
,112,133,007,076,253,254,032,1
90,222,032,123,221,032,108,221
,160,002,165,155"
15040 GOSUB 15100
15050 X* = "024,113,155,133,008,200
,165,156,113,155,133,009,160,0
00,177,160,240,015,200,177,160
,056,229,006,145,160,200,177,1
60,229,007,145,160,169,003,024
,101,160,133,160,144,002,230,1
61,197,008,208,220,165,009,197
,161,208,214,096"
15060 GOSUB 15100:MP* = "PASSWORD"

```

```

: RETURN
15100 FOR I = 0 TO 54:J = VAL ( MID$
(X$,I * 4 + 1,3)): POKE M + I,
J: NEXT :M = M + 55: RETURN

```

**APPLE™ SWAT TABLE FOR:  
CATS2 - TAPE VERSION**

LINES	SWAT CODE	LENGTH
110 - 190	KZ	513
200 - 1000	GV	403
1010 - 2040	ZM	411
2050 - 2160	IT	464
2170 - 3070	BW	440
3080 - 4040	WD	434
4050 - 5010	HJ	390
5020 - 5130	YJ	288
5140 - 6050	QN	382
7000 - 10000	ZG	511
10010 - 11100	OO	246
11110 - 15030	OG	576
15040 - 50010	FN	519
50020 - 50040	XJ	142

```

SS SS
SS
SS APPLESOFT BASIC SS
SS 'CATS3' SS
SS AUTHOR: JON R. VOSKUIL SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS

```

Display title page and initialize program

```

100 D$ = CHR$(4): PRINT D$"NOMONC
,I,0"
110 TEXT : HOME : VTAB 8: PRINT TAB(
17)"C A T S": VTAB 12: PRINT TAB(
12)"BY JON R. VOSKUIL"
120 VTAB 18: INVERSE : HTAB 10: PRINT
"SCORE ANALYSIS MODULE": NORMAL
130 DIM Q%(100),S%(100),NAM$(100),
TD$(100),WR%(100),W%(100),TS%(
100)
140 KEY = - 16364:PRESS = 128:CLR =
- 16368:BR$ = CHR$(93) + CHR$(
91):S = - 16336:BEL$ = CHR$(
7)
150 GOSUB 15000
160 VTAB 23: FOR I = 1 TO 100:N%(I
) = 0:WR%(I) = 0: NEXT : GOSUB
1000
170 GOSUB 3000

```

Main options menu

```

180 RF = 0: HOME : VTAB 6: PRINT "R
)VIEW THE SAME SCORES": PRINT
: PRINT "NEW SCORES FROM DISK
": PRINT : PRINT "Q)UIT": GOSUB
10010

```

```

190 IF X$ = "R" THEN RF = 1: GOTO
160
200 IF X$ = "N" THEN HOME : GOTO
160
210 IF X$ = "Q" THEN GOSUB 9000
220 GOTO 180
Main line evaluation routines
1000 IF RF THEN 1200
1010 PRINT "INSERT TEST SCORE DISK
AND PRESS RETURN": GOSUB 1001
0:N = 0:AV = 0:HI = 0:LO = 100
:PF = 0: PRINT
1020 ONERR GOTO 1190
1030 HOME : PRINT D$"CATALOG"
1040 PRINT : INPUT "FILE NAME: S.
":F$:SF$ = "S." + F$
1050 PRINT : PRINT "MASTER PASSWOR
D: ":P$ = ""
1060 GOSUB 10010: IF X$ = CHR$(1
3) THEN 1080
1070 POKE S, PEEK (S): POKE S, PEEK
(S): PRINT "P$:P$ = P$ + X$: GOTO
1060
1080 HOME : VTAB 8: PRINT "READING
SCORES...": PRINT D$"VERIFY"S
F$
1085 PRINT D$"OPEN"SF$
1090 PRINT D$"READ"SF$
1100 INPUT MP$: IF MP$ < > P$ THEN
PRINT D$"CLOSE": INVERSE : PRINT
: PRINT BEL$"INVALID PASSWORD"
: NORMAL : PRINT : GOSUB 10000
: PRINT : HOME : GOTO 1030
1110 N = N + 1
1120 PRINT D$"READ"SF$
1130 INPUT NAM$(N),TD$(N),QF,QL,TV
,TS
1140 S% = 100 * TS / TV + .5:TS%(N)
= S%:AV = AV + S%
1150 IF S% > HI THEN HI = S%
1160 IF S% < LO THEN LO = S%
1170 PRINT D$"POSITION"SF$,R200"
1180 GOTO 1110
1185 PRINT D$"VERIFY"SF$
1190 PRINT D$"CLOSE":E = PEEK (22
2):ERL = PEEK (218) + PEEK (
219) * 256: POKE 216,0: CALL F
IX:N = N - 1:AV = INT (AV / N
+ .5)
1192 IF E = 6 THEN VTAB 10: PRINT
BEL$ + BEL$ + "FILE " + F$ +
" NOT ON THIS DISK!": VTAB 22
: GOSUB 10000: PRINT : GOTO 10
20
1194 IF E < > 5 THEN VTAB 22: PRINT
BEL$:BEL$: "ERROR :":E;" IN LIN
E ":ERL: GOSUB 10000: GOTO 100
0
1200 HOME : INVERSE : IF PF THEN GOSUB
5000: GOTO 1220

```

```

1210 PRINT SPC(39): HTAB 1
1220 PRINT N$: TESTS LO=:LO: "%
HI=:HI: "% AV=:AV: "%Z": NORMAL
: PRINT
1230 PRINT TAB(9)"STUDENT
DATE - PCT (ADJ)": PRINT "
-----"
-----": POKE 34,4
1240 FOR I = 1 TO N
1250 PRINT SPC( I < 10);I: ". :NA
M$(I);
1260 HTAB 22: PRINT TD$(I);
1270 S$ = STR$(TS%(I)): HTAB 34 -
LEN (S%): PRINT S%;
1280 IF HI > 0 THEN AS% = 100 * TS
%(I) / HI + .5:S$ = STR$(AS%
): HTAB 35: PRINT "(" : SPC(3 -
LEN (S%));S%;")";
1290 PRINT : IF I / 18 = INT (I /
18) THEN PRINT : GOSUB 10000:
PRINT : PRINT
1300 NEXT I: PRINT : PR# 0:PF = 0:
ID = 0: VTAB 24
1310 HTAB 1: PRINT BEL$;"R)VIEW I
NDIV P)RINT C)ONT N)EW Q)UIT"
: POKE 34,0: GOSUB 10010
1320 IF X$ = "R" THEN 1200
1330 IF X$ = "I" THEN CALL - 875
: HTAB 1: GOSUB 2000
1340 IF X$ = "N" THEN PRINT : PRINT
: GOTO 1000
1350 IF X$ = "Q" THEN GOSUB 9000
1360 IF X$ < > "P" THEN 1390
1370 PF = 1: IF ID THEN GOSUB 2050
: GOTO 1310
1380 GOTO 1200
1390 IF X$ < > "C" THEN 1310
1400 RETURN
Lines 2000-2160 are the routine to review
scores for an individual student
2000 POKE 34,23: HOME : INPUT "TES
T NUMBER: ":X$:X = VAL (X%): POKE
34,0: IF X < 1 OR X > N THEN I
D = 0: GOTO 2160
2010 PRINT D$"OPEN"SF$
2020 PRINT D$"POSITION"SF$,R"206 *
(X - 1) + 1
2030 GOSUB 4000
2040 PRINT D$"CLOSE"
Print out number of correct/incorrect
answers for an individual student
2050 HOME : INVERSE : IF PF THEN GOSUB
5000: GOTO 2500
Screen print version
2060 PRINT X: ": ":N$: TAB(26)TD$:
TAB(36)TS%(X): "%": NORMAL
2070 VTAB 2: HTAB 1: PRINT "Q# SCR
Q# SCR Q# SCR Q# S
CR": PRINT "-- -- -- -- --
-- -- -- -- --"

```

```

2080 FOR I = 1 TO NQ
2090 C = INT ((I - 1) / 20):R = 3 +
      I - 20 * C
2100 VTAB R: HTAB C * 8 + 1
2110 SS = SZ(QZ(I)) / 100: PRINT SPC(
      QZ(I) < 10);QZ(I); SPC( 1 + (S
      S < 1 AND SS > 0));
2120 IF SS > = 0 THEN PRINT SS;:
      GOTO 2140
2130 PRINT "S";
2140 NEXT I: PRINT : PRINT
2150 ID = 1: PR# 0
2160 PF = 0: VTAB 24: RETURN

```

Printer version of above routine

```

2500 PRINT X;": ";N%; TAB( 26)TD%;
      TAB( 36)TSZ(X);"Z"
2510 PRINT "": PRINT "Q# SCR Q# S
      CR Q# SCR Q# SCR Q# SCR Q#
      SCR Q# SCR Q# SCR Q# SCR
      Q# SCR": PRINT "----
      -----
      ----"
2520 FOR J = 1 TO NQ
2530 SS = SZ(QZ(J)) / 100: IF SS <
      0 THEN SS% = " S": GOTO 2550

```

```

2540 SS% = " " + STR%(SS):SS% =
      RIGHT%(SS%,4)
2550 Q% = " " + STR%(QZ(J)):Q% =
      RIGHT%(Q%,2)
2560 PR% = PR% + Q% + SS% + " ": IF
      LEN (PR%) > 75 THEN PRINT PR
      %:PR% = "": PRINT ""
2570 NEXT J: PRINT PR%
2580 PF = 0:ID = 1: PR# 0: VTAB 24:
      RETURN

```

Print out total number of correct/incorrect answers for all students

```

3000 HOME : INVERSE : PRINT "PERCE
      NT OF WRONG ANSWERS, BY QUESTI
      ON #": NORMAL : PRINT "Q# # %W
      Q# # %W Q# # %W Q# # %W Q# #
      %W": PRINT "-----
      -----"
3010 INVERSE : FOR C = 0 TO 4: FOR
      R = 1 TO 20:NN = R + 20 * C:N%
      = RIGHT%( " " + STR%(NN),2
      ): VTAB R + 3: HTAB C * 8 + 1:
      PRINT N%;: NEXT : NEXT : PRINT
      : NORMAL : ONERR GOTO 3090
3030 PRINT D*"OPEN"SF%: PRINT D*"R
      EAD"SF%: INPUT MP%
3040 EF = 0: GOSUB 4000: IF EF THEN
      3100
3050 FOR I = 1 TO NQ:Q = QZ(I):WRZ
      (Q) = WRZ(Q) + (SZ(QZ(I)) < =
      0):NZ(Q) = NZ(Q) + 1
3060 WZ = INT (100 * WRZ(Q) / NZ(Q)
      ) + .5): IF WZ = 100 THEN WZ =

```

```

99
3070 C = INT ((Q - 1) / 20):R = Q -
      20 * C + 3: VTAB R: HTAB C * 8
      + 3 + (NZ(Q) < 10): PRINT NZ(
      Q); SPC( 1 + (WZ < 10));WZ;: NEXT
      : VTAB 23: PRINT
3080 GOTO 3040
3090 PRINT D*"CLOSE": POKE 216,0: CALL
      FIX
3100 VTAB 24: HTAB 5: PRINT BEL%;"
      C)ONTINUE P)RINTOUT Q)UI
      T";: GOSUB 10010
3110 IF X% = "P" THEN GOSUB 3500
3120 IF X% = "Q" THEN GOSUB 9000
3130 IF X% < > "C" THEN 3100
3140 RETURN

```

Print out answer percentages

```

3500 HOME : GOSUB 5000
3510 PRINT " % OF INCORRECT ANSWER
      S, BY QUESTION #": PRINT "====
      =====
      =====": PRINT
3520 PRINT " QUESTION # # OF T
      ESTS % MISSED": PRINT " --
      -----
      ----"
3530 FOR I = 1 TO 100: IF NZ(I) >
      0 THEN WZ = INT (100 * WRZ(I)
      / NZ(I) + .5): PRINT TAB( 7 +
      (I < 10));I; TAB( 21 + (NZ(I) <
      10));NZ(I); TAB( 34 + (WZ < 10
      ));WZ

```

3540 NEXT : PRINT : PR# 0: RETURN  
Read in test score data from disk

```

4000 ONERR GOTO 4100
4010 PRINT D*"READ"SF%
4020 INPUT N%,TD%,QF,QL
4030 PRINT D*"POSITION"SF%,R"QF +
      1
4040 PRINT D*"READ"SF%
4050 NQ = QL - QF + 1: FOR I = 1 TO
      NQ: INPUT QZ(I): NEXT
4060 PRINT D*"POSITION"SF%,R"100 -
      QL
4070 PRINT D*"READ"SF%
4080 FOR I = 1 TO 100: INPUT SZ(I)
      : NEXT
4090 GOTO 4110
4100 PRINT D*"CLOSE": CALL FIX:EF =
      1
4110 POKE 216,0: RETURN

```

Verify that user really wants to quit

```

9000 CALL - 875: HTAB 1: PRINT "D
      O YOU WANT TO QUIT? ";: GET Z%
      : IF Z% < > "Y" THEN CALL -
      875: RETURN
9010 TEXT : HOME : END
Single-key input routine. Enter at line
10000 for optional pause message
10000 HTAB 5: PRINT BR%;" PRESS RE
      TURN TO CONTINUE ";BR%;
10010 POKE CLR,0: WAIT KEY,PRESS: GET
      X%; RETURN
Poke in stack-fix routine
15000 X% = "104,168,104,166,223,154
      ,072,152,072,096"
15010 FIX = 768: FOR I = 0 TO 9:J =
      VAL ( MID$( X%,I * 4 + 1,3)):
      POKE FIX + I,J: NEXT
15020 RETURN

```

### APPLE™ SWAT TABLE FOR: CATS3

LINES	SWAT CODE	LENGTH
100 - 210	PI	469
220 - 1090	VJ	353
1100 - 1192	JB	422
1194 - 1300	BD	469
1310 - 2010	LO	313
2020 - 2130	GV	362
2140 - 2580	SI	456
3000 - 3090	BY	504
3100 - 4010	ON	452
4020 - 9000	SI	371
9010 - 15020	HC	188

# ATARI®

```

SS SS
SS
SS ATARI BASIC SS
SS 'CATS2 - DISK VERSION' SS
SS AUTHOR: JON R. VOSKUIL SS
SS TRANSL: ALAN J. ZETT SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS

```

Display title page and initialize program.

```

100 N0=0:N1=1:N2=2:N3=3:N4=4:N5=5:N10=
10:N37=37:N38=38:N48=48:N84=84:N100=10
0:N200=200
105 N1000=1000:N4000=4000:N5000=5000:N
8000=8000:N8100=8100:N10000=10000:N11
00=11000

```

```
110 CRS=752:GRAPHICS NO:POKE CRS,N1:CL
OSE #N1:OPEN #N1,N4,N0,"K":TRAP N8000
115 POSITION 17,N5:? "C A T S"
120 POSITION 12,7:? "by Jan R. Voskuil
"
```

```
125 POSITION 7,9:? "Translation by Ala
n J. Zett"
```

```
130 POSITION 13,14:? "TESTING MODULE"
```

```
135 DIM P$(N10),MP$(N10),PW$(N10),TF$(
N10),TD$(8),F$(8),QF$(12),SF$(12),N$(2
0)
```

```
137 DIM NF$(N10),BEL$(N4),C$(N1),S$(N1
),I$(LN*N48)
```

```
140 DIM Q(N100),S(N100):AZ=INT(FRE(N0)
*0.3):Q=INT(AZ/(LN*N48+152)):H=Q:A=H*N
3
```

```
145 DIM Q$(Q*(LN*N48)),H$(H*N38),A$(A*
N38)
```

```
147 S$=CHR$(156):BEL$=CHR$(253):BEL$(N
2)=" "
```

```
150 TF$(N1,N5)="FALSE":TF$(6,N10)="TRU
E ":S=53279:GOSUB 16000
```

```
155 A$(N1,N1)=" ":A$(A*N38)=" ":A$(N2)
=A$(N1):Q$=A$:H$=A$
```

```
160 VFLAG=N0:PFLAG=N0:HYNT=N1:MODE=N1:
SM=N1:X=RND(N0)
```

```
170 TRAP 170:POSITION N4,19:POKE CRS,N
0:? "TODAYS DATE (MM/DD/YY) ";;INPUT T
D$
```

```
180 IF LEN(TD$)<>8 OR TD$(N3,N3)<>"/"
OR TD$(6,6)<>"/" THEN 170
```

```
190 FOR I=N1 TO N100:Q(I)=I:S(I)=N0:NE
XT I
```

At this point, the user may hit CTRL-Q to quit, CTRL-S to set test options, or RETURN to go on. User must know master password to set options.

```
200 TRAP N8000:POSITION 8,23:GOSUB N10
000:IF AZ=155 THEN GOTO N1000
```

```
210 IF AZ=17 THEN GRAPHICS NO:CLR:END
```

```
220 IF AZ<>15 THEN GOTO N200
```

```
300 GRAPHICS NO:POSITION N2,7:? "WASTE
R PASSWORD ";;P$=""
```

```
310 GET #N1,X:IF X=155 OR LEN(P$)=N10
THEN 330
```

```
320 POKE S,N0:POKE S,N1:POKE S,N2:? "*"
";P$(LEN(P$)+N1)=CHR$(X):GOTO 310
```

```
330 IF P$=MP$ THEN GOSUB 3000
```

```
340 GOTO N200
```

Entry section of routine to administer test. Get student's name, test name, and password.

```
1000 GRAPHICS NO:POSITION N2,N3:? "ENT
ER YOUR NAME (FIRST AND LAST),":? "AND
PRESS RETURN":?
```

```
1005 TRAP N1000:INPUT N$:L=LEN(N$)
```

```
1010 SP=N0:FOR I=L TO N1 STEP -N1:C$=N
$(I,I):IF C$="" THEN SP=I
```

```
1020 NEXT I:IF SP<N2 OR SP=L THEN GOTO
N1000
```

```
1030 TRAP N8000:NF$=N$(N1,SP-N1):? :?
"TEST NAME ";;INPUT F$:GOSUB 15000
```

```
1040 ? :? "PASSWORD ";;P$=""
1050 GET #N1,AZ:IF AZ=155 OR LEN(P$)=N
10 THEN 1070
```

```
1060 POKE S,N0:POKE S,N1:POKE S,N2:? "
*";P$(LEN(P$)+N1)=CHR$(AZ):GOTO 1050
```

```
1070 POKE CRS,N1:? :? :? "INSERT DISK
AND PRESS RETURN":GET #N1,AZ:? :GOSUB
7000
```

```
1075 IF PAF THEN ? "INCORRECT PASSWORD
";BEL$:GOTO N200
```

```
1080 IF EF THEN GOTO N200
```

```
1090 SF$=QF$:SF$(LEN(SF$))="S":TRAP 11
10
```

```
1100 OPEN #N2,N4,N0,SF$:GOTO 1150
```

```
1110 IF PEEK(195)<>170 THEN GOSUB 8100
:GOTO N200
```

```
1120 CLOSE #N2:OPEN #N2,8,N0,SF$
```

```
1140 ? #N2:MP$
```

```
1150 CLOSE #N2
```

```
1160 TRAP N8000
```

Main routine to administer test.

```
2000 GRAPHICS NO:QF=N1:QL=QN:NQ=QN:IF
MODE<>N2 THEN 2030
```

```
2010 FOR J=N1 TO N3:FOR I=N1 TO NQ:R=I
NT(RND(N0)*NQ)+N1:T=Q(I):Q(I)=Q(R):Q(R)
=T:NEXT I:NEXT J:NQ=NUM
```

```
2015 IF NQ=QN THEN NQ=QN
```

```
2020 QL=NQ
```

```
2030 IF MODE<>N3 THEN 2070
```

```
2040 QL=Q2:IF QL>QN THEN QL=QN
```

```
2050 QF=Q1:IF QF>QL THEN QF=QL
```

```
2060 NQ=QL-QF+N1
```

```
2070 TV=NQ:IF NOT VFLAG THEN TV=N0:FO
R I=QF TO QL:Q=Q(I):L=Q*LN*N48-N10:TV=
TV+VAL(Q$(L+N2,L+N2)):NEXT I
```

```
2080 FOR I=QF TO QL:Q=Q(I):GOSUB N4000
:GOSUB N5000:NEXT I
```

```
2090 GRAPHICS NO:POSITION N2,7:POKE CR
S,N1:? "PRESS 'R' TO REVIEW SKIPPED QU
ESTIONS,OR 'E' TO END TEST."
```

```
2095 GET #N1,AZ:IF AZ=69 THEN 2130
```

```
2100 IF AZ<>82 THEN 2090
```

```
2110 FOR I=QF TO QL:Q=Q(I):IF S(Q)<N0
THEN GOSUB N4000:GOSUB N5000
```

```
2120 NEXT I:GOTO 2090
```

```
2130 IF SM<>N2 THEN 2180
```

```
2140 RF=N1:FOR I=QF TO QL:Q=Q(I):GOSUB
N4000
```

```
2150 POSITION N2,PEEK(N84)-N1:IF S(Q)>
N0 THEN ? "YOU SCORED ";S(Q)/N100:? "F
OR A CORRECT ANSWER":GOTO 2170
```

```
2160 ? "YOUR ANSWER WAS INCORRECT;":G0
SUB 5090
```

```
2170 ? :GOSUB N10000:NEXT I:RF=N0
```

```
2180 TS=N0:FOR I=QF TO QL:Q=Q(I):TS=TS
+S(Q)*S(Q)>N0:NEXT I:TS=TS/N100
```

```
2190 GRAPHICS NO:IF SM<N4 THEN ? :? NF
```

# T.H.E. SMART TERMINAL<sup>®</sup>



TURN YOUR ATARI 400 OR 800 INTO A REAL SMART TERMINAL

Get up to date information from services like Dow Jones, Compuserve, The Source, and local timesharing computers.

Save the information on disk or cassette for editing or reviewing when you disconnect from the telephone line!

Send the edited information back to the timesharing system when you are ready.

**REDUCE YOUR CONNECT CHARGES BY READING AND WORKING OFF LINE!!**

- User Friendly
- Disk or Cassette Based
- Works with Hayes Smart Modem
- X-ON/X-OFF Protocol
- Runs in 16K
- Serial or Parallel Printers
- Menu or Command Driven
- Save Data on Cassette or Disk
- Upload/Download Atari 400 or 800
- Multiple files in memory

This package allows you to define, transmit and receive characters so you can send characters and control codes not found on the Atari keyboard and receive characters that the Atari can translate into something it understands.

**A POWERFUL COMMUNICATIONS PACKAGE AT A SUPER PRICE!**

T.H.E. MOST Sophisticated Communications Package Available for the Atari, 400 or 800 and it's available on Cassette, too!

**\$49.95** cassette or disk

### ORDERING INFORMATION

Call BINARY directly to place your order. Our order lines are open 24 hours per day, 7 days per week.

### Shipping and handling charges:

North America: Add \$2.50  
Outside N.A.: Add 10%  
Michigan Residents: Add 4% tax.

### Payment Methods:

VISA, Master Charge, AMEX, cash, certified check, personal check (allow for clearance), money order.

Look for Binary Software Products at your local computer store.

Dealer Inquiries invited

## BINARY<sup>™</sup>

COMPUTER SOFTWARE  
3237 Woodward Ave.  
Berkley, MI 48072  
(313) 548-0533

BINARY CORPORATION

```

%;", YOUR SCORE IS ";TS;" OUT OF ";TV:
? "OR ";INT(0.5+N100*TS/TV);" PERCENT"
2200 GOSUB 6000:?:?:GOTO 190
Routine to allow teacher to choose
number of questions to be administered
and to reset test options if desired.
3000 GRAPHICS NO:RESTORE :POKE CRS,N1:
GOSUB N11000:SM=IT
3010 GOSUB N11000:MODE=IT:IF MODE=N1 T
HEN 3080
3020 IF MODE=N3 THEN 3050
3030 POSITION 24,8:?" /HOW MANY";:INPU
T NUM:IF NUM<N1 THEN 3030
3040 ? :GOTO 3070
3050 POSITION 25,9:?"FROM";:INPUT Q1:
IF Q1<N1 THEN 3050
3060 POSITION 33,9:?"TO";:INPUT Q2:IF
Q2<N1 OR Q2<Q1 THEN 3060
3070 ?
3080 GOSUB N11000:VFLAG=IT-N1
3090 GOSUB N11000:HYNT=N2-IT:IF HYNT=N
0 THEN 3120
3100 GOSUB N11000:PFLAG=NO:IF IT=N1 TH
EN 3120
3110 PFLAG=N1:POSITION 25,21:?" /10THS
? (0-9)";:GET #N1,HP:HP=INT(HP-N48):IF
HP<N0 OR HP>9 THEN 3110
3120 POSITION N2,23:?"ALL OK?";:GET #
N1,AZ:IF AZ=78 THEN 3000
3130 IF AZ<>89 THEN 3120
3140 ? S%;:RETURN
Routine to administer an individual test
question.
4000 GRAPHICS NO:I%=Q*(Q*LN*N48-(LN*N4
8-N1),(Q*LN*N48)-N1):GOSUB 9000
4010 ? "#";I-QF*N1;?" OF ";NQ;" / VAL="
;V;" OF ";TV;" ";
4015 IF HS<>NO THEN VH=INT(N10*V*(N10-
HP)+0.5)/N100:?"( ";VH;" W/HINT)":GOTO
4030
4020 ? "(NO HINT)"
4030 ? :? I%;IF T=1 THEN ? "TRUE OR FA
LSE?":?
4040 ? :IF T<>N2 THEN 4070
4050 FOR J=N1 TO N:JJ=A1+J-N1:POSITION
N2,PEEK(N84)-N1:?" J";. ";A%(JJ*N38-N3
7,JJ*N38)
4055 IF PEEK(N84)>19 THEN ? "2";:GET #
N1,AZ:?" S%;
4060 NEXT J:?"
4070 QV=V:IF RF<>NO THEN 4140
4080 TRAP 4080:POSITION N2,PEEK(N84)-N
1:?" ANSWER (S TO SKIP) ";:INPUT I%
4085 IF I%<>"H" THEN 4110
4090 ? :IF HS<>NO THEN QV=VH:?" H%(Q*N3
8-N37,Q*N38):?" :GOTO 4080
4100 ? "NO HINT AVAILABLE":?" :GOTO 408
0
4110 IF I%="S" THEN 4140
4120 IF T=N2 THEN X=ASC(I%(N1,N1))-N48

```

```

:IF X<N1 OR X>N THEN 4080
4130 IF T=N1 THEN I%=I%(N1,N1):IF I%<>
"T" AND I%<>"F" THEN 4080
4140 TRAP N8000:RETURN
Routine to score the response to an in-
dividual question.
5000 POKE CRS,N1:SQ=NO:IF I%="S" THEN
SQ=-QV:GOTO 5160
5010 IF T=N1 AND ((I%="T" AND TF) OR (
I%="F" AND NOT TF)) THEN SQ=QV
5020 IF T=N2 AND X=C THEN SQ=QV
5030 IF T<>N3 THEN 5070
5040 AZ=LEN(I%)+N1:I%(AZ)=" ":I%(N48)=
" ":I%(AZ+N1)=I%(AZ):FOR J=A1 TO A1+N-
N1:L=C:IF C=NO THEN L=N38
5050 IF A%(J*N38-N37,J*N38-N38+L)=I%(N
1,L) THEN SQ=QV
5060 NEXT J
5070 IF SM<>N1 THEN 5160
5080 ? :IF SQ THEN ? "CORRECT! ";:GOSU
B 10000:GOTO 5160
5090 ? "CORRECT ANSWER IS ";:IF T=N1 T
HEN ? TF*(TF*N5+N1,TF*N5+N5)
5100 IF T=N2 THEN ? C
5110 IF T<>N3 THEN 5140
5120 FOR J=A1 TO A1+N-N1:?" A%(J*N38-N3
7,J*N38):POSITION N2,PEEK(N84)-N1:IF J
<A1+N-N1 THEN ? "OR ";
5130 NEXT J
5140 IF RF<>NO THEN 5190
5150 ? :GOSUB N10000
5160 IF S(Q)=NO THEN S(Q)=INT(SQ*N100+
0.5):GOTO 5190
5170 X=ABS(S(Q)):XX=ABS(SQ)*N100+0.5:I
F X>XX THEN X=XX
5180 S(Q)=SGN(SQ)*X
5190 RETURN
Save score data to disk.
6000 POKE CRS,N1:?" :? :?" "NOW RECORDIN
G SCORE ON DISK":TRAP 6090
6010 OPEN #N2,9,NO,SF%
6030 ? #N2;N%;? #N2;TD%;? #N2;QF:?" #N2
;QL:?" #N2;TV:?" #N2;TS
6040 FOR I=N1 TO N100:?" #N2;Q(I):NEXT
I
6050 FOR I=N1 TO N100:?" #N2;S(I):NEXT
I
6060 CLOSE #N2
6070 TRAP N8000
6080 RETURN
6090 GOSUB N8100:GOTO 6000
Read in test data from disk.
7000 PAF=NO:EF=NO:POKE CRS,N1:?"
7010 TRAP 7090
7020 OPEN #N2,N4,NO,QF%
7040 INPUT #N2;LN:INPUT #N2;I%;QV=VAL(
I%(N1,N3)):AF=VAL(I%(N4,6)):PW%=I%(7)
7050 IF P%<>PW% THEN PW%="":PAF=N1:GOT
O 7100

```

```

7060 FOR I=N1 TO QN:INPUT #N2;I%;Q%(I%
LN*N48-(LN*N48-N1))=I%;NEXT I
7070 FOR I=N1 TO AP:INPUT #N2;I%;A%(I%
N38-N37)=I%;NEXT I
7080 FOR I=N1 TO QN:INPUT #N2;I%;H%(I%
N38-N37)=I%;NEXT I:GOTO 7100
7090 EF=N1:PW%="":IF PEEK(195)=170 THE
N ? :?" FILE NOT FOUND";BEL%:GOSUB 811
0:GOTO 7100
7095 GOSUB N8100
7100 CLOSE #N2:RETURN
General error-handling routine.
8000 GOSUB N8100:POKE CRS,NO:?" :?" "PRO
GRAM TERMINATED...":TRAP 40000:END
8100 POKE CRS,N1:?" BEL%:?" "ERRDR ";PEE
K(195);" TRAPPED AT LINE ";PEEK(186)+P
EEK(187)*256;BEL%
8110 CLOSE #N2:FOR AZ=N1 TO N100*N3:NE
XT AZ:TRAP N8000:RETURN
Separate encoded question/response.
9000 L=LN*N48-N10:T=VAL(I%(L+N1,L+N1))
:V=N1:IF NOT VFLAG THEN V=VAL(I%(L+N2
,L+N2))
9010 IF T=N1 THEN TF=VAL(I%(L+N3,L+N3)
):GOTO 9030
9020 A1=VAL(I%(L+N3,L+N5)):N=VAL(I%(L+
6,L+6)):C=VAL(I%(L+7,L+7))
9030 X=9-N4*(T=N1):HS=VAL(I%(L+X-N1,L+
X-N1))*HYNT:IF NOT PFLAG THEN HP=VAL(
I%(L+X,L+X))
9040 I%=I%(LN*N48-(LN*N48-N1),(LN*N48)
-N10)
9050 RETURN
Single key input routine. Enter at line
10000 for optional pause message.
10000 POKE CRS,N1:?" PRESS RETURN TO C
ONTINUE";
10010 GET #N1,AZ:RETURN
Read menu headings and options
11000 READ I%,N:V=PEEK(N84):?" " ";I%
Display menu heading.
11010 FOR I=N1 TO N
Display menu options.
11020 READ I%
11030 ? " ";I;". ";I%
11040 POKE S,NO:POKE S,N1:POKE S,N2
11050 NEXT I
Display menu prompt.
11060 POSITION N2,V:?" "=";
11070 GET #N1,IT:IT=IT-N48
11080 POKE S,NO:POKE S,N1:POKE S,N2
11090 IF IT<N1 OR IT>N THEN 11060
11100 POSITION N2,V:?" " ";VV=V+IT:PO
SITION N2,VV:?" "=";
11110 GET #N1,X
11120 IF X<>155 THEN IT=X-N48:POSITION
N2,VV:?" " ";:GOTO 11080
11130 POSITION N2,V+N+2:RETURN

```

Set filename extensions.

```
15000 QF$="D:":QF$(N3,N10)=F$:QF$(LEN(QF$)+N1)=".Q":RETURN
```

Set master password.

```
16000 MP$="PASSWORD":RETURN
30000 DATA SCORING FEEDBACK ,4,SCORE
QUESTIONS IMMEDIATELY,SCORE & REVIEW A
T END
30005 DATA GIVE FINAL SCORE ONLY,WITHH
OLD TEST SCORES
30010 DATA QUESTIONS TO ADMINISTER ,3
,ALL,RANDOM SELECTION,RANGE OF NUMBERS
30020 DATA QUESTION VALUES ,2,AS ENTE
RED,ALL THE SAME
30030 DATA HINT OPTION ,2,AS ENTERED,
OFF
30040 DATA HINT PENALTY ,2,AS ENTERED
,SET FIXED PENALTY
```

**ATARI® SWAT TABLE FOR: CATS2**

LINES	SWAT CODE	LENGTH
1 - 135	TY	552
137 - 200	6Q	542
210 - 1030	6V	425
1040 - 1150	WZ	373
1160 - 2090	SA	413
2095 - 2200	XB	476
3000 - 3110	RW	405
3120 - 4060	SY	434
4070 - 5020	ZE	390
5030 - 5140	IM	365
5150 - 6070	JE	310
6080 - 7095	MT	383
7100 - 10000	DS	505
10010 - 11100	PD	223
11110 - 30040	UI	468

```
SS SS
SS
SS ATARI BASIC SS
SS 'CATS2 - TAPE CHANGES' SS
SS AUTHOR: JON R. VOSKUIL SS
SS TRANSL: ALAN J. ZETT SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS
```

DELETE Lines 1080-1160

```
135 DIM P$(N10),MP$(N10),PW$(N10),TF$(
N10),TD$(8),N$(20),SS(20),NAM$(N200+N2
00)
155 A$(N1,N1)=" ":A$(A*N38)=" ":A$(N2)
=A$(N1):Q$=A$:H$=A$:NAM$=A$
160 VFLAG=N0:PFLAG=N0:HYNT=N1:MODE=N1:
SM=N1:X=RND(N0):SN=N0
190 FOR I=N1 TO N100:Q(I)=I:NEXT I:SN=
SN+N1
200 TRAP N8000:POSITION 8,23:GOSUB N10
000:IF AZ=155 THEN GOTO N1000-N100
220 IF AZ<>15 AND AZ<>19 THEN GOTO N20
0
330 IF P$=MP$ AND AZ=15 THEN GOSUB 300
0
335 IF P$=MP$ AND AZ=19 AND SN>N1 THEN
GOSUB 6000
900 IF LODED=N0 THEN GOSUB N8000-N1000
910 IF SN>20 THEN GRAPHICS NO:?:?:?
"20-STUDENT MAXIMUM REACHED";BEL$:GOTO
N200
1030 TRAP N8000:NAM$(SN*20-19,SN*20)=N
$:NF$=N$(N1,SP-N1)
1070 IF P$=PW$ THEN 2000
1075 ? :? :? "INCORRECT PASSWORD";BEL$
:GOTO N200
2180 TS=N0:FOR I=QF TO QL:Q=Q(I):TS=TS
+S(Q)*(S(Q)>N0):S(Q)=N0:NEXT I:TS=TS/N
100:SP=INT(.5+N100*TS/TV):SS(SN)=SP
2190 GRAPHICS NO:IF SM<N4 THEN ? :? NF
$:", YOUR SCORE IS ";TS;" OUT OF ";TV:
? "DR ";SP;" PERCENT"
2200 GOTO 190
```

Substitute the following for lines 6000-6100

```
6000 GRAPHICS NO:POKE CRS,N1:AV=N0:FOR
I=N1 TO SN-N1:AV=AV+SS(I):NEXT I:AV=I
NT(AV/(SN-N1)+0.5)
6010 FOR I=N1 TO SN-N1
6030 POSITION N2+(I<N10),I-N1:?" I;" "
;NAM$(I*20-19,I*20);
6040 L=LEN(STR$(SS(I))):POSITION N38-L
-N5,I-N1:?" SS(I);" ?"
6050 NEXT I:IF SN>20 THEN SN=N1
6060 ? :? CHR$(127);" AVERAGE SCORE
=";AV;" ?"
6070 IF SN=N1 THEN ? " PRESSING RETURN
ERASES THESE SCORES "
6080 RETURN
6090 REM
```

Substitute the following for lines 7000-7130

```
7000 GRAPHICS NO:POSITION N2,N5:POKE C
RS,N1:?"POSITION TAPE TO BEGINNING OF
TEST":?"PRESS PLAY AND HIT RETURN"
7020 OPEN #N2,N4,N0,"C:"
7050 IF P$<>PW$ THEN PW$="":GOTO 7100
```

```
7080 FOR I=N1 TO QN:INPUT #N2;I$:H$(I$
N38-N37)=I$:NEXT I:LODED=N1:GOTO 7100
7090 PW$=""
```

DELETE Lines 8000-8010

Substitute the following for lines 15000-15030. Instead of the ONERR fix, these pokes set up the tape read and write routines.

15000 REM

**ATARI® SWAT TABLE FOR: CATS2 - TAPE VERSION**

LINES	SWAT CODE	LENGTH
1 - 135	NU	537
137 - 190	FM	508
200 - 1000	FA	463
1005 - 2015	LN	449
2020 - 2120	DW	380
2130 - 3030	CD	501
3040 - 4000	AF	418
4010 - 4090	HL	438
4100 - 5060	QQ	388
5070 - 5180	CU	343
5190 - 7010	ND	423
7020 - 8110	RE	437
9000 - 11030	WG	402
11040 - 16000	SG	244
30000 - 30040	CR	331

```
SS SS
SS
SS ATARI BASIC SS
SS 'CATS3' SS
SS AUTHOR: JON R. VOSKUIL SS
SS TRANSL: ALAN J. ZETT SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS
```

Display title page and initialize program

```
100 N0=0:N1=1:N2=2:N3=3:N4=4:N5=5:N7=7
:N9=9:N10=10:N100=100:CR5=752:OPEN #N1
,N4,N0,"K:":POKE 82,N0
110 GRAPHICS NO:POKE CRS,N1:POSITION 1
6,N5:?"C A T S"
112 POSITION 11,N7:?"by Jon R. Voskui
l"
114 POSITION 8,N9:?"Translation by RB
& AJZ"
120 POSITION N9,15:?"SCORE ANALYSIS M
ODULE"
130 DIM Q(N100),S(N100),NAM$(2000),TD$
```

```
(800),WR(N100),N(N100),TS(N100)
132 DIM BEL$(N1),F$(8),SF$(12),P$(10),
MP$(10),I$(20),N$(20)
140 BEL$=CHR$(253):S=53279
160 FOR I=N1 TO N100:N(I)=NO:WR(I)=NO:
NEXT I:POSITION NO,21:GOSUB 1000
170 GOSUB 3000
```

#### Main options menu

```
180 RF=NO:GRAPHICS NO:POKE CRS,N1:POSIT
TION NO,N5:? "R)review same scores":? :
? "N)ew scores from disk":? :? "Q)uit"
190 GET #N1,X:IF X=82 THEN RF=N1:GOTO
160
200 IF X=78 THEN GRAPHICS NO:GOTO 160
210 IF X=81 THEN GOSUB 9000
220 GOTO 180
```

#### Main line evaluation routines

```
1000 IF RF THEN 1200
1010 POKE CRS,N1:? " Insert score di
sk and type RETURN":GET #N1,X:N=NO:AV
=NO:HI=NO:LO=N100:PF=NO:GOSUB 5000
1020 TRAP 1190
1040 ? :? "Test Name ":INPUT F$:GOSUB
15000
1050 ? :? "Master Password: ":P$=""
1060 GET #N1,X:IF X=155 OR LEN(P$)=N10
THEN 1080
1070 POKE S,NO:POKE S,N1:POKE S,N2:PRI
NT "X":P$(LEN(P$)+N1)=CHR$(X):GOTO 10
60
1080 GRAPHICS NO:POKE CRS,N1:POSITION
NO,N7:? "Reading scores...":OPEN #N3,N
4,NO,SF$
1100 INPUT #N3,MP$:IF MP$<>P$ THEN CLD
SE #N3:? :? BEL$:"INVALID PASSWORD":?
:GOSUB 10000:? :GRAPHICS NO:GOTO 1020
1102 NAM$(2000)="":NAM$(1,1)=" ":NAM$
(2)=NAM$(1):TD$=NAM$
1110 N=N+N1
1130 INPUT #N3,I$:NAM$(N*20-19,N*20)=I
$:INPUT #N3,I$:TD$(N*8-N7,N*8)=I$:INPU
T #N3,QF:INPUT #N3,QL:INPUT #N3,TV
1140 INPUT #N3,TS:S=INT(N100*TS/TV+0.5
):TS(N)=S:AV=AV+S
1150 IF S>HI THEN HI=S
1160 IF S<LO THEN LO=S
1170 FOR AZ=N1 TO N100+N100:INPUT #N3;
I$:NEXT AZ
1180 GOTO 1110
1190 CLOSE #N3:? :E=PEEK(195):ERL=PEEK
(186)+PEEK(187)*256:TRAP 40000
1192 IF E=170 THEN ? BEL$:"FILE ":F$;
" NOT ON DISK":GOTO 1020
1194 IF E<>136 THEN ? BEL$:"ERROR ":E;
" IN LINE ":ERL:GOTO 1020
1196 N=N-1:AV=INT(AV/N+0.5)
1200 GRAPHICS NO:GOSUB 5000
```

```
1220 ? #N2;" ";N;" TESTS LO=";LO;"
% HI=";HI;"% AV=";AV;"%":? #N2
1230 ? #N2;" Student D
ate PCT ADJ":? #N2;" -----
-----"
```

```
1240 FOR I=N1 TO N
1250 IF I<N10 THEN ? #N2;" ";
1252 ? #N2;I;"-";NAM$(I*20-19,I*20);
1260 ? #N2;TD$(I*8-N7,I*8);
1270 FOR AZ=1 TO 4-LEN(STR$(TS(I))):?
#N2;" ";NEXT AZ:? #N2;TS(I);
1280 IF HI>0 THEN AS=INT(100*TS(I)/HI+
0.5):FOR AZ=1 TO 4-LEN(STR$(AS)):#N2
;" ";NEXT AZ:? #N2;AS;
1290 ? #N2;IF I/18=INT(I/18) AND PF=NO
THEN ? #N2:GOSUB 10000:? #N2:? #N2
1300 NEXT I:? #N2:PF=NO:ID=NO
1310 POSITION N1,23:POKE CRS,N1:? BEL$
;"R)review I)ndiv P)rint C)ont N)ew Q)u
it":GOSUB 10010
1320 IF X=82 THEN 1200
1330 IF X=73 THEN POSITION NO,22:? CHR
$(156);CHR$(156);:GOSUB 2000
1340 IF X=78 THEN ? :? :GOTO 1000
1350 IF X=81 THEN GOSUB 9000
1360 IF X<>80 THEN 1390
1370 PF=N1:IF ID>NO THEN GOSUB 2050:GO
TO 1310
1380 GOTO 1200
1390 IF X<>67 THEN 1310
1400 RETURN
```

Lines 2000-2160 are the routine to review scores for an individual student

```
2000 POKE CRS,NO:? "Test Number ":INP
UT A:IF A<N1 OR A>N THEN ID=NO:GOTO 21
60
2010 OPEN #N3,N4,NO,SF$
2020 FOR AZ=1 TO 206:(A-1)+1:INPUT #N3
;I$:NEXT AZ
2030 GOSUB 4000
2040 CLOSE #N3
```

Print out number of correct/incorrect answers for an individual student

```
2050 GOSUB 5000:IF PF THEN 2500
```

Screen print version

```
2060 GRAPHICS NO:? A;" ";N$;" ";P$;
" ";TS(A);%"
2070 ? :? "Q# SCR Q# SCR Q# SCR Q#
SCR Q# SCR":? "----
-----"
2080 FOR I=N1 TO N0
2090 C=INT((I-N1)/20):R=N3+I-20*C
2100 POSITION C*8,R:Q=Q(I)
2110 SS=S(Q)/N100:IF Q<N10 THEN ? " ";
2120 ? Q;" ";
2125 IF SS>=NO THEN ? "S";:GOTO 2140
2130 ? SS;
```

```
2140 NEXT I:? :?
2150 ID=N1
2160 PF=NO:POSITION NO,23:RETURN
```

Printer version of above routine

```
2500 ? #N2;A;" ";N$;" ";P$;" ";TS
(A);%"
2510 ? #N2:? #N2;"Q# SCR Q# SCR Q# S
CR Q# SCR Q# SCR":? #N2;"--
----
-----":I=NO
2520 FOR J=N1 TO N0
2530 SS=S(Q(J))/N100:IF Q(J)<N10 THEN
? #N2;" ";
2540 ? #N2;Q(J);" ";
2550 IF SS<NO THEN ? #N2;"S ";:GOTO 2
565
2560 ? #N2;SS;" ";
2565 IF (SS<=NO OR SS>=N1) AND I<27 TH
EN ? #N2;" ";
2570 I=I+8:IF I>35 THEN I=NO:? #N2
2580 NEXT J:? #N2:? #N2
2590 ID=N1
2600 PF=NO:GOSUB 5000:POSITION NO,23:R
ETURN
```

Print out total number of correct/incorrect answers for all students

```
3000 GRAPHICS 0:? "PERCENT OF WRONG AN
SWERS, BY QUESTION #":? "Q# # %W Q# #
%W Q# # %W Q# # %W"
3002 ? "-----
-----":POKE CRS,N1
3010 FOR C=0 TO 4:FOR R=1 TO 20:NN=R+2
0*C:POSITION C*8,R+2:GOSUB 16000:NEXT
R:NEXT C:? :TRAP 3090
3030 OPEN #N3,N4,NO,SF$:INPUT #N3,MP$
3040 EF=NO:GOSUB 4000:IF EF THEN 3100
3050 FOR I=N1 TO N0:Q=Q(I):WR(Q)=WR(Q)
+(S(Q)<=0):N(Q)=N(Q)+1
3060 W=INT(N100*WR(Q)/N(Q)+0.5):IF W=N
100 THEN W=W-N1
3070 C=INT((Q-N1)/20):R=N2+Q-20*C:POSIT
ION C*8+N2+(N(Q)<N10),R:? N(Q);" ";:I
F W<10 THEN ? " ";
3072 ? W;NEXT I:POSITION NO,22:?
3080 GOTO 3040
3090 CLOSE #N3:TRAP 40000
3100 POSITION N4,23:? BEL$;"C)ontinue
Printout Q)uit":GOSUB 10010
3110 IF X=80 THEN GOSUB 3500
3120 IF X=81 THEN GOSUB 9000
3130 IF X<>67 THEN 3100
3140 RETURN
```

Print out answer percentages

```
3500 GRAPHICS NO:PF=N1:GOSUB 5000
3510 ? #N2;" % OF INCORRECT ANSWERS, B
Y QUESTION #":? #N2;"=====
=====":? #N2
```

```

3520 ? #N2;" QUESTION # # OF TESTS
% MISSED":? #N2;"=====
=====":? #N2
3530 FOR I=N1 TO N100:IF N(I)=NO THEN
3550
3535 W=INT(N100*WR(I)/N(I)+0.5):? #N2;
" ";:IF I<N10 THEN ? #N2;" ";
3540 ? #N2;I;" ";:IF N(I)<
N10 THEN ? #N2;" ";
3542 ? #N2;N(I);" ";:IF W<N10
THEN ? #N2;" ";
3544 IF W<N100 THEN ? #N2;" ";
3546 ? #N2;W
3550 NEXT I
3560 ? #N2:PF=NO:GOSUB 5000:RETURN
Read in test score data from disk
4000 TRAP 4100
4020 INPUT #N3;N$:INPUT #N3;P$:INPUT #
N3;QF:INPUT #N3;QL
4030 FOR AZ=N1 TO QF+N1:INPUT #N3;I$:N
EXT AZ
4050 NQ=QL-QF+1:FOR I=N1 TO NQ:INPUT #
N3;X:Q(I)=X:NEXT I
4060 FOR AZ=N1 TO N100-QL:INPUT #N3;I$
:NEXT AZ
4080 FOR I=N1 TO N100:INPUT #N3;X:S(I)

```

```

=X:NEXT I
4090 GOTO 4110
4100 CLOSE #N3:EF=N1
4110 TRAP 40000:RETURN
Routine to activate printer
5000 CLOSE #N2:GRAPHICS NO:POSITION N2
,b
5010 IF PF THEN POKE CRS,N1:? "Press R
ETURN when printer is ready.":GOSUB 10
010:OPEN #N2,B,NO,"P":RETURN
5020 OPEN #N2,B,NO,"E":RETURN
Verify that user really wants to quit
9000 ? CHR$(156);" Do you want to qu
it? ";:GET #N1,X:IF X<>B9 THEN ? CHR$(
156);:RETURN
9010 GRAPHICS NO:END
Single-key input routine. Enter at line
10000 for optional pause message
10000 ? "Press RETURN to continue"
10010 POKE 764,255:GET #N1,X:RETURN
Set file name extensions
15000 SF$="D:";SF$(3)=F$:SF$(LEN(SF$)+
N1)=".S":RETURN

```

```

16000 IF NN=100 THEN NN=0
16010 I$=STR$(NN):IF NN<10 THEN ? "0";
16020 FOR AZ=1 TO LEN(I$):? CHR$(ASC(I
$(AZ,AZ))+128);:NEXT AZ:RETURN

```

### ATARI® SWAT TABLE FOR: CATS3

LINES	SWAT CODE	LENGTH
100 - 160	JB	532
170 - 1060	ZR	444
1070 - 1190	DL	557
1192 - 1280	FD	590
1290 - 1400	RP	404
2000 - 2110	GX	410
2120 - 2550	WD	334
2560 - 3050	UN	537
3060 - 3510	BD	464
3520 - 4030	JJ	406
4050 - 10000	UE	403
10010 - 16020	NN	187

## TRS-80®

```

SS SS
SS
SS TRS-80 BASIC SS
SS 'CATS2 - DISK VERSION' SS
SS AUTHOR: JON R. VOSKUIL SS
SS TRANS: ALAN J. ZETT SS
SS COPYRIGHT (C) 1982 SS
SS SS
SS SS SS SS SS SS SS SS SS SS

```

Display title page and initialize program.

```

100 CLS
110 PRINT@152,CHR$(23)"C A T S":PRINT@270,"BY JON R. VOSKUIL"
120 PRINT@388,"TRANSLATION BY ALAN J. ZETT":PRINT@592,"TESTING
MODULE"
130 CLEAR0:DIMQ$(100),A$(300),H$(100),QX(100),SZ(100),TF$(2):IFM
EM$.9>32767THENCLEAR32767ELSECLEARNEM$.9
135 A1=0:AP=0:AZ=0:C=0:E=0:EF=0:HP=0:HS=0:HY=0:I=0:IT=0:J=0:JJ=0
:L=0:MO=0:N=0:NQ=0:NU=0:PA=0:PF=0:Q=0:Q1=0:Q2=0:QF=0:QL=0:QN=0:Q
V=0:R=0:RF=0:SM=0:SP=0:SQ=0:T=0:TF=0:TS=0:TV=0:V=0:VF=0:VM=0:VV=
0:X=0:XX=0:XXZ=0:Z=0
140 A$="":B$="":C$="":F$="":MP$="":N$="":NF$="":P$="":PW$="":Q$=
"":QF$="":SF$="":TD$="":U$="":X$="":DIMQ$(100),A$(300),H$(100),Q
X(100),SZ(100),TF$(2)
145 IFPEEK(293)=73THENPRINTCHR$(21);:U$=CHR$(244)+CHR$(245)+CHR$
(246)ELSEU$=""
150 TF$(0)="FALSE":TF$(1)="TRUE":B$=CHR$(8):GOSUB15000
160 VFLAG=0:PFLAG=0:HYNT=1:MODE=1:SM=1:RANDOM:ONERRORGOTO8000
170 PRINT@776,"TODAY'S DATE MM/DD/YY":PRINT@802,,:INPUTTD$
180 IFLEN(TD$)<>8ORMID$(TD$,3,1)<>"/"ORMID$(TD$,6,1)<>"/"THEN170

```

```
190 FORI=1TO100:Q%(I)=I:S%(I)=0:NEXT
```

At this point, the user may hit CTRL-Q to quit, CTRL-S to set test options, or RETURN to go on. User must know master password to set options.

```

200 PRINT@966,,:GOSUB10000:X=ASC(X$):IFX=13THEN1000
210 IFX=17THENCLEAR50:CLS:PRINTCHR$(21);:END
220 IFX<>15THEN200
300 CLS:PRINT@512,"MASTER PASSWORD: ";:P$=""
310 GOSUB10010:IFX$=CHR$(13)THEN330
320 PRINT"*";:P$=P$+X$:GOTO310
330 IFP$=MP$THENGOSUB3000
340 GOTO200

```

Entry section of routine to administer test. Get student's name, test name, and password.

```

1000 CLS:PRINT@128,"PLEASE TYPE YOUR NAME (FIRST AND LAST), AND
PRESS <ENTER>":PRINT:INPUTN$:L=LEN(N$):IFL=0THEN1000
1010 SP=0:FORI=LTO1STEP-1:C$=MID$(N$,I,1):IFC$="" THENSP=I
1020 NEXTI:IFSP=0ORSP=LTHEN1000
1030 NF$=LEFT$(N$,SP-1):PRINT:INPUT"TEST NAME ";F$:QF$=F$+"/@"
1040 PRINT:PRINT"PASSWORD: ";:P$=""
1050 GOSUB10010:IFX$=CHR$(13)THEN1070
1060 PRINT"*";:P$=P$+X$:GOTO1050
1070 PRINT:PRINT:PRINT"INSERT TEST DISK AND PRESS <ENTER>":GOSUB
10010:GOSUB7000:IFPAFTHENPRINT"==== INCORRECT PASSWORD =====":GOT
0200
1080 IFEFTHEN200
1090 SF$=F$+"/S":ONERRORGOTO1110
1100 OPEN"I",1,SF$:GOTO1160
1110 CLOSE:E=ERR/2-1:IFE<>52THENRESUME8000ELSERESUME1120
1120 OPEN"D",1,SF$
1140 PRINT#1,MP$
1160 CLOSE:ONERRORGOTO8000

```

Main routine to administer test.

```

2000 CLS:QF=1:QL=QN:NQ=QN:IFMODE<>2THEN2030
2010 FORJ=1TO3:FORI=1TONQ:R=RND(NQ):T=QZ(I):QZ(I)=QZ(R):QZ(R)=T
:NEXT:NEXT:NQ=NUM:IFNQ>NTHENNQ=QN
2020 QL=NQ
2030 IFMODE<>3THEN2070
2040 QL=Q2:IFQL>QTHENQL=QN
2050 QF=Q1:IFQF>QLTHENQF=QL
2060 NQ=QL-QF+1
2070 TV=NQ:IFVFLAG=0THENTV=0:FORI=QFTOQL:Q=QZ(I):TV=TV+VAL(MID$(
Q$(Q),2,1)):NEXT
2080 FORI=QFTOQL:Q=QZ(I):GOSUB4000:GOSUB5000:NEXTI
2090 CLS:PRINT@512,"PRESS 'R' TO REVIEW SKIPPED QUESTIONS, OR 'E
' TO END THE TEST.":GOSUB10010:IFX$="E"THEN2130
2100 IFX$<>"R"THEN2090
2110 FORI=QFTOQL:Q=QZ(I):IFSZ(Q)<0THENGOSUB4000:GOSUB5000
2120 NEXTI:GOTO2090
2130 IFSM<>2THEN2180
2140 RF=1:FORI=QFTOQL:Q=QZ(I):GOSUB4000
2150 IFSZ(Q)>0THENPRINT"YOU SCORED"SZ(Q)/100"FOR YOUR CORRECT AN
SWER":GOTO2170
2160 PRINT"YOUR ANSWER WAS INCORRECT;":GOSUB5090
2170 PRINT:GOSUB10000:NEXTI:RF=0
2180 TS=0:FORI=QFTOQL:Q=QZ(I):TS=TS-SZ(Q)*(SZ(Q)>0):NEXT:TS=TS/1
00
2190 CLS:IFSM<4THENPRINT:PRINTNF$, "YOUR SCORE IS"TS"OUT OF"TV"D
R"INT(.5+100*TS/TV)"PERCENT":IFTS=TVTHENC$=STRING$(2,191):PRINT@
256,C$STRING$(43,143)C$:PRINTC$ "EXCELLENT! YOU HAVE MADE A PERF
ECT SCORE! "C$:PRINTSTRING$(47,143)
2200 GOSUB6000:PRINT:PRINT:GOTO190

```

Routine to allow teacher to choose number of questions to be administered and to reset test options if desired.

```

3000 CLS:RESTORE:V=0:GOSUB11000:SM=IT
3010 V=6:GOSUB11000:MODE=IT:IFMODE=1THEN3080
3020 IFMODE=3THEN3050
3030 PRINT@896,"HOW MANY?";INPUTX$:NUM=VAL(X$):IFNUM<1THEN3030EL
SEPRINT@537,"OF"NUM;
3040 GOTO3070
3050 PRINT@896,"FROM";INPUTX$:Q1=VAL(X$):IFQ1<1THEN3050
3060 PRINT@906,"TO";INPUTX$:Q2=VAL(X$):IFQ2<1ORQ2<Q1THEN3060ELS
EPRINT@584,CHR$(31)"QUESTIONS"Q1"TO"Q2;
3070 PRINT@896,CHR$(31);
3080 V=6.55:GOSUB11000:VFLAG=IT-1
3090 V=11:GOSUB11000:HYNT=2-IT:IFHYNT=0THEN3120
3100 V=11.55:GOSUB11000:PFLAG=0:IFIT=1THEN3120
3110 PFLAG=1:PRINT@960,"10THS? (0-9)";GOSUB10010:IFX$<"0"ORX$>
9"THEN3110ELSEHP=VAL(X$):PRINT@874,CHR$(31)HP"/ 10THS";
3120 PRINT@960,"ALL OK?";GOSUB10010:IFX$="N"THEN3000
3130 IFX$<"Y"THEN3120
3140 PRINT@960,CHR$(31);RETURN

```

Routine to administer an individual test question.

```

4000 CLS:Q$=Q$(Q):GOSUB9000
4010 PRINT"NO."I-QF+1;"OF"NQ" VAL ="V"OF"TV" ";IFHSTHENVH=I
NT(10*V*(10-HP)+.5)/100:PRINT("VH"V/HINT)":GOTO4030
4020 PRINT(" NO HINT )"
4030 PRINT:PRINTQ$:PRINT:IFT=1THENPRINT"TRUE OR FALSE?":PRINT
4040 IFT<>2THEN4070
4050 FORJ=1TON:JJ=A1+J-1:PRINTJ"- "A$(JJ):IFPEEK(16417)>62THENPR
INT"?":GOSUB10010:PRINTB$ "B$;
4060 NEXT:PRINT

```

```

4070 QV=V:IFRFTHEN4140ELSEPRINT
4080 PRINTCHR$(27);A$="":INPUT"ANSWER (S TO SKIP) ";A$:IFA$<>"H
"THEN4110
4090 PRINT:IFHSTHENQV=VH:PRINTH$(Q):PRINT:GOTO4080
4100 PRINT"NO HINT AVAILABLE":PRINT:GOTO4080
4110 IFA$="S"ORA$=""THEN4140
4120 IFT=2THENX=VAL(A$):IFX<1ORX>NTHEN4080
4130 IFT=1THENA$=LEFT$(A$,1):IFA$<>"T"ANDA$<>"F"THEN4080
4140 RETURN

```

Routine to score the response to an individual question.

```

5000 SQ=0:IFA$="S"ORA$=""THENSQ=-QV:GOTO5160
5010 IFT=1AND((A$="T"ANDTF)OR(A$="F"ANDTF=0))THENSQ=QV
5020 IFT=2ANDX=CTHENSQ=QV
5030 IFT<>3THEN5070
5040 FORJ=A1TOA1+N-1:L=LEN(A$(J)):IFC>0ANDC<=LTHENL=C
5050 IFLEFT$(A$(J),L)=LEFT$(A$,L)THENSQ=QV
5060 NEXTJ
5070 IFSM<>1THEN5160
5080 PRINT:IFSQTHENPRINT"CORRECT":FORZ=1TO400:NEXTZ:GOTO5160
5090 PRINT"CORRECT ANSWER IS ";IFT=1THENPRINTTF$(TF)
5100 IFT=2THENPRINTB$C
5110 IFT<>3THEN5140
5120 FORJ=A1TOA1+N-1:PRINTA$(J):IFJ<A1+N-1THENPRINT"OR ";
5130 NEXTJ:PRINT
5140 IFRFTHEN5190
5150 PRINT:GOSUB10000
5160 IFSZ(Q)=0THENSZ(Q)=SQ#100+.5:GOTO5190
5170 XZ=ABS(SZ(Q)):XXZ=ABS(SQ)#100+.5:IFXZ>XXZTHENXZ=XXZ
5180 SZ(Q)=SGN(SQ)*XZ
5190 RETURN

```

Save score data to disk.

```

6000 PRINT:PRINT:PRINT"NOW RECORDING SCORE ON DISK":ONERRORGOTO6
090
6010 OPEN"E",1,SF$
6030 PRINT#1,N$:PRINT#1,TD$:PRINT#1,QF$:PRINT#1,QL$:PRINT#1,TV:PRI
NT#1,TS
6040 FORI=1TO100:PRINT#1,QZ(I):NEXT
6050 FORI=1TO100:PRINT#1,SZ(I):NEXT
6060 CLOSE
6070 ONERRORGOTO8000
6080 RETURN
6090 CLOSE:GOSUB8100
6100 PRINT:GOSUB10000:GOTO6000

```

Read in test data from disk.

```

7000 PAF=0:EF=0:PRINT
7010 ONERRORGOTO7090
7020 OPEN"I",1,QF$
7040 INPUT#1,Q$(0):QN=VAL(MID$(Q$(0),4,3)):AP=VAL(LEFT$(Q$(0),3)
):PW$=RIGHT$(Q$(0),LEN(Q$(0))-6)
7050 IFP$<>PW$THENPW$="":PAF=1:GOTO7120
7060 FORI=1TOQN:INPUT#1,Q$(I):NEXTI
7070 FORI=1TOAP:INPUT#1,A$(I):NEXTI
7080 FORI=1TOQN:INPUT#1,H$(I):NEXTI:GOTO7120
7090 EF=1:PW$="":E=ERR/2-1:IFE=52THENPRINT"FILE 'F$' IS NOT ON
THIS DISK":RESUME7120ELSERESUME7100
7100 GOSUB8100
7120 CLOSE:ONERRORGOTO8000
7130 RETURN

```

General error-handling routine.

```
8000 GOSUBB100:PRINT:PRINT"PROGRAM TERMINATED..."CHR$(21):END
8100 E=ERR/2-1:PRINT:PRINT"==== ERROR"E"IN LINE"ERL"====":CLOSE:
FORAZ=1TO1300:NEXT:ONERRORGOTOB000:RETURN
```

Separate encoded question/response.

```
9000 T=VAL(LEFT$(Q$,1)):V=1:IFVFLAG=0THENV=VAL(MID$(Q$,2,1))
9010 IFT=1THENTF=VAL(MID$(Q$,3,1)):GOTO9030
9020 A1=VAL(MID$(Q$,3,3)):N=VAL(MID$(Q$,6,1)):C=VAL(MID$(Q$,7,1)
)
9030 X=9+4*(T=1):HS=VAL(MID$(Q$,X-1,1))*HYNT:IFPFLAG=0THENHP=VAL
(MID$(Q$,X,1))
9040 Q%=RIGHT$(Q$,LEN(Q$)-X)
9050 RETURN
```

Single key input routine. Enter at line 10000 for optional pause message.

```
10000 PRINT"PRESS <ENTER> TO CONTINUE";
10010 X%=INKEY$:IFX%=""THEN10010ELSERETURN
```

Read menu headings and options

```
11000 READN$:READN:FORI=1TON:READN$(I):NEXT
```

Display menu heading.

```
11010 PRINT@V%64+4,CHR$(143)"N%"CHR$(143);
```

Display menu options.

```
11020 FORI=1TON
11030 PRINT@ (V+I)%64+3,I"-N$(I);
11050 NEXT
```

Display menu prompt.

```
11060 PRINT@V%64,U%;
11070 GOSUB10010:IT=VAL(X%)
11090 IFIT<1ORIT>NTHEN11060
11100 PRINT@V%64," ";VV=V+IT:PRINT@VV%64,U%;
11110 GOSUB10010:X=ASC(X%)
11120 IFX<>13THENIT=VAL(X%):PRINT@VV%64," ";GOTO11090
11130 PRINT@ (V+N+2),;:RETURN
```

Set master password.

```
15000 MP%="PASSWORD":RETURN
```

Menu data

```
50000 DATA TEST SCORING FEEDBACK,4,SCORE IMMEDIATELY,SCORE & REV
IEW AT END,GIVE FINAL SCORE ONLY,WITHHOLD TEST SCORES
50010 DATA QUESTIONS TO ADMINISTER,3,ALL QUESTIONS,RANDOM SELECT
ION,RANGE OF NUMBERS
50020 DATA QUESTION VALUES,2,AS ENTERED,ALL THE SAME
50030 DATA HINT OPTION,2,AS ENTERED,TURNED OFF
50040 DATA HINT PENALTY,2,AS ENTERED,SET FIXED PENALTY
```

TRS-80® SWAT TABLE FOR:  
CATS2 - DISK VERSION

LINES	SWAT CODE	LENGTH
100 - 140	JE	563
145 - 320	PL	451
330 - 1090	EN	456
1100 - 2060	YS	302
2070 - 2180	DP	507
2190 - 3080	PN	515
3090 - 4050	TC	513
4060 - 5020	GS	351
5030 - 5140	RX	283
5150 - 6070	OX	298
6080 - 7090	SQ	346
7100 - 10000	NH	395
10010 - 11120	PI	288
11130 - 50040	HY	373

```
SS SS
SS
SS TRS-80 BASIC SS
SS 'CATS2 - TAPE CHANGES' SS
SS AUTHOR: JON R. VOSKUIL SS
SS TRANS: ALAN J. ZETT SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS
```

These are the changes to be made in CATS2 in order to use it with tape.

DELETE Lines 1080-1160

Change or add lines as follows.

```
130 CLEAR:DIMQ$(100),A$(300),H$(100),Q$(100),SZ(100),TF$(2),NAM
$(20),SS$(20):IFMEM$.9>32767THENCLEAR32767ELSECLEARNEM$.9
140 A$="":B$="":C$="":F$="":MP$="":N$="":NF$="":P$="":PW$="":Q$=
"":QF$="":SF$="":TD$="":U$="":X$="":DIMQ$(100),A$(300),H$(100),Q
$(100),SZ(100),TF$(2),NAM$(20),SS$(20)
160 VFLAG=0:PFLAG=0:HYNT=1:MODE=1:SM=1:SN=0:RANDOM:ONERRORGOTOB0
00
190 FORI=1TO100:Q$(I)=I:NEXT:SN=SN+1
200 PRINT@966,;:GOSUB10000:X=ASC(X%):IFX=13THEN900
220 IFX<>15ANDX<>19THEN200
330 IFP%=MP%ANDX=15THENGOSUB3000
335 IFP%=MP%ANDX=19ANDSN>1THENGOSUB6000
900 IFLODED=0THENGOSUB7000
910 IFSN>20THENCLS:PRINT@192,"20-STUDENT MAXIMUM REACHED":GOTO20
0
1000 CLS:PRINT@128,"PLEASE TYPE YOUR NAME (FIRST AND LAST), AND
PRESS <ENTER>":PRINT:INPUTN$:N%=LEFT$(N$,20):L=LEN(N%):IFL=0THEN
1000
1030 NAM$(SN)=N$:NF%=LEFT$(N$,SP-1)
1070 IFP%<>PW%THENPRINT:PRINT"==== INCORRECT PASSWORD ====":GOTO
200
2180 TS=0:FORI=0FTOQL:Q=Q$(I):TS=TS-SZ(Q)*(SZ(Q)>0):SZ(Q)=0:NEXT
:TS=TS/100:SP=INT(.5+100*TS/TV):SS$(SN)=SP
2190 CLS:IFSM<4THENPRINT:PRINTNF$, "YOUR SCORE IS"TS"OUT OF"TV"D
R"SP"PERCENT":IFTS=TVTHENC$=STRING$(2,191):PRINT@256,C$STRING$(4
3,143)C$:PRINTC$ "EXCELLENT! YOU HAVE MADE A PERFECT SCORE! "C$:
PRINTSTRING$(47,143)
2200 GOTO190
```

Substitute the following for lines 6000-6100. Instead of saving the scores to disk for later analysis, this routine displays them on the screen.

```

6000 CLS
6010 AV=0:FORI=1TOSN-1:AV=AV+SS(I):NEXT:AV=INT(AV/(SN-1)+.5)
6030 FORI=1TOSN-1:IFI<11THENPRINT@I#64,;
6040 IFI>10THENPRINT@I-10#64+34,;
6050 L=LEN(STR$(SS(I)):QF#=RIGHT$(STR$(I),2)+". "+NAM$(I)
6060 QF#=QF#+ " "+STRING$(27-L-LEN(QF#),45)+STR$(SS(I))+"%
6070 PRINTQF#;:NEXT:IFSN>20THENSN=1
6080 PRINT@768,"AVERAGE SCORE ="AV"%
6090 IFSN=1THENPRINT" * PRESSING <ENTER> ERASES THESE SCORES *"
6100 QF#="":RETURN

```

Substitute the following for lines 7000-7130

```

7000 CLS:LO=0
7010 PRINT@256,"POSITION TAPE TO BEGINNING TO TEST DATA":X=FRE(0)
):PRINT"THEN PRESS <ENTER> TO START LOADING...":GOSUB10010
7020 PRINT@512,"LOADING..."
7040 INPUT#-1,Q$(0):QN=VAL(MID$(Q$(0),4,3)):AP=VAL(LEFT$(Q$(0),3)
)):PW#=RIGHT$(Q$(0),LEN(Q$(0))-6)
7050 PRINT@512,"LOADING CONFIRMED..."
7060 FORI=1TOQN:INPUT#-1,Q$(I):GOSUB7090:NEXT
7070 FORI=1TOAP:INPUT#-1,A$(I):GOSUB7090:NEXT
7080 FORI=1TOQN:INPUT#-1,H$(I):GOSUB7090:NEXT:LODED=1:FORX=1TO500
0:NEXT:RETURN
7090 PRINT@512,CHR$(31)"LOADING ";:IFI=QNRI=APTHENLO=LO+1:PRINT
"COMPLETED FOR SECTION"LO"OF"3-(QN<AP)":GOTO7130
7100 IFI/2<>INT(I/2)THENPRINT"***":GOTO7130
7120 PRINT" * "
7130 RETURN

```

TRS-80® SWAT TABLE FOR:  
CATS2 - TAPE VERSION

LINES	SWAT CODE	LENGTH
100 - 140	AU	597
145 - 320	WB	461
330 - 1060	YH	417
1070 - 2100	NJ	474
2110 - 2200	NX	502
3000 - 3110	VG	491
3120 - 4080	CJ	428
4090 - 5050	LI	351
5060 - 5170	NW	292
5180 - 6100	GZ	359
7000 - 7130	MP	505
8000 - 11010	FD	452
11020 - 50000	QG	354
50010 - 50040	NN	220

```

SS SS
SS
SS TRS-80 BASIC SS
SS 'CATS3' SS
SS AUTHOR: JON R. VOSKUIL SS
SS TRANS: ALAN J. ZETT SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS

```

Display title page and initialize program

```

100 CLS
110 PRINT@152,CHR$(23)"C A T S":PRINT@270,"BY JON R. VOSKUIL"
120 PRINT@388,"TRANSLATION BY ALAN J. ZETT":PRINT@650,"SCORE ANALYSIS MODULE"

```

```

130 CLEAR0:DIMB$(100),SZ(100),NAM$(100),TD$(100),WRZ(100),NZ(100)
),TSZ(100):IFMEM$.9>32767THENCLEAR32767ELSECLEARMEM$.9
140 DIMB$(100),SZ(100),NAM$(100),TD$(100),WRZ(100),NZ(100),TSZ(100)
150 LP#=STRING$(55,32):GOSUB16000
160 PRINT@896,;:FORI=1TO100:NZ(I)=0:WRZ(I)=0:NEXT:GOSUB1000
170 GOSUB3000

```

Main options menu

```

180 RF=0:CLS:PRINT@256,"R)VIEW THE SAME SCORES":PRINT:PRINT"WE
W SCORES FROM DISK":PRINT:PRINT"Q)UIT":GOSUB10010
190 IFX#="R"THENRF=1:GOTO160
200 IFX#="N"THENCLS:GOTO160
210 IFX#="Q"THENGOSUB9000
220 GOTO180

```

Main line evaluation routines

```

1000 IFRFTHEN1200
1010 PRINT"INSERT TEST DISK, PRESS <ENTER>"
1015 GOSUB10010:IFX$(CHR$(13))THEN1015ELSEN=0:AV=0:HI=0:LO=0:PF=0
:GOSUB17000:PF=USR(PF):CLS
1020 ONERRORGOTO1190
1040 PRINT@256,"FILE NAME ";:INPUTF$:SF#="F#"/S"
1050 PRINT:PRINT"MASTER PASSWORD ";:P#=""
1060 GOSUB10010:IFX$(CHR$(13))THEN1080
1070 PRINT" * ";:P#="P#X#":GOTO1060
1080 CLS:PRINT@256,"READING SCORES...":OPEN"1",1,SF#
1100 INPUT#1,MP#:IFMP#<P#THENCLOSE:PRINT:PRINT"==== INVALID PAS
SWORD =====":PRINT:GOSUB10000:CLS:GOTO1040
1110 N=N+1
1130 INPUT#1,NAM$(N),TD$(N),QF,QL,TV,TS
1140 SZ=100*TS/TV+.5:TSZ(N)=SZ:AV=AV+SZ
1150 IFSX>HITHENHI=SZ
1160 IFSX<LOTHENLO=SZ
1170 FORRP=1TO200:INPUT#1,RP#:NEXT
1180 GOTO1110
1190 CLOSE:E=ERR/2-1:IFE=61THENN=N-1:AV=INT(AV/N+.5):RESUME1200E
LSECLS:IFE=52THENPRINT@128,"FILE 'F#' NOT ON THIS DISK":RESUME
1040ELSEPRINT@128,"ERROR"E"IN LINE"ERL:RESUME1040
1200 ONERRORGOTO0:CLS:IFPFTHENGOSUB5000
1220 PRINT"TESTS LO ="LO"% HI ="HI"% AV ="AV"%:PRINT"
"
1230 PRINTTAB(8)"STUDENT DATE PCT (ADJ)":PRINT" ---
-----"
1240 FORI=1TON
1250 PRINTRIGHT$(STR$(I),2)". "NAM$(I);
1260 PRINTTAB(22)TD$(I);
1270 S#=STR$(TSZ(I)):PRINTTAB(34-LEN(S#))S#;
1280 IFHI>0THENASZ=100*TSZ(I)/HI+.5:S#=RIGHT$(STR$(ASZ),3):PRINT
TAB(35)"("STRING$(3-LEN(S#),32)S#)";
1290 PRINT" ":IFI/10=INT(I/10)ANDPF=0THENPRINT:GOSUB10000:PRINT@
256,CHR$(31);
1300 NEXT:PRINT" ":ID=0:PF=0:GOSUB17000:PF=USR(PF)
1310 PRINT@960,"R)VIEW I)NDIV P)RINT C)ONT N)EW Q)UIT";:GOSUB10
010
1320 IFX#="R"THEN1200
1330 IFX#="I"THENGOSUB15000:GOSUB2000
1340 IFX#="N"THENCLS:PRINT@256,;:GOTO1000
1350 IFX#="Q"THENGOSUB9000
1360 IFX$(CHR$(13))P#THEN1390
1370 PF=1:IFIDTHENGOSUB2050:GOTO1310
1380 GOTO1200
1390 IFX$(CHR$(13))C#THEN1310
1400 RETURN

```



# GENERAL INFORMATION

## Concerning *SoftSide* line listings, *SWAT* & Magnetic Media



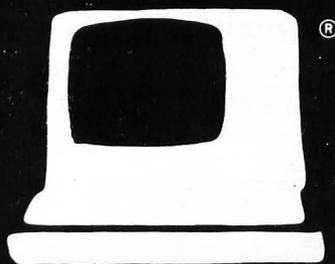
APPLE™/SIDE

page\_\_\_ 66



ATARI®/SIDE

page\_\_\_ 78



TRS-80®/SIDE

page\_\_\_ 84

Follow these procedures unless otherwise instructed by the documentation in the magazine. Back issues may differ in some details.

### SWAT TABLES

At the conclusion of each line listing of a *SoftSide* program, we include a *SWAT* (*Strategic Weapon Against Typos*) Table. *SWAT* was published in issue #30 of *SoftSide* and is available as a free reprint. Please send a self-addressed, stamped envelope to *SoftSide* Publications, Inc., Dept. *SWAT*, 6 South Street, Milford, NH 03055.

### APPLE™

*Disks* are in 13-sector format, created under DOS 3.2.1. If your system is set up for 16-sector disks (DOS 3.3), first boot your *BASICS* disk or *BRUN BOOT13* from the System Master Diskette, then insert the *SoftSide* disk. A cover/menu program will run automatically.

*Tapes* *LOAD* in the normal manner. Advance the tape to the beginning of the lead-in tone; stop the tape; insert the plug into the EAR jack; type *LOAD*; start the tape; and press *RETURN*. Side two of the tape is a duplicate of side one, unless one or more Integer *BASIC* programs are included, in which case side two contains the Integer programs.

### ATARI®

*Line Listings* use the following conventions in representing unprintable characters, unless otherwise noted:

Characters (including blank spaces) which are underlined should be typed in inverse video.

When graphics or control characters are to be included in a string (between quotation marks), it will be noted in a nearby *REMark*. In such cases, graphics characters are represented by the corresponding lower-case letter, and control characters are represented by the corresponding unshifted key symbol. For example: The lower-case letter *s* represents a control-down-arrow, entered by first pressing and releasing the *ESC* key, then holding down the *CTRL* key and pressing the = key. (See Appendix F, and the back cover, of the *ATARI® BASIC Reference Manual*.)

*The one exception to the above practice is that a clear-screen character (ESC CTRL-␣) is represented in listings by a right-hand brace, which looks like this: }*

*A shifted = is represented in the listings by a vertical line with a small gap in it: †*

*SWAT* — Before appending *SWAT* to a program in memory, the program to be *SWATed* must first be *LISTed* to disk or cassette (using *LIST "D:FILENAME"* for disk or *LIST "C:"* for tape). Next, turn the computer off, then on again, to clear the system and *ENTER* the program back into

memory (using *ENTER "D:filename"* for disk or *ENTER "C:"* for tape). Because of the unique method in which *ATARI® BASIC* stores variables in a program, the variable table must always be in the same order to produce accurate *SWAT* codes. *LISTing* and *ENTERing* the program is the only known way to rebuild the variable table in a specific order so that *SWAT* codes can match.

*Disks* do not contain *DOS.SYS* files, and are therefore not bootable by themselves. First boot a disk which contains any version of *DOS*, then insert the *SoftSide* disk and *RUN "D:COVER"* (*Adventure of the Month* — *RUN "D:INTRO"*).

*Tapes* *CLOAD* in the normal manner. If you have difficulty, try this procedure:

- (1) Type *POKE 54018,54* and press *RETURN*.
- (2) Turn up the volume on your TV.
- (3) Type *CLOAD* and press *RETURN* once.
- (4) Press the *PLAY* button and listen.
- (5) When you hear a steady lead-in tone, press *RETURN* again.

Side two of the tape is a duplicate of side one.

### TRS-80®

*Disks* are available in Model I or Model III format. They contain the *DOS PLUS* operating system, and a cover program which automatically runs upon booting. Back issues prior to May, 1982, are available only in Model I format, and may be converted using the *TRSDOS CONVERT* utility on a two-drive Model III. Older back issues (with Model I *TRSDOS*) require you to enter *BASIC* and then type *RUN "COVER"*.

*Tapes* *CLOAD* in the normal manner on Model I's, and at low speed (500 baud) on Model III's. The first program is a cover/menu program. Side two of the tape is a duplicate of side one.

### NOTES ABOUT MAGNETIC MEDIA

*SoftSide* disks and tapes are duplicated by reliable, professional duplication services; bad copies are very rare. However, the trip through the mail occasionally wreaks havoc with sensitive magnetic media. If, after a reasonable number of tries and a careful check and cleaning of your equipment, you are not able to load a program from a tape or disk, please return it to us with an exact description of the problem. If we cannot duplicate the problem on our systems, we will advise you when we send the replacement copy.

We use no copy-protection on our media. We urge you to make a backup copy of every disk or tape as soon as you receive it (and at the same time resist the urge to give copies to friends). Our replacement policy does not extend beyond 30 days.

**Aye! Come aboard for the fight of your life**

# SWASHBUCKLER



**Fantastic HI-RES sword fighting**

You're in a duel to the death against a black-hearted pirate crew. Look out! Here they come swinging and slashing with their pikes, swords, and knives.

Pirates to your right, pirates to your left . . . poison snakes, spiders, scorpions and killer rats! Hear your sword ring as you parry, thrust, lunge . . . run them through! If you're swordsman enough to win below decks, you go

topside, and the fight continues . . . with Blackbeard himself and the motley crew. What a game! What a test of skill against opponents who really attack and fight back! What a chance to become the swordsman of the year! Get your copy now - thrill to the most realistic HI-RES graphics ever!

**SWASHBUCKLER.** Only **\$34.95** for the Apple II\*. At computer stores or:

## **DATAMOST**

9748 Cozycroft Ave.  
Chatsworth, Ca. 91311  
(213) 709-1202

VISA/MASTERCHARGE accepted.  
\$1.00 shipping/handling charge.  
(California residents add 6% tax)

\*Apple II a trademark of Apple Computer, Inc.



# THE CLOZE TEST

APPLE™ DV

SoftSide

## The Classroom Teacher's Diagnostic Reading Test.

By Dr. William D. Hedges, Professor of Education, University of Florida.

The CLOZE Test (Classroom Teacher's Diagnostic Reading Test) is an educational utility requiring a 32K Apple™ with Applesoft and disk drive. It is included as the bonus program on this month's Apple™ Disk Version.

### I. THE CLOZE TEST: WHAT IT IS AND WHAT IT DOES.

#### A. What The CLOZE Test Is:

The CLOZE test is a fast, practical way to enable the teacher or social worker to determine whether a student or client is able to read selected materials with understanding. It removes every fifth word from a selection of prose, replacing the word with a blank; the student is asked to fill in the blanks with the missing word.

#### B. What It Does:

The CLOZE test counts the number of correct words and furnishes a diagnosis as follows:

1. The person can read the material independently and understand it. (Above 57% correct)
2. The person, with assistance, can understand the material. (44% to 57%)
3. The material is too difficult; it will be frustrating. (Less than 44% correct)

#### C. General Teacher Usage.

Stripped to the essence, the steps in doing the CLOZE test by hand are as follows:

1. Choose a passage or passages of reading material which, presumably, are representative of the whole.
2. Beginning with the second sentence, first word, remove every fifth word, replacing it with a blank.
3. Do not remove any words from either the first or the last sentence.
4. Endeavor to make sure you have at least 50 blanks to insure reliability of the test. This suggests you need at least 250 words in addition to first and last sentences.
5. Blanks should be of equal length. Sentences may be of variable length.
6. Initially ask students to read material silently in its entirety.
7. Then ask students to begin filling in each blank with the word they think belongs there. (Note: it has been found that the exact word only should be counted as correct; nothing is gained by using synonyms.
8. Teacher scores the paper.

9. Student score is percent correct, i.e. number of correct answers divided by the number of blanks.

10. Material is classified in category 1, 2 or 3 as indicated above.

### II. ENTERING AND USING THE MATERIAL ON THE MICRO COMPUTER

#### A. How The Teacher Can Enter Material:

This program is designed to take most of the drudgery out of the teacher's hands; for one thing, material need only be typed in once; from then on it is stored on a diskette. Five versions of the test are created for the student; more can be enabled by changing the number in the random number generator at line 1890.

*Step 1:* Place the program diskette in the disk drive.

*Step 2:* Type "Run Cloze Test" and press return key.

*Step 3:* When the Main Menu comes up, choose option 1 entitled "Teacher Input."

*Step 4:* Type in the sentences of the reading passage. Make sure you include the period, question mark, or exclamation mark at the end of each sentence.

*Step 5:* At the end of each sentence, press return. From this point on, presentation to students, scoring, etc. is done by the micro-computer.

*Step 6:* Of course, you will be asked to give the file a name and press return. It is suggested that you assign a name meaningful to you. Samples are: "The Scarlet Letter," or "Scot Foresman 4th Reader," etc..

#### B. How The Student Takes The CLOZE Test:

*Step 1:* Insert the diskette and turn on the console.

*Step 2:* Type "RUN CLOZE TEST".

*Step 3:* When the main menu comes up, choose option 2 entitled "STUDENT INPUT".

*Step 4:* At this point, the material is obtained from the file and presented at slow speed to the student. The student is asked to read the material

and try to understand it. *Note:* The material is slow the first time because this is when the processing is done on each sentence. However, the waiting time will not be very apparent because the student will be reading the material slowly anyway. *Step 5:* The student is asked if he/she wishes to reread the material; if not, then the test begins. The first sentence, which has no blanks, is presented. Immediately, the second sentence is also presented.

*Step 6:* The computer pauses and allows the student to enter the word he/she thinks is correct for the first blank. If there is no more than one blank, the sentence is represented and the student is asked to fill in that blank, and so on. The word the student entered is now contained in the sentence with parentheses around it.

*Step 7:* At the conclusion of the test, the student's score is presented.

*Step 8:* If the student or teacher wish, they may see both the student's answers and the correct answers.

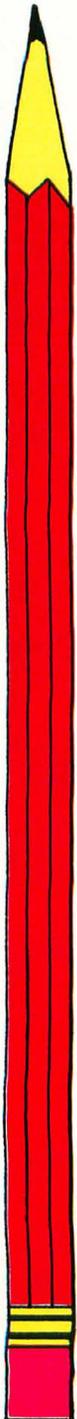
### III. PARAMETERS YOU MAY CHANGE

The dimension statements may have the numbers in them increased or decreased depending on the memory you have. Thus, if the teacher is entering short selections he or she may reduce the dimension numbers from the 52 sentences presently allowed.

### IV. TIPS AND PRECAUTIONS IN USING

The testing runs faster with short selections, i.e. ten to fifteen sentences; however, it is less reliable than 40-50 sentence selections. On the other hand, several short selection test scores are probably as reliable, if averaged, as one long one.

Now and then, the computer may pause for a few seconds; this is to clear out memory, so do not be alarmed by it. Usually, in 15-20 seconds, at most, it will commence once again.



## Wizardry

Reviewed by Hartley G. Lesser

Scenario 1: Proving Grounds of the Mad Overlord

Scenario 2: The Knight of Diamonds

\*For a review of another product available for the Apple™ see page 82.

from Sir-Tech Software, 6 Main St., Ogdensburg, NY 13669. System requirements: 48K Apple™ II or Apple™ II Plus with one or more disk drives, DOS 3.3. Suggested retail price: *Proving Grounds of the Mad Overlord* — \$49.45, *Knight of Diamonds* — \$34.95.

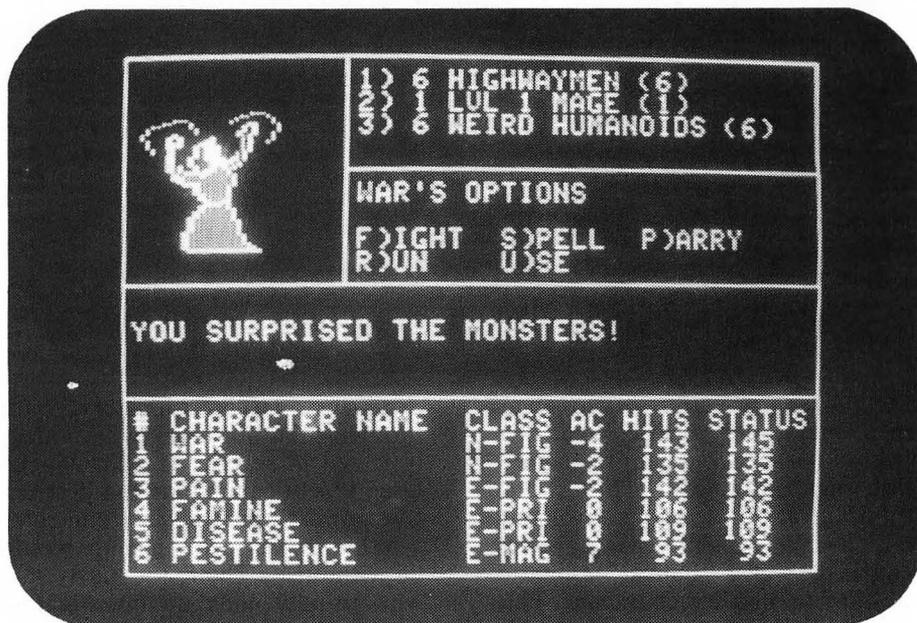
"I am in deep mourning. A companion...nay, more than mere companion...a brother...did pass away today, his life taken most cruelly. The others grieve, as do I, for Rumbler, human thief extraordinaire. There was an aura of luck about Rumbler, a second sense that oft delivered us from slyly concealed traps and fiendish devices. His accumulated wealth he gladly spent in the purchase of mystical potions which aided us all.

"Our first forays into the dungeon of the Mad Overlord were fraught with many dangers. Yet, these were mere hints of the terrors which awaited us while hunting for the *Staff of Gnilda* in our later quests.

"Nefarious creatures, possessed of powers the likes of which I pray never to witness again, seemingly awaited us at every turn of a corridor, and at the entry to every room. Never shall I forget the caustic sting of the first enemy-inflicted wound, nor Rumbler's smart efforts to staunch the flow of blood.

"I shall never gainsay Rumbler. Aye, it is my life I do owe Rumbler just as surely as do the others in our party. He will be sorely missed, that I can tell you. We shall never see his likes again."

It is of such makings that *Wizardry*, *A Game of Fantasy and Adventure*, calls to its players. Designed for the Apple™ with 48K and 3.3 DOS, the multi-faceted adventure uses the Apple™ Pascal 1.1 Runtime Operating System to extol its virtues onto your monitor's screen.



A typical display from *Wizardry*

More the pity other systems cannot take advantage of this marvelous software.

The authors, Andrew Greenberg and Robert Woodhead, deserve the Academy Award of Computer Gaming for successfully writing and coding both scenarios of *Wizardry*.

It is inadequate to call *Wizardry* a game. An "experience" is a much more suitable label. The player may create as many as twenty characters per disk, and may make as many scenario disks as he likes. Successful characters may also be transferred from one scenario to the next. Initially, characters may be in the following categories: Fighter, Mage, Priest, and Thief. As one gains in experience, characters can become Bishops, Samurai, Lords, or even Ninja. All abilities are taxed to the limit the deeper into the dungeon one adventures. The dangers of the second scenario, *Knight of Diamonds*, are so extreme that one may not even create new characters there. One may foray only with characters developed in the first

scenario, and the authors warn that even characters of thirteenth level may not be strong enough to withstand the onslaught of the powers fighting to maintain control over the dungeon and the *Staff of Gnilda*.

Each character class has a minimum requirement for Strength, I.Q., Piety, Vitality, Agility, and Luck. Strength affects melee combat, while I.Q. and Piety determine the ability to learn Mage and Priest spells, respectively. The score for Vitality indicates your character's vigor and his chances for successful resurrection, while Agility is the factor deciding who attacks first...your character, or his/her opponent...a most important facet of the game. Lastly, Luck can aid in many mysterious ways, so should not be overlooked any more than the other characteristics listed above.

The player must initially decide the race of his character (Human, Elf, Dwarf, Gnome, Hobbit), and his alignment (Good, Neutral, or Evil). Some character classes may only be Good, others Neutral, and



the player's selection of an alignment may totally restrict what profession he/she would enjoy playing. But, such is life, is it not?

Based on the decisions regarding race and alignment, the player is awarded several "bonus" points to allocate to his/her character. This point administration will decide your character's class. For example, to be a Priest, your character must have a Piety of at least eleven.

The Priest, whether Evil or Good (but never Neutral), is a powerful character, with the ability to Dispell (send them back where they came from) the "undead." They also possess healing spells, protection spells, light spells...all manner of help to an adventuring party.

Using the bonus points to cause Strength to reach eleven will allow your character to become a Fighter, able to use any weapon or any armor he or she can afford to purchase (or find within the deadly dungeon corridors). The Fighter will have high hit points, which measure the damage a character may sustain before dying. These particular adventurers are normally found leading a party through a dungeon, since only the first three members of the party may attack or be attacked.

Should I.Q. reach eleven, the Mage may be your character's profession. A Magic-User is somewhat limited in the use of weaponry and armor, but watch out for his/her spells, especially as the figure gains

experience levels. Monsters, once thought of as powerful, will suddenly lessen in severity if your Magic-User wisely commands the destructive powers under his/her control.

What adventuring group would be without its Thief, the hardy soul who usually ends up investigating locked chests for traps, and disarming them. This category requires at least an eleven in Agility, and he/she may never be of Good alignment. Though restricted regarding weaponry and armor, the Thief does manage to build more hit points than the Mage, which is good, especially if a trap he/she is disarming decides to trip.

The distribution of bonus points goes even further. Once you have decided on a character's profession, distribution between Vitality and Luck is important. The former will decide how likely it is that your character can be resurrected if he dies, and the latter will work with you, but in a strange and devious manner.

You will also be informed as to the amount of gold your character(s) has in his possession, and your Spell-Users (Priest, Mage, Lord and Samurai) will also carry spell books, within whose pages the secrets of power are written.

Should you, at any time, realize that your character(s) is not working out as you had hoped, there are several utility functions to aid in either completely deleting him,

rerolling his statistics, attempting to change the character's class (this will be possible as he gains in experience), or give him a new password (protects him from unauthorized use by another person). The latter should be used only in the most delicate of situations. For, if one should happen to forget the password of a character, you have to go through the dickens to clear him out of the created character file.

At any time, one may inspect any of the characters journeying through the dungeons. A complete statistical report is available for perusal, which is especially handy if you have forgotten the number of remaining spells for the Priest or Magic-User. Also revealed is the equipment currently being carried by your party, as well as items found. While in this mode, a number of other actions may also be undertaken.

Characters may freely trade possessions among themselves. Perhaps your Thief has found chain mail armor, but being unable to wear such fine protection, offers it to a first level Fighter who has only been able to afford leather armor. This would be a handsome trade, and worth a fair price between the two. You may also re-equip your character with items purchased or found.

Spell casting and using items may also be carried out in this mode. The latter would be for giving an antidote to a character pierced by a poison needle. Such actions tend to draw your party of characters close together.

Your game always begins and ends at the Castle, whether playing Scenario One or Scenario Two. It is from this fine edifice that your selections are made from the Master Menu offered on the screen.

The adventurers must assemble at Gilgamesh's Tavern. Here, any six of the as many as twenty characters you have created will meet and band together. A quick trip to Boltac's Trading Post by each character will equip him or her with whatever equipment may be purchased. The prices charged by Boltac's monopoly are, to say the least, exorbitant. One could save his/her treasure in order to buy that "something

special" gathering dust on the trader's shelves — like a +1 short sword...or the +2 staff, a tempting tidbit for any Mage...a +3 robe, or even a Ring of Healing.

One may also have an item found within the dungeon investigated by Boltac regarding its special properties, if any. Again, this is not inexpensive, but once the item is identified, the character may sell it...but, at half its original value.

Other areas accessible from the Castle are The Temple of Cant, an area where the dead members of your party are carried by the Castle guards if you bring them out of the dungeon. Should any wish to attempt a resurrection of their dead ally, here is where such is attempted. Resurrection is not a simple job, and the priests charge highly for their service. Still, if one of your characters can afford the required "donation," the odds are rather good that a resurrection will succeed. However, the older your character, the less likely it is that he/she will ever look upon the smil-

ing faces of comrades again.

The Adventurer's Inn, a friendly respite from the woes and wars of the dungeon, is probably the first place one would rush after leaving the Maze. It is here a character may rest, and have wounds healed at a cost determined by the type of accommodations the character selects. The stables, which are free, do nothing to aid in one's healing, although spells are recovered while resting, even here.

The more expensive the room, the faster the rate of healing. The number of hit points that may be regenerated is limited only by the amount of gold one has to pay the innkeeper for the service. The rooms are rated as Cots, Economy, Merchant Suites, and Royal Suites. The latter cost five hundred gold pieces per week, but heal at the rate of fifty hit points per week — just the place for a high-level Fighter who comes out of the Maze in tatters. It is here, too, that a character is checked to see if he has achieved his next level of experience. If he

has, he will be told of any increase (or decrease!) in his abilities, hit points, spells, and so on. In any event, he will be told how many experience points he will need to make the next level.

When it is time to proceed with the main task at hand, it's on to the Edge of Town, where one may choose to enter the Maze, return to the Castle (due to a sudden infusion of fear), or go to the Training Grounds.

The Training Grounds area is where all characters are initially created (in the first scenario), re-created, permanently deleted, inspected, equipped, and checked to see if they may change their class. For example, a Good Fighter might be capable of attaining Lordhood. This would entitle that character to not only retain the hit points and the warrior abilities of a Fighter, but also start to acquire priestly spells, a most powerful combination.

The same would be true of a Priest about to become a Bishop, at which time he/she would acquire

## BOOK/DISK COMBINATIONS

BECAUSE YOU DIDN'T BUY YOUR APPLE™  
TO PRACTICE YOUR TYPING

■ **Practical manuals that show you how to program your Apple™ for business, learning, and pleasure.**  
■ **Convenience disks that contain all the programs and subroutines in the books they accompany—error free and ready to run.**

**PLUS the Wiley expertise that has helped more than a million people learn how to program, use, and enjoy microcomputers.**

**Look for them at your favorite bookshop or computer store. Or, check the sets that interest you, fill in the ordering information, and mail us this ad.**



**JOHN WILEY & SONS, Inc.**

605 Third Avenue, New York, N. Y. 10158  
In Canada: 22 Worcester Road, Rexdale, Ontario M9W 1L1

For faster service call toll free: 800-526-5368.  
In New Jersey, call collect:  
(201) 797-7809. Order Code # 3-6769.  
VISA, MasterCard, American Express  
accepted on phone orders.

**APPLE™ BASIC: DATA FILE PROGRAMMING SET**

LeRoy Finkel & Jerald R. Brown

How to program and maintain data files for billings, catalogs and lists, numerical and statistical data, and much more. Includes one 5¼" disk for Apple II™. (Requires one 16 sector disk drive, 32K of memory) 1-89843-0 \$32.90

**GOLDEN DELICIOUS GAMES FOR THE APPLE™ COMPUTER SET**

Howard M. Franklin, Joanne Koltnow, & LeRoy Finkel

Step-by-step instructions for designing game programs that turn your Apple II™ into a home entertainment center—whether you're a novice, intermediate or advanced programmer. Includes two 5¼" disks for Apple II™. (Requires one 16 sector disk drive, 32K of memory) 1-89842-2 \$47.90

Please send the sets indicated for 15-DAY FREE EXAMINATION. (Restricted to the continental U.S. and Canada.)  
MAIL TO: JOHN WILEY & SONS, Inc.  
P.O. Box 092, Somerset, N.J.  
08873

**Payment enclosed**, plus sales tax. Wiley pays normal postage/handling. We normally ship within 10 days. If shipment cannot be made within 90 days, payment will be refunded.

**Bill me.**  **Bill firm or institution.**

NAME	AFFILIATION
ADDRESS	CITY/STATE/ZIP
<b>SIGN HERE</b>	092-3-6769

Prices subject to change without notice. Apple and Apple II are trademarks of Apple Computer, Inc.

Mage spells, and also have the ability to identify items found in the dungeon. Other attainable classes are Samurai and Ninja, most applicable to the fighters of your party.

The decision has been made. It's into the dungeon, for better or worse. The single letter commands are displayed for ease of use, although they will become second nature as you continue to play the game. All information concerning your party of daring adventurers is shown at the bottom of your monitor, and a three-dimensional, HI-RES display in the upper-left hand corner shows you where you are in the maze. If your Priest(s) has spells active, this will be revealed to the right of the 3-D display, with a blank message area beneath. At various times, messages containing hints or warnings will appear here.

As you move through the maze, the 3-D display moves with you. Doors, turns, and secret doors (if you have a special light-spell activated), will be revealed in this

amazing mini-screen. When the party has an encounter, the creatures they face will be pictured here, and when the fight is over, the treasure will be here to see.

For maximum enjoyment, the many secrets of these fantastic dungeons will not be revealed. It would behoove you to pay particular attention to the Tips for Beginners, found on page 34 of the well-written instruction manual. I'd suggest you learn to make maps if you don't know how. These are complex dungeons, and parties that do not keep accurate maps seldom make it out. Regular graph paper works very nicely for the purpose of maze mapping.

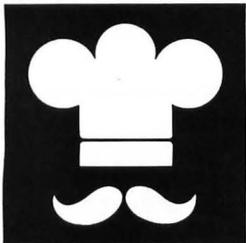
All in all, there are seven levels of Mage and Priest spells, and hundreds of possible creature combinations to confront your characters. The frightening Scenario One encounters seem almost mild when compared to those found in the *Knight of Diamonds* adventure. It will be no easy matter for you to find the one valuable item hidden by

the evil Wizard Werdna somewhere in the treasure rooms of the Mad Overlord Trebor's dungeon. Nor will it be simple to locate the *Staff of Gnilda* in Scenario Two, where the adversaries are stronger and more powerful than any previously encountered. From Dragon Zombies to the multiplying Fuzzballs, each adventurer should carefully evaluate his/her life's goals before commencing such treacherous undertakings.

Rest assured, the key items to be found in the dungeons, no matter the scenario, are protected by heinous entities, magical darkness, teleporting rooms, and difficult riddles. Your characters' experience levels must be developed to their peak if they are to delve into the lower regions of the dungeons.

I wish you well, and can but recommend this software with the highest of ratings. Never shall you be faced with a dull moment, and the character generation — maze mapping — combat — spell casting — why, it's all *WIZARDRY!* ☺

## C&H Best Sellers: Programs That Work!



### The Menu II

**\$39.95**

(+ \$2.00 shipping)

The MENU]I stores your favorite recipes in any of 6 "user-defined" categories. You "browse" through the alphabetized recipe list in these categories, choosing

your menu selections with a simple keystroke. You can write your menu for up to 2 weeks (42 meals); the rest is automatic! You can request a printed copy of your menu, your recipes (in any serving size up to 1,295 people) and an alphabetized shopping list of the ingredients needed to prepare the meals you've selected (a check list of misc. items can be added to the list with the touch of a button). The shopping list automatically combines "like" ingredients and adjusts each recipe for the number of people eating the meal. Recipes and menus can be changed, deleted, or added to at any time. Also included is a Special Meal/Party planner menu option, and 2 "user-defined" counters (calories, sodium intake, etc.) Since you won't be overbuying anymore, "THE MENU]I" may pay for itself on your very first shopping trip!

Operates with 1 or 2 disk drives (DOS 3.3). Requires 48K Apple, Disk Drive, and Printer. Written in APPLESOFT and MACHINE LANGUAGE.



### The Slide Show

**\$49.95**

(+ \$2.00 shipping)

Finally, high-res pics are more versatile than 35mm slides...and less expensive! The "slide" show allows you to use your high-res pics in a programmed se-

quence. Not another Graphics Package, but a versatile enhancement to all of the excellent Graphics Packages already available. Think of it as a standard slide show on your television with the advantage of "TV-like" special effect transitions between "slides." 20 different transitions (including: horizontal and vertical wipes, ripple effects, column wipes, etc.). Works with standard Apple pics. Includes the capability for producing a stand-alone "Run-Time" package. Hi-res pics (slides) load in approximately 6 seconds. Supports single or multiple disc drives. Free run, timed, keypress or game button slide advance. User friendly. NOT copy protected! Compatible with NTSC TV signal processing hardware. Prints slide/transition listing. Written in Applesoft/Machine Language. 48K, 1 or more disc drives, printer optional. (DOS 3.3)

#### PERFECT FOR:

- Educators
- Salespeople
- Businesspeople
- Executives
- Cable or closed circuit TV nets
- Free running store displays
- Lectures
- Presentations
- Exhibits
- Broadcast Television

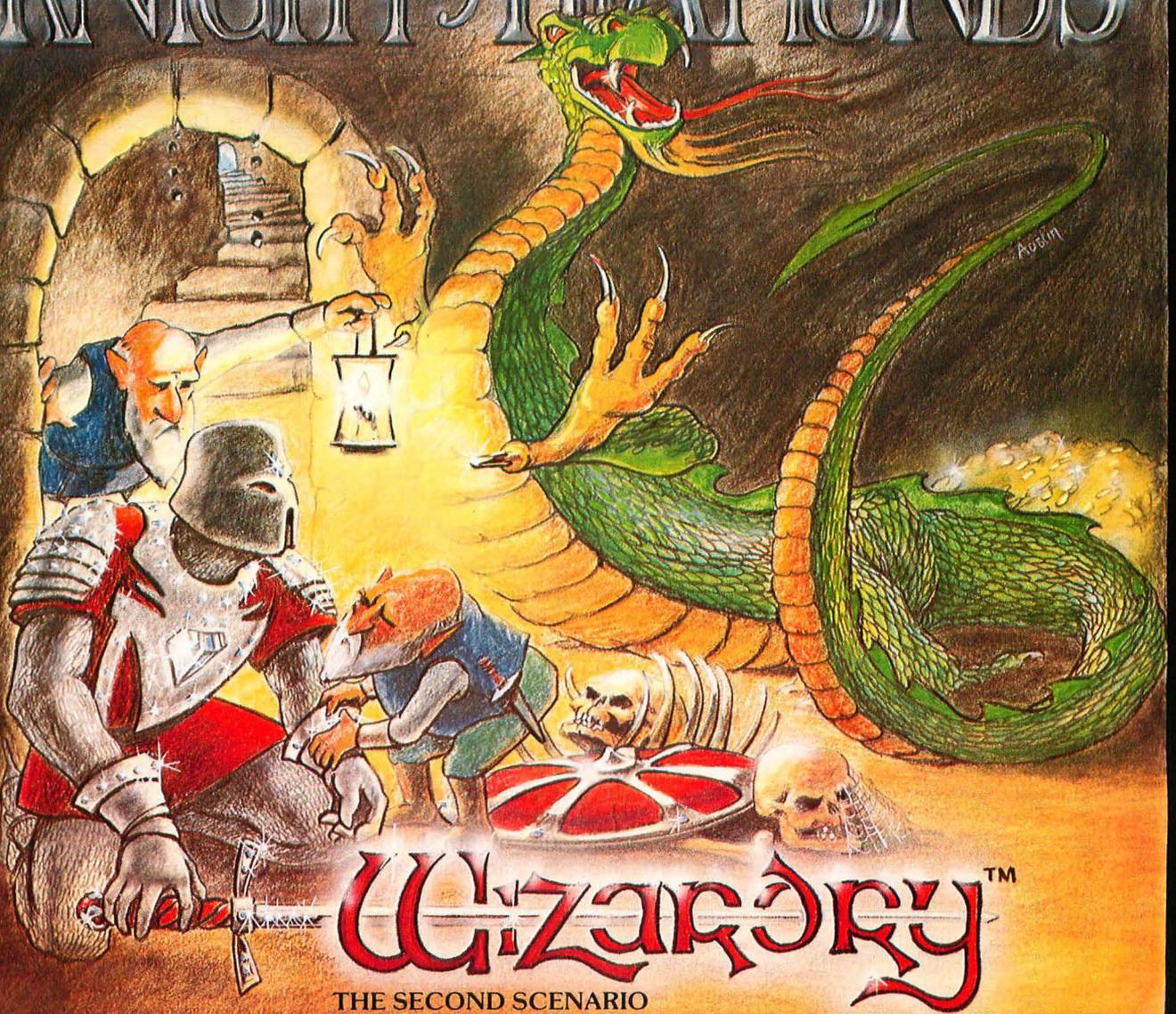
See your favorite APPLE dealer or order direct. Send check or money order to:

**C & H VIDEO**  
110 W. Caracas Ave.  
Hershey, PA 17033

DEALER INQUIRIES WELCOME  
PA Res. Add 6% sales tax  
**717-533-8480**  
Between 9am and 9pm



# KNIGHT OF DIAMONDS



## Wizardry™

### THE SECOND SCENARIO

**The Wizardry Adventure Continues!** Barbarians gather, preparing to sack the city of Llylgamyn. News that the mighty staff of Gnilda has been reclaimed by its earthgod creator has travelled fast. Protected for a thousand years by the power of the wondrous artifact, and now defenseless, Llylgamyn is doomed, unless you help!

What have people said about the first scenario?

*[Wizardry] has the potential to become a classic.*

David Lubar, *Creative Computing*

*The most eagerly awaited adventure . . .*

*Computer Merchandising*

*. . . perhaps the most advanced adventure program on the market.*

Forest Johnson, *The Space Gamer*

*The amount of detail is fantastic.*

Neil Shapiro, *Popular Mechanics*

*Knight of Diamonds*, a challenging scenario for experienced *Wizardry* players (13th level characters will barely survive!).

*Proving Grounds of the Mad Overlord* (shown at right) and *Knight of Diamonds* operate on **any** Apple Computer\* with at least 48K, DOS 3.3, and 1 disk drive.

*Proving Grounds of the Mad Overlord*—\$49.95

*Knight of Diamonds*—\$34.95

Join over ten thousand adventurers exploring the ultimate fantasy experience.



C.O.D.

Shipping \$2.00 in U.S.; \$5.00 outside U.S.

**NOW AVAILABLE AT YOUR LOCAL COMPUTER AND HOBBY STORE**

\*APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTER, INC.



**SOFTWARE, INC.**

6 MAIN STREET, OGDENSBURG, N.Y. 13669

(315) 393-6633

# APPLE DISKOURSE

## Part two of a series

By Cary W. Bradley

If you used the Apple™ DOS manual to learn how to use the Disk Operating System, chances are that you reached page 94, read the first paragraph, and moved on to the next chapter. In that paragraph, you were told that, if you were not familiar with Machine Language, you would do just as well to skip the material dealing with the RWTS (Read or Write a Track and Sector) subroutine. Don't let that intimidate you. Even though Machine Language programmers can use RWTS to do all kinds of tricks, you can use it, too, even if your programming knowledge is limited to BASIC. I wrote a BASIC program using the RWTS subroutine within a month after I bought my Apple™, long before I started learning 6502 Machine Language. I recommend that you read that section in conjunction with this article.

The RWTS subroutine is the most fundamental element of DOS. Almost every DOS command involves either reading from or writing to the disk. Reading and writing are done one sector at a time. The RWTS subroutine is a Machine Language routine within DOS which either reads a disk sector's 256 bytes into a 256-byte range of the Apple™'s memory or writes the contents of a 256-byte range of memory onto a disk sector.

Other routines within DOS, or possibly another program, must determine what the data means. RWTS simply moves a chunk of data to or from the disk, as specified by the calling program. (This same subroutine is also used to format a diskette, but we'll gloss over that fact for the time being.)

To use the RWTS subroutine from a BASIC program, you must provide three things; first, a 9-byte machine language routine, which has already been written for you. Second, you must provide 17 bytes of data which tell RWTS which slot

and drive you wish to use, to which track and sector you want to read or write, and where in the Apple™'s memory your input/output buffer is to begin. Finally, you must give RWTS a 4-byte Device Characteristics Table, which describes the disk hardware. All of this information is already written out for you on page 95 of the DOS manual. If you're working in BASIC, as we are in this series, all you have to do is convert the hexadecimal numbers listed on that page to their decimal equivalents and POKE them into an appropriate place in RAM.

Compare the data in line 2420 of the program accompanying this article to the data on page 95 of the DOS manual. Except for a few bytes relevant to the specific tasks of this program, the data is the same. Only the location at which they are stored has been changed. I have chosen memory address 768 (\$300) as the beginning location for this data in my program. This range of memory is handy to use for short Machine Language routines, because it is out of the way of everything which normally goes on during execution of a BASIC program. Once the proper data is loaded, you can execute the RWTS subroutine by CALLing the address at which you have stored the data.

I have entitled this month's program Disk Snooper, and when you run it, you'll see how apropos that name is. With it, you can snoop around your disks and see exactly which sectors each file occupies, and what is stored on each of them. The program is easy to use, with self-explanatory prompts, but a brief description of its usage is in order.

The program is designed to run solely on drive 1 plugged into slot 6, and assumes you are using DOS 3.3. With a few modifications, these restrictions can easily be changed. I've used RWTS similarly in DOS 3.2, but there are a few small differences. This program has not been tested under DOS 3.2. Each of the program's functions is accessed by pressing a single key. The keys which you may select are indicated by inverse characters in each of the program's menus.

The main menu shows the major program functions: Catalog, File Summary, and Read Sector. The

catalog is a BASIC adaptation of the machine language routine in DOS which runs in response to a DOS CATALOG command. It reads data from track 17, starting with sector 15, and displays the name of each active file on the disk. As you might expect, it is slower than its DOS counterpart, and it does not show all of the same information. It does, however, assign each file a number, which is a big help in the File Summary utility.

The File Summary shows where on the disk a particular file is located. When you select this option, you are asked for the file name or number. You will probably always want to enter the number, because it is so much easier than typing out the whole file name. For the masochists among you, the file name will also work. If you can't remember the number of the file you want, you can go directly to the Catalog utility by typing the letters CAT. Note that you don't have to view the entire catalog if it contains more files than will fit on the screen at one time. When the file you're interested in has been shown, you can hit the ESC key to abort the catalog and return to the File Summary utility.

If the file you select is an Applesoft or Integer BASIC program or a binary file, you will first be shown some special information about the file; its length in all cases, and, for a binary file, the address at which it will be BLOADED into RAM if you don't specify some other address in your BLOAD command. These numbers are shown in both decimal and hexadecimal. Hex numbers are in parentheses, and designated by the conventional prefix \$.

Next, you will see which disk sectors contain the data for that file. File sectors are of two types. One type contains a track/sector list, which is simply a listing of the track and sector numbers of sectors containing the file's data. The second type holds the actual data. Track/sector list sectors are shown at the upper part of the screen, and the data sectors are shown below, in columns, with a separate column for each track which holds a part of the file.

A disk file requires one track/sector list sector for every 122 data sec-

tors in the file. If your file is large enough to have more than one sector containing track/sector numbers, they will all be shown, and the one whose contents are currently displayed on the lower part of the screen will be shown in inverse. After all of the file's sectors have been shown, the total number of sectors in the file will be displayed. Both track/sector list sectors and data sectors are counted. This number is a true sector count, as opposed to the mod 255 sector counts which are shown in the standard DOS catalog.

The Read Sector utility allows you to read any sector on the disk and to see its contents. The 35 tracks are numbered from 0 to 34, and the 16 sectors, from 0 to 15. The displays of sector contents will show the sector most recently read by the RWTS subroutine, whether it was read by the Read Sector utility, or by the Catalog or File Summary utilities. If no sector has been read, or if an error occurred on the last attempted read, you will be unable to display the buffer's contents. The sector bytes are numbered from 0 to 255 (decimal), and the number of any byte can be determined by adding the headings (shown in inverse) of the row and column in which the byte appears.

Several different types of data are stored on a disk, but they all look the same to RWTS. Hence, there are three display modes from which to choose, so that you can meaningfully interpret the sector's contents. Some sectors will have some bytes which are meaningful as decimal or hex numbers, and others which are best interpreted as characters. Experiment with viewing a single sector in each of the display modes.

Because the display data is converted to character strings for the decimal and hex displays, you will have to wait a few seconds for the strings to be built when you select these types of displays. Only 136 of the 256 bytes in the sector can be shown on the screen at one time, so you must use the right and left arrow keys to scroll the display up and down. Scrolling in the direction opposite to that which your Apple™ is used to requires a short Machine Language routine, which is stored beginning at memory location 800

(\$320). This routine is POKed into RAM, and is represented by the data in the last two lines of the program.

Even though this program will read any part of a standard DOS 3.3 disk, notice that it does not include a single DOS command. That is the power of the RWTS subroutine. Although the DOS commands use it, if we know how to call it directly, we can bypass those commands to do all kinds of additional tasks. This program only scratches the surface of the RWTS subroutine's potential.

As I discussed in my first article, the program is designed to run on the smallest possible RAM configuration. It is written for a 24K Apple™, and will run on any machine with that much, or more, RAM, and Applesoft in ROM. If you want to make any modifications which would require additional RAM, you only need to change three lines of the existing program. Line 1520 must show your new value for HIMEM, which should be 257 less than the default value set when DOS is loaded. Line 1530 must initialize BF% to the starting address of the data buffer you will be using. This number is 1 greater than the new HIMEM value. If this is larger than 32767, you must use the two's complement form (subtract 65536). Finally, you must change the 18th and 19th values in line 2420 to the decimal equivalents of the low byte and high byte of the hexadecimal address of your data buffer (0 and 53, respectively, in this version of the program.)

You can use the DOS manual, and the program itself, to investigate exactly how each part of the program works. With a little experimentation, you can gain a lot of insight into the workings of DOS. Appendix C of the DOS manual will help you a great deal. We'll be doing much of this in upcoming articles, to see what is happening when we use the DOS commands. If any part of the program seems mysterious, be patient — we'll clear it all up as we explore the ins and outs of DOS in future installments.

Next time, we'll look at the DOS catalog, and how it keeps track of what's on your disks. We'll also introduce a handy utility program to help you manage files while you're

programming. Meanwhile, let me know, c/o *SoftSide*, of any DOS topics you'd like to see us tackle in this column.

## Variables:

AD: Decimal program length and starting address  
 AD\$: Hex equivalent of AD  
 B%: Bottom line of sector display  
 BA%: Base address of current screen line  
 BB%: Base address of bottom screen line  
 BF%: Starting address of data buffer  
 BS%: Number of sector in the buffer  
 BT%: Number of track in the buffer  
 C%: Track/sector display column number  
 C1%, C2%: Used in dec/hex conversion  
 CT%: File sector counter  
 C\$(\*): Menu choices  
 D: Decimal byte being converted to hex  
 D%: Display type indicator  
 E%: (390-410) Error flag  
 (1310-1340) Keyboard input  
 ER%: RWTS error flag address  
 F%: File counter  
 H\$: Hexadecimal character string  
 HI: Address of high byte  
 I, J, K: Index variables (misc. uses)  
 IN\$: Keyboard input string  
 L%: Screen line counter  
 LO: Address of low byte  
 L\$(\*): Display line strings  
 M\$: File data message  
 M%: Error message type (0-1)  
 M\$(\*): File Summary error messages  
 N\$: File name  
 N%: Number of file being sought  
 NS%: Next sector to be read  
 NT%: Next track to be read  
 R%: Display row (line) number  
 RW%: Starting address of RWTS calling routine  
 S%: Last track number displayed  
 SC%: Sector selection address  
 SR\$(\*): Sectors of track/sector list  
 T\$: File type  
 T%: Temporary variable (misc. uses)  
 TB%: Base address of top screen line  
 TK%: Track selection address  
 TR\$(\*): Tracks of track/sector list

```

SS SS
SS
SS APPLESOFT BASIC SS
SS 'DISK SNOOPER' SS
SS AUTHOR: CARY BRADLEY SS
SS COPYRIGHT (C) 1982 SS
SS
SS SS SS SS SS SS SS SS SS SS SS

```

1 GOTO 1500

Convert decimal byte to 2-byte hex string

```

10 C1% = INT (D / 16); C2% = D -
    C1% * 16
20 H% = CHR% (48 + C1% + 7 * (C1
    % > 9)) + CHR% (48 + C2% +
    7 * (C2% > 9)); RETURN

```

Build character strings for sector display lines

```

30 GOSUB 2340; FOR I = 0 TO 31:L
    $(I) = ""
40 FOR J = BF% + 8 * I TO BF% +
    7 + 8 * I
50 L$(I) = L$(I) + RIGHT$ (" "
    + STR$ ( PEEK (J)),4); NEXT
    J,I; RETURN
60 GOSUB 2340; FOR I = 0 TO 31:L
    $(I) = ""
70 FOR J = BF% + 8 * I TO BF% +
    7 + 8 * I
80 D = PEEK (J); GOSUB 10:L$(I) =
    L$(I) + " " + H%; NEXT J,I;
    RETURN

```

Catalog subroutine

```

100 HOME : POKE TK%,17; POKE SC%
    ,15; CALL RW%; GOSUB 2350
110 PRINT "CATALOG";; HTAB 25; PRINT
    "DISK VOLUME "; PEEK (791); POKE
    34,2; HOME
120 LZ = 0; F% = 0
130 FOR I = BF% + 11 TO BF% + 22
    1 STEP 35; TZ = PEEK (I); IF
    TZ = 255 OR TZ = 0 THEN 180
140 IF LZ = 18 THEN GOSUB 2290
150 LZ = LZ + 1; F% = F% + 1
160 PRINT CHR% (91) + RIGHT$ (
    " " + STR$ (F%) + CHR% (9
    3),4); SPC( 4);
170 FOR J = I + 3 TO I + 32; PRINT
    CHR% ( PEEK (J));; NEXT : PRINT
180 NEXT
190 NT% = PEEK (BF% + 1); NS% = PEEK
    (BF% + 2); IF NS% = 0 THEN PRINT
    ; GOSUB 2300; TEXT : RETURN

```

```

200 POKE TK%,NT%; POKE SC%,NS%; CALL
    RW%; GOSUB 2350; GOTO 130

```

File Summary main subroutine

```

300 HOME : VTAB 6; INVERSE : HTAB
    14; PRINT " DISK SNOOPER "
310 VTAB 8; HTAB 10; PRINT " FIL
    E SUMMARY UTILITY "; NORMAL
320 VTAB 11; HTAB 8; PRINT "ENTE
    R FILE NAME OR NUMBER:"
330 VTAB 16; HTAB 11; PRINT "YOU
    MAY ALSO ENTER:"
340 VTAB 18; HTAB 11; PRINT "'CA
    T' FOR CATALOG"; HTAB 11; PRINT
    "OR <RETURN> TO END"
350 POKE - 16368,0; VTAB 13; HTAB
    8; INPUT ""; IN%; IF IN% = ""
    THEN RETURN
360 IF IN% = "CAT" THEN GOSUB 1
    00; GOTO 300
370 IF LEN (IN%) > 30 THEN IN% =
    LEFT$ (IN%,30)
380 IF LEFT$ (IN%,1) < "0" OR LEFT$
    (IN%,1) > "9" THEN 500
390 E% = 0; FOR I = 1 TO LEN (IN
    %)
400 IF MID$ (IN%,I,1) < "0" OR
    MID$ (IN%,I,1) > "9" THEN E
    % = 1
410 NEXT : IF E% THEN M% = 0; GOSUB
    2260; GOTO 300
420 N% = VAL (IN%); POKE TK%,17;
    POKE SC%,15; F% = 1
430 CALL RW%; GOSUB 2350; I = BF%
    + 11
440 TZ = PEEK (I); IF TZ = 0 OR
    TZ = 255 THEN 470
450 IF N% = F% THEN N% = ""; FOR
    J = I + 3 TO I + 32; N% = N% +
    CHR% ( PEEK (J)); NEXT : GOTO
    700
460 F% = F% + 1
470 I = I + 35; IF I < = BF% + 2
    21 THEN 440
480 NT% = PEEK (BF% + 1); NS% = PEEK
    (BF% + 2); IF NS% = 0 THEN M
    % = 1; GOSUB 2260; GOTO 300
490 POKE TK%,NT%; POKE SC%,NS%; GOTO
    430
500 POKE TK%,17; POKE SC%,15
510 CALL RW%; GOSUB 2350; I = BF%
    + 11
520 TZ = PEEK (I); IF TZ = 0 OR
    TZ = 255 THEN 590

```

```

530 J = I + 2; K = 1
540 IF MID$ (IN%,K,1) < > CHR%
    ( PEEK (J + K) - 128) THEN 5
    90
550 K = K + 1; IF K < = LEN (IN
    %) THEN 540
560 IF PEEK (J + K) < > 160 THEN
    590
570 K = K + 1; IF K < = 30 THEN
    560
580 N% = IN%; GOTO 700
590 I = I + 35; IF I < = BF% + 2
    21 THEN 520
600 NT% = PEEK (BF% + 1); NS% = PEEK
    (BF% + 2); IF NS% = 0 THEN M
    % = 1; GOSUB 2260; GOTO 300
610 POKE TK%,NT%; POKE SC%,NS%; GOTO
    510

```

File Summary data subroutine

```

700 FOR K = 0 TO 5; TR%(K) = 255;
    NEXT
710 HOME : INVERSE : PRINT "FILE
    ";; NORMAL : PRINT " N%:TZ
    = PEEK (I + 2)
720 T% = " (UNLOCKED)"; IF TZ > =
    128 THEN T% = " (LOCKED)"; TZ
    = TZ - 128
730 IF TZ = 0 THEN T% = " TEXT F
    ILE" + T%
740 IF TZ = 1 THEN T% = " INTEGE
    R PROGRAM" + T%
750 IF TZ = 2 THEN T% = " APPLES
    OFT PROGRAM" + T%
760 IF TZ = 4 THEN T% = " BINARY
    FILE" + T%
770 IF LEFT$ (T%,2) = " (" THEN
    T% = " UNKNOWN" + T%
780 INVERSE : PRINT "TYPE:";; NORMAL
    : PRINT T%; POKE 34,3; HOME
790 IF TZ = 1 OR TZ = 2 OR TZ =
    4 THEN GOSUB 1800
800 TR%(0) = PEEK (I); SR%(0) = PEEK
    (I + 1); K = 0; CT% = 0
810 POKE TK%,TR%(K); POKE SC%,SR
    %(K); CALL RW%; GOSUB 2350
820 IF PEEK (BF% + 1) = 0 AND PEEK
    (BF% + 2) = 0 THEN 840
830 K = K + 1; TR%(K) = PEEK (BF%
    + 1); SR%(K) = PEEK (BF% +
    2); GOTO 810
840 K = 0; INVERSE : PRINT "T/S L
    IST:";; NORMAL

```

```

850 HTAB 11 + 6 * K: PRINT TRZ(K
)"/"SRZ(K);
860 K = K + 1: IF K < 5 AND TRZ(K
) < 255 THEN 850
870 K = 0: VTAB 6: HTAB 1: INVERSE
: PRINT "FILE DATA SECTORS:"
: NORMAL : POKE 34,6: HOME
880 POKE TKZ,TRZ(K): POKE SCZ,SR
Z(K): CALL RWZ: GOSUB 2350
890 IF K < 1 THEN 910
900 VTAB 4: HTAB (K - 1) * 6 + 1
1: PRINT TRZ(K - 1)"/"SRZ(K -
1);
910 VTAB 4: HTAB K * 6 + 11: INVERSE
: PRINT TRZ(K)"/"SRZ(K): NORMAL

920 SZ = PEEK (BFZ + 12):CZ = 0:
RZ = 7
930 FOR I = BFZ + 12 TO BFZ + 25
4 STEP 2
940 IF PEEK (I) = 0 AND PEEK (
I + 1) = 0 THEN 1000
950 IF PEEK (I) < > SZ THEN CZ
= CZ + 1:RZ = 7
960 IF CZ > 3 THEN VTAB 24: HTAB
9: GOSUB 2300: HOME :CZ = 0:
RZ = 7
970 VTAB RZ: HTAB 2 + 10 * CZ
980 SZ = PEEK (I): PRINT RIGHT$
("0" + STR$(SZ),2);
990 PRINT "/" RIGHT$ ("0" + STR$(
PEEK (I + 1)),2):RZ = RZ +
1:CTZ = CTZ + 1

1000 NEXT
1010 K = K + 1: IF K > 4 OR TRZ(K
) = 255 THEN GOSUB 2230: VTAB
24: HTAB 9: GOSUB 2300: TEXT
: GOTO 300
1020 VTAB 24: HTAB 9: GOSUB 2300
: HOME : GOTO 880

```

## Read Sector main subroutine

```

1100 TEXT : HOME : VTAB 5: HTAB
14: INVERSE : PRINT " DISK S
NOOPER "
1110 VTAB 7: HTAB 10: PRINT " RE
AD SECTOR UTILITY ": NORMAL
1120 TZ = 1: GOSUB 2000
1130 VTAB 21: HTAB 11: PRINT "SE
LECTION:": CALL - 868: POKE
- 16368,0
1140 GET IN$: PRINT IN$: IF IN$ =
"R" THEN GOSUB 2150: GOTO 1
100

```

```

1150 IF IN$ = "Q" THEN RETURN
1160 IF IN$ < > "D" THEN PRINT
CHR$(7): GOTO 1130
1170 IF PEEK (ERZ) = 255 THEN VTAB
23: HTAB 13: INVERSE : PRINT
CHR$(7):"BUFFER IS EMPTY":
NORMAL : HTAB 8: GOSUB 2300
: GOTO 1100

1180 TZ = 12: GOSUB 2000
1190 VTAB 21: HTAB 11: PRINT "SE
LECTION DISPLAY:": CALL - 868
: POKE - 16368,0

1200 GET IN$: PRINT IN$: IF IN$ =
"D" THEN DZ = 0: GOSUB 30: GOTO
1240
1210 IF IN$ = "H" THEN DZ = 0: GOSUB
60: GOTO 1240
1220 IF IN$ < > "C" THEN PRINT
CHR$(7): GOTO 1190
1230 DZ = 1
1240 GOSUB 2100
1250 FOR I = 0 TO 16: VTAB I + 6
: HTAB 1: INVERSE
1260 PRINT RIGHT$ ("00" + STR$(
I * 8) + " ",4):BAZ = PEEK
(40) + 256 * PEEK (41)
1270 IF DZ THEN FOR J = 0 TO 7:
POKE BAZ + 7 + 4 * J: PEEK
(BFZ + I * 8 + J): NEXT J,I:
GOTO 1290

1280 NORMAL : PRINT L$(I): NEXT

1290 TZ = 0:BZ = 16: POKE - 1636
8,0
1300 IF PEEK (- 16384) < 128 THEN
1300
1310 EZ = PEEK (- 16384): POKE
- 16368,0
1320 IF EZ = 141 THEN 1100
1330 IF EZ = 136 THEN 1400
1340 IF EZ < > 149 THEN 1300
1350 IF BZ > 30 THEN 1300
1360 BZ = BZ + 1:TZ = TZ + 1: CALL
- 912
1370 INVERSE : VTAB 22: HTAB 1: PRINT
RIGHT$ ("00" + STR$(BZ *
8) + " ",4);
1380 IF DZ THEN FOR I = 0 TO 7:
POKE BBZ + 4 * I, PEEK (BFZ
+ BZ * 8 + I): NEXT : GOTO
1300
1390 NORMAL : PRINT L$(BZ): GOTO
1300
1400 IF TZ < 1 THEN 1300

```

```

1410 TZ = TZ - 1:BZ = BZ - 1: CALL
800: INVERSE
1420 VTAB 6: HTAB 1: PRINT RIGHT$(
"00" + STR$(TZ * 8) + " "
,4);
1430 IF DZ THEN FOR I = 0 TO 7:
POKE TBZ + 4 * I, PEEK (BFZ
+ TZ * 8 + I): NEXT : GOTO
1300
1440 NORMAL : PRINT L$(TZ): GOTO
1300

```

## Program initialization

```

1500 POKE 235, PEEK (115): POKE
236, PEEK (116)
1510 DIM TRZ(5),SRZ(5),M$(1),L$(
31)
1520 HIMEM: 13567
1530 RWZ = 768:TKZ = 781:SCZ = 78
2:ERZ = 790:TBZ = 1671:BBZ =
1751:BFZ = 13568
1540 FOR I = 0 TO 6: READ C$(I):
NEXT
1550 FOR I = 0 TO 1: READ M$(I):
NEXT
1560 FOR I = RWZ TO RWZ + 29: READ
TZ: POKE I,TZ: NEXT
1570 FOR I = 800 TO 862: READ TZ
: POKE I,TZ: NEXT

```

## Main menu

```

1600 TEXT : HOME : VTAB 5: HTAB
14: INVERSE : PRINT " DISK S
NOOPER "
1610 VTAB 7: HTAB 15: PRINT " MA
IN MENU ": NORMAL
1620 FOR I = 0 TO 3: VTAB 11 + 2
* I: HTAB 15: PRINT C$(I);
1630 HTAB 15: INVERSE : PRINT LEFT$(
C$(I),1): NORMAL : NEXT
1640 VTAB 20: HTAB 15: PRINT "SE
LECTION:": CALL - 868: POKE
- 16368,0
1650 GET IN$: PRINT IN$: IF IN$ =
"C" THEN 1700
1660 IF IN$ = "F" THEN 1710
1670 IF IN$ = "R" THEN 1720
1680 IF IN$ = "Q" THEN POKE 115
, PEEK (235): POKE 116, PEEK
(236): HOME : PRINT "END DIS
K SNOOPER": END
1690 PRINT CHR$(7): GOTO 1640
1700 GOSUB 100: GOTO 1600
1710 GOSUB 300: GOTO 1600
1720 GOSUB 1100: GOTO 1600

```

Show special information for BASIC and binary file

```
1800 NT% = PEEK (TK%);NS% = PEEK
      (SC%); VTAB 4
1810 POKE TK%, PEEK (I); POKE SC
      %, PEEK (I + 1); CALL RW%; GOSUB
      2350
1820 POKE TK%, PEEK (BF% + 12); POKE
      SC%, PEEK (BF% + 13); CALL R
      W%; GOSUB 2350
1830 IF T% < 3 THEN M% = "PROGRA
      M LENGTH: "; GOTO 1850
1840 M% = "DEFAULT BLOAD ADDR: "
1850 LO = BF%:HI = BF% + 1
1860 AD = PEEK (LO) + 256 * PEEK
      (HI)
1870 D = PEEK (HI); GOSUB 10:AD%
      = H%
1880 D = PEEK (LO); GOSUB 10:AD%
      = AD% + H%
1890 PRINT M%;AD ("AD%");: IF
      T% < 3 THEN PRINT "BYTES":
      PRINT
1900 IF T% = 4 THEN PRINT :M% =
      "RAM IMAGE LENGTH: ":LO = BF
      % + 2:HI = BF% + 3:T% = 0: GOTO
      1860
1910 POKE TK%,NT%; POKE SC%,NS%;
      CALL RW%; GOSUB 2350: GOSUB
      2300: RETURN
```

Miscellaneous subroutines

```
2000 FOR I = 1 TO 5: VTAB 8 + 2 *
      I: HTAB 11
2010 IF I = 1 AND T% = 12 THEN PRINT
      SPC( 11): GOTO 2040
2020 IF I = 5 AND T% = 12 THEN PRINT
      SPC( 4): GOTO 2040
2030 PRINT C%(2 + I - 4 * (I = 5
      ) - (I = 1))
2040 NEXT : INVERSE
2050 FOR I = 1 TO 5: VTAB 8 + 2 *
      I: HTAB 10 + T%
2060 IF (I = 1 OR I = 5) AND T% =
      12 THEN 2080
2070 PRINT MID% (C%(2 + I - 4 *
      (I = 5) - (I = 1)),T%,1)
2080 NEXT : NORMAL : RETURN
2090 HTAB 3: PRINT "BUFFER CONTA
      INS: TRACK " PEEK (TK%), SE
      CTOR " PEEK (SC%): RETURN
2100 TEXT : HOME : VTAB 2: GOSUB
      2090
2110 VTAB 4: HTAB 1: INVERSE : PRINT
      "BYTE";: FOR I = 0 TO 7
```

```
2120 PRINT SPC( 3);I;: NEXT : PRINT
      " ": PRINT SPC( 4): NORMAL
2130 VTAB 24: HTAB 3: PRINT "PRE
      SS: <--, -->, OR <RETURN> TO
      QUIT";
2140 POKE 34,5: POKE 35,22: RETURN
2150 VTAB 10: HTAB 1: CALL - 95
      8: HTAB 11: INVERSE : PRINT
      C%(2): NORMAL
2160 VTAB 12: HTAB 11: INPUT "TR
      ACK (0-34):";IN%
2170 IF LEN (IN%) > 2 OR LEFT%
      (IN%,1) < "0" OR LEFT% (IN%
      ,1) > "9" OR RIGHT% (IN%,1)
      < "0" OR RIGHT% (IN%,1) >
      "9" OR VAL (IN%) > 34 THEN
      2150
2180 BT% = VAL (IN%)
2190 VTAB 14: HTAB 11: INPUT "SE
      CTOR (0-15):";IN%
2200 IF LEN (IN%) > 2 OR LEFT%
      (IN%,1) < "0" OR LEFT% (IN%
      ,1) > "9" OR RIGHT% (IN%,1)
      < "0" OR RIGHT% (IN%,1) >
      "9" OR VAL (IN%) > 15 THEN
      2150
2210 BS% = VAL (IN%); POKE TK%,B
      T%; POKE SC%,BS%; CALL RW%; GOSUB
      2350
2220 VTAB 17: GOSUB 2090: VTAB 2
      4: HTAB 9: GOSUB 2300: RETURN
2230 FOR I = 0 TO 5: IF TR%(I) <
      255 THEN CT% = CT% + 1
2240 NEXT
2250 VTAB 6: HTAB 25: INVERSE : PRINT
      "SECTOR COUNT:"CT%; NORMAL :
      RETURN
2260 VTAB 14: HTAB 8: FLASH
2270 PRINT CHR% (7);M%(M%): FOR
      I = 1 TO 3000: NEXT
2280 NORMAL : TEXT : RETURN
2290 LZ = - 1: VTAB 24: PRINT "O
      R <ESC> TO ABORT";: VTAB 23:
      HTAB 1: PRINT "(MORE) ";
2300 PRINT "PRESS ANY KEY TO CON
      TINUE";: POKE - 16368,0
2310 IF PEEK ( - 16384) < 128 THEN
      2310
2320 IF PEEK ( - 16384) = 155 AND
      LZ = - 1 THEN POP : TEXT
2330 HOME : RETURN
2340 VTAB 24: HTAB 11: FLASH : PRINT
      " ONE MOMENT, PLEASE ";: NORMAL
      : RETURN
```

```
2350 IF PEEK (ER%) < > 64 AND
      PEEK (ER%) < > 128 THEN RETURN
```

```
2360 INVERSE : VTAB 23: HTAB 14:
      PRINT CHR% (7);"UNABLE TO
      READ"
```

```
2370 NORMAL : HTAB 8: GOSUB 2300
      : POKE ER%,255: POP : RETURN
```

String data

```
2400 DATA CATALOG,FILE SUMMARY,R
      EAD SECTOR,QUIT,DISPLAY IN D
      ECIMAL,DISPLAY IN HEXADECIMA
      L,DISPLAY AS CHARACTERS
```

```
2410 DATA ILLEGAL ENTRY,FILE NOT
      FOUND
```

RWTS subroutine data

```
2420 DATA 169,3,160,9,32,217,3,9
      6,0,1,96,1,0,0,0,26,3,0,53,0
      ,0,1,255,0,96,1,0,1,239,216
```

Reverse scrolling routine data

```
2430 DATA 165,35,56,233,1,72,32,
      36,252,165,40,133,66,165,41,
      133,67,164,33,136
```

```
2440 DATA 104,56,233,1,197,34,48
      ,13,72,32,36,252,177,40,145,
      66,136,16,249,48
```

```
2450 DATA 224,160,0,32,82,3,176,
      12,164,36,169,160,145,40,200
      ,196,33,144,249,96,76,34,252
```

## APPLE™ SWAT TABLE FOR: DISK SNOOPER

LINES	SWAT CODE	LENGTH
1 - 120	LD	374
130 - 330	KB	396
340 - 450	RK	412
460 - 570	LV	308
580 - 770	DW	406
780 - 890	NQ	399
900 - 1010	ZS	365
1020 - 1200	LC	401
1210 - 1320	DL	301
1330 - 1440	MN	323
1500 - 1630	SB	354
1640 - 1820	KD	341
1830 - 2020	TC	383
2030 - 2140	KA	333
2150 - 2260	ZQ	438
2270 - 2400	IU	444
2410 - 2450	VS	346



# THE ADVENTURES OF PROFESSOR Von Chip and Orbie



## TRICKY TUTORIALS (tm)

There are many things that the ATARI computers can do either better, or easier than other small computers. The following series of programs is designed for anyone who is at least familiar with BASIC programming. What each tutorial offers is similar to an extensive magazine article with all discussion in as simple language as possible, plus you get MANY examples already typed in and running. The instruction manuals range from 10 to 50 pages, and some tutorials fill up a complete tape or disk. There is little overlap in what is taught, so anyone wanting to know all they can should buy them all (my banker thanks you). ATARI buys these from us to use in training their own people! Rave reviews have been published in ANTIC, ANALOG, CREATIVE COMPUTING, and even INFOWORLD. You trust INFOWORLD, don't you?

**TT #1: DISPLAY LISTS**—This program teaches you how to alter the program in the ATARI that controls the format of the screen. Normally, when you say "Graphics 8", the machine responds with a large Graphics 8 area at the top of the screen and a small text area at the bottom. Now, you will be able to mix various Graphics modes on the screen at the same time. The program does all of the difficult things (like counting scan lines). You will quickly be able to use the subroutines included in your own programs. **16K Tape or 24K Disk. \$19.95**

**TT #2: HORIZONTAL/VERTICAL SCROLLING**—The information you put on the screen, either GRAPHICS or TEXT, can be moved up, down, sideways, or diagonally. We provide the basic methods and leave the rest up to your skill and imagination. Includes 18 examples to get you started, with several using a small machine language subroutine for smoothness. **16K Tape or 24K Disk. \$19.95**

**TT #3: PAGE FLIPPING**—Now you don't have to redraw the screen every time you change the picture or text. You will learn how to have the computer draw the next screen you want to see while you are still looking at the previous screen, then flip to it instantly. You won't see it being drawn, so a complicated picture can seem to just appear. Depending on your memory size and which graphics or text modes you are using, you can instantly look at up to 50 pages. The basic method takes only 9 lines and the usefulness is infinite. **16K Tape or 24K Disk. \$19.95**

**TT #4: BASICS OF ANIMATION**—This program shows you how to animate simple shapes (with some sound) using the PRINT and PLOT commands, and it also has a nice little PLAYER/MISSILE GRAPHICS game you can learn from. The P/M example is explained and will get you started on this complicated subject (more fully explained in TT #5). This would be an excellent way to start making your programs come alive on the screen with movement! Recommended for beginning users. **16K Tape or 24K Disk. \$19.95**

**TT #5: PLAYER/MISSILE GRAPHICS**—Learn to write your own games and other animated applications! The tutorial begins with many small examples that compliment the 50 page manual, then gradually builds up to a complete game where everything you need to know is fully explained. Also included are two machine language utilities that you can use to animate Players with from BASIC. Next we include two of the best editors currently available; one for editing playfield shapes (backgrounds); and one to edit your players, and all in glorious Technicolor!! Everything except the two editors run in 16K Tape or 32K Disk. **\$29.95**

**TT #6: SOUND AND MUSIC**—Unless you have spent many years experimenting with the four voice channels, you will learn a lot from this one! Learn to play standard notes, chords, and whole songs using some simple "tricks". One of the nicest parts are the examples of special sound effects that you can refer to whenever you need a sound for a program or to impress a friend. This program will be of interest to all ages and levels of experience! **16K Tape or 24K Disk. \$19.95**

### SPECIAL DISCOUNT

Order the first six tutorials in a 3-ring binder for \$99.95, a \$30.00 savings!

**TT #7: DOS UTILITIES**—We at Educational Software have been shocked by some of the prices others are charging to offer you small utilities to help in the use of your Disk Drive. We now offer you all of the following plus explanation as to how each was written, and how to use them: A UNIQUE MENU PROGRAM, AN AUTORUN.SYS BUILDER, DISK INSPECTOR (LOOK AT SECTORS), DISK JACKET PRINTER, AUTOMATIC FORMATTER, RECORD SAVE AND LOAD UTILITY. **32K Disk Only. \$29.95**

## MASTER MEMORY MAP (tm)

This book is the most valuable source of information for your ATARI you can buy. It starts out by explaining how to PEEK and POKE values into memory, so that even new computer owners can use many of these "Tricks". Then you are given 32 pages of the memory locations that are the most useful, along with hints on how to use many of the locations. Finally, it includes hints on problems you may be having with the computer and discusses the new Graphics modes 9 to 11. Even ATARI buys this book from us! **\$6.95**

## CONTACT YOUR LOCAL DEALER or ORDER BELOW

We have other fine programs for ATARI computers. Write for a catalog. Send us your programs to sell too!

## USER SUBMITTED PROGRAMS

**MINI-DATABASE/DIALER**—stores and edits up to 8 lines of information such as names & addresses, phone numbers, messages, inventories, or anything you want. It has the usual sort, search, and print options, but it also has an unusual feature: If your file includes phone numbers and your phone company allows touch-tone phone signals, the program will DIAL THE PHONE NUMBER FOR YOU! **16K Tape or 24K Disk. \$24.95**

**THE GRAPHICS MACHINE**—allows the ATARI to act like more expensive graphics computers using simple commands like line, box, circle, polygon, fill, and savescreen to get a high resolution picture you can save on disk in only five seconds! Many more features! **48K Disk Only. \$19.95**

**BOB'S BUSINESS**—14 small business type programs accessed from a common menu. **16K Tape or 32K Disk. \$14.95**

**MINI WORDPROCESSOR**—A simple text editor to write, save, and print several pages at a time. **32K Tape or Disk. \$19.95**

**KID'S #1**—Includes a MATH QUIZ, a children's TREASURE HUNT, and a DIALOGUE program. **16K Tape or 24K Disk. 3 for \$14.95**

**KID'S #2**—SPELLING BEE, WORD SCRAMBLE, and TOUCH. **16K Tape or 24K Disk. 3 Educational games for \$14.95**

**PLAYER PIANO**—Turns your keyboard into a mini-piano and more. **24K Tape or 32K Disk. \$14.95**

**GRAPHIC SYMBOL LABELS**—for your keyboard to remind you of the built-in Graphics symbols. 2 complete sets for... **\$2.95**

## OUR NEWEST PROGRAMS

**DOG DAZE**—Two cute little doggies race for the fire hydrants, shoot their bones, and just have a lot of fun! A fast action program for all ages. **8K Tape or 16K Disk, in machine language. \$16.95**

### \* OUR BEST GAME \*

**SPACE GAMES**—Our family is being attacked by ALIENS, and only you can save us. A comic book manual will guide you through three games that test your ability in space skills. Includes ALIENS, SURVIVE, and ROBOT ATTACK, and is for all ages. The first two games require 16K for Tape. The last game and all Disk users need 32K. **\$24.95**

**MATHS FOR FUN**—Another ENGLISH import teaching basic math skills. Very colorful and enjoyable to use. For ages 5 to 16. **16K Tape or 24K Disk. \$19.95**

**MARATHON**—This is a unique math quiz for one or two players. You are in a race to move your runner across the screen first! There are four levels of play with five modes of operation for each. The game uses joysticks for all input, so play is easy for young children. This wonderful learning tool is imported from ENGLAND for your learning pleasure. Your kids will never even notice they are playing an EDUCATIONAL program. **16K Tape or 24K Disk. \$19.95**

**GRADE BOOK**—This nice record keeper will maintain a file of 35 students' test scores along with comments. Up to 7 subjects are allowed. **24K Tape or 32K Disk. A printer is optional. \$24.95**

To order COD, VISA or MasterCard call — (408) 476-4901

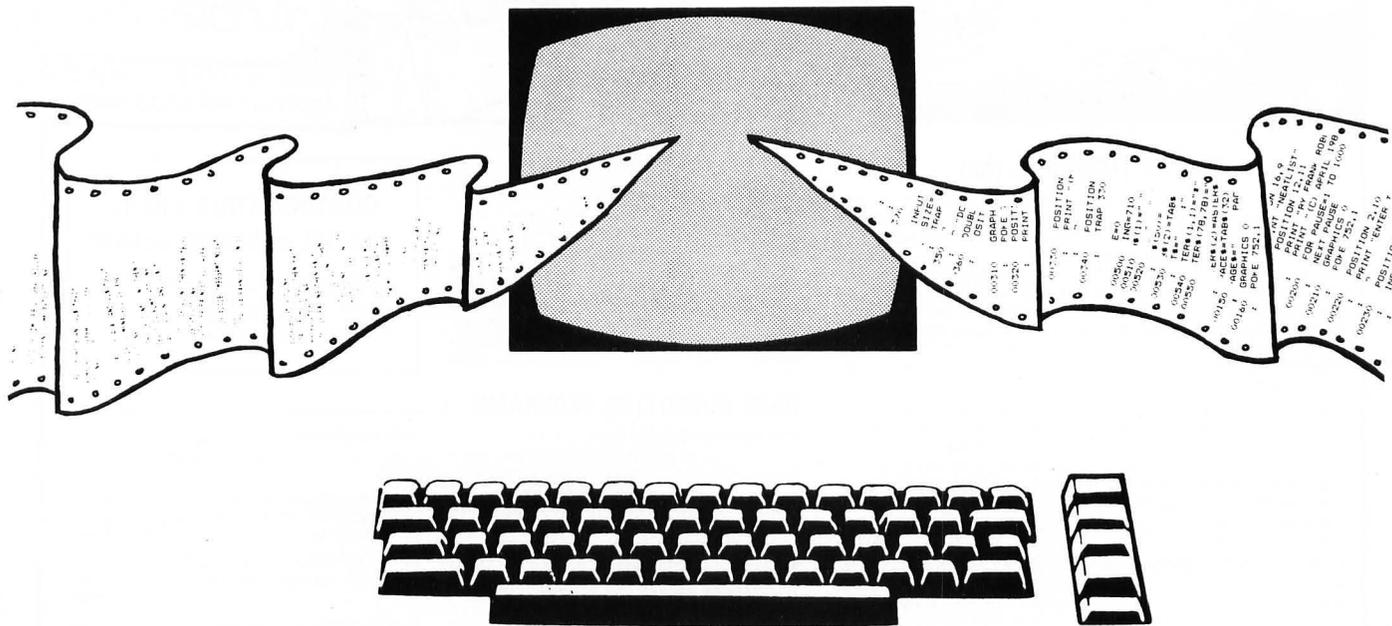
By mail include \$3.00 postage (USA only) or \$1.50 Memory Map only—California residents add 6.5% TAX.—Specify Tape or disk.



EDUCATIONAL SOFTWARE inc.

5425 JIGGER DR.  
SOQUEL, CA 95073

# NEATLIST



by Frank Roberts

**NEATLIST** is an ATARI® utility program requiring an 80-column printer, disk drive (see cassette modification), and 16K RAM. It is included as the bonus program on this month's ATARI® DV.

If you have ever tried to debug or change a BASIC program containing several statements in each line, you may enjoy **NEATLIST**. It is a utility which lists a program to the printer in a special format: Every statement in each line is listed vertically on an individual line beneath its corresponding line number. REM statements are set apart from the program with a border of asterisks. This makes documentation much easier to locate and check. This program utilizes the ATARI's LIST "D:" command; the user must first list the program to a disk, then run **NEATLIST**. The program will take care of

the rest. This utility is nearly indispensable when debugging heavily "crunched" programs — and healthier than aspirin and eyedrops!

## Cassette Modification

**NEATLIST** can be modified for use with cassette systems by changing line 180 to read:

```
180 F$="C":F$(3)=R$
```

This will read the program from the cassette, providing that the user has stored the program to be printed on the cassette with the command LIST "C:".

It may take you a little experimentation to get the hang of this technique the first time you use the program, but the results will be well worth your efforts.

# ATARI® SILENCER

by John J. Anderson

It is commonly known that, in addition to the capability of driving sound through a television or monitor speaker, the ATARI® has an onboard speaker, similar to the Apple II™. This speaker can and does serve in a number of capacities, not the least of which is to sound a prompt or signal tone, to flag a specific mode or indicator.

Users of the 410 program recorder are familiar with the record and play tones sounded as an indicator before data input or output to tape. All users should be familiar with the chirp of keyboard feedback. This feature lends a surer "feel" to the keyboard than that found with other computers.

These features are, essentially, well-designed and helpful. However, I've discovered that there are times I wish I could fit a silencer onto my ATARI® 800. Late night editing sessions or programming when my roommate is trying to catch forty winks have caused friction. Certain programs I use very frequently, like the ATARI® *Word Processor*, seem to exploit the feature to a point beyond distraction. These features are helpful in a noisy office environment, but seem a bit heavy-handed in a quiet work area at home, the most common environment for the ATARI®. I nearly discontinued exploration of a hi-res adventure because the program continually prompted for pressing RETURN with a long, shrill "blat" — shades of operant conditioning! Is it too much to ask to be able to turn the thing on and off at will?

What could be simpler than the installation of a single pole, single throw switch to cut out the speaker when desirable? A "take-apart" since earliest childhood, I had already snooped around a bit inside the ATARI®, and knew how easy it really would be. But, I still had a

problem. The mere thought of snipping wires or drilling holes in my pristine machine made the hairs on the back of my neck stand on end. Also, though my warranty had long since expired, I wasn't happy with the idea of doing anything that couldn't be undone. Service people can be put off quickly when they see user modifications. I determined, rather wistfully, that I could live with the buzzers.

Then, while staring at all the little packages hanging on the wall of a nearby Radio Shack, I made a fascinating discovery — I saw a product called "two prong connectors," catalog number 274-342 — \$2.49 for a package of six. I noticed that the fit would be quite close to the connector used on the ATARI® speaker. I then noticed "SPST micro miniature toggle switch," catalog number 275-624 — \$1.59. Smaller than the smallest switch Radio Shack had stocked previously — it occurred to me that it would fit between the vent slots on

the bottom of the ATARI®. I suddenly envisioned a switch modification that was totally, and easily, reversible.

The modification was a complete success. Now that I can toggle the speaker off, I realize it's something I should have done long ago. In case I need to bring the computer in for service, the modification can be slipped out in under five minutes.

## The Project

If you wish to modify your ATARI®, you will need, in addition to the products listed above, about two feet of bell or other light wire, a flat blade and Phillips screwdriver, soldering iron and solder, and a bit of tape.

Snip the wire into two ten inch lengths. Then, take one of the wires and snip it into two five inch lengths. Strip a quarter inch of insulation off the ends of all the leads. Twist the shorter wires onto the longer wire in the manner indicated in Figure 1. This will make the

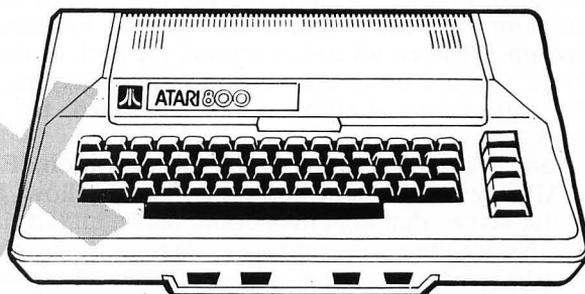
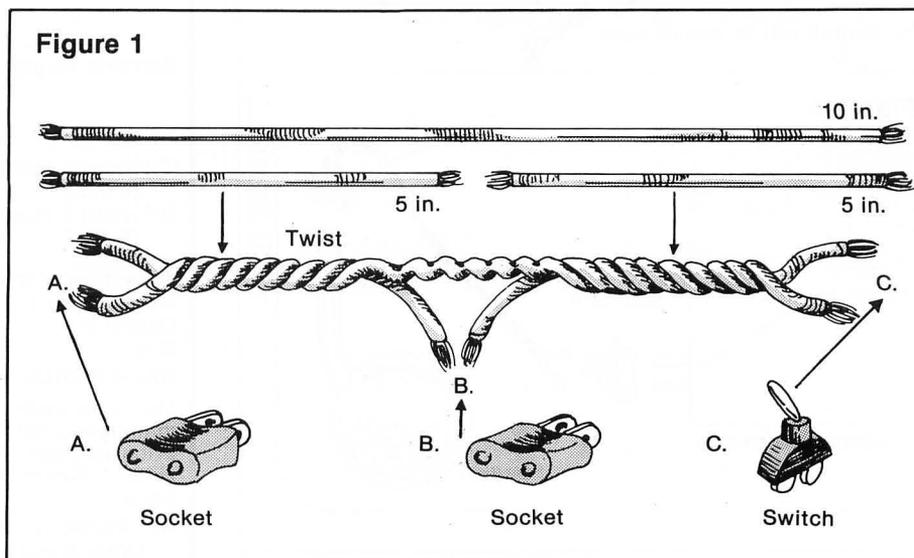


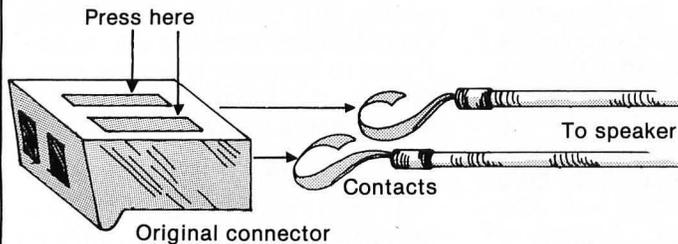
Figure 1



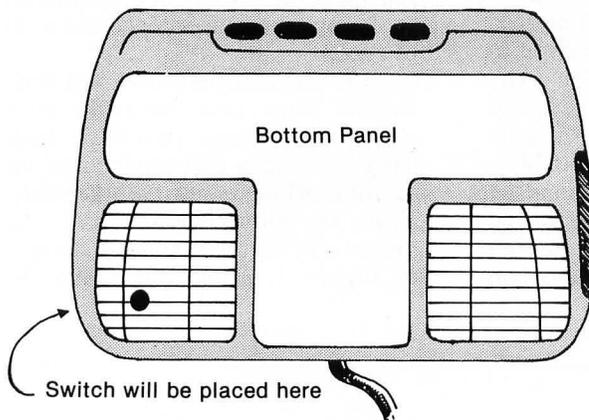
modification easier to slip in and out later. Next, solder two connectors and the switch to the wires as indicated in the diagram. Unscrew all collars around the neck of the switch. Notice you are using only the *socket* connectors, not the *plug* connectors. Leftovers can be saved for another project.

Now you are ready to begin the operation. Flip your ATARI® over onto something soft, like a pillow. Unscrew the five screws that hold the bottom panel, and lift it toward you. Notice that the controller ports must be cleared in order to remove the panel. Can you believe how small that speaker is? You now know another reason why you're lucky to own an ATARI®. You don't depend on that little thing for all your sound effects. To disconnect the speaker, pull gently on the connector. Once the speaker is disconnected, remove it from the machine.

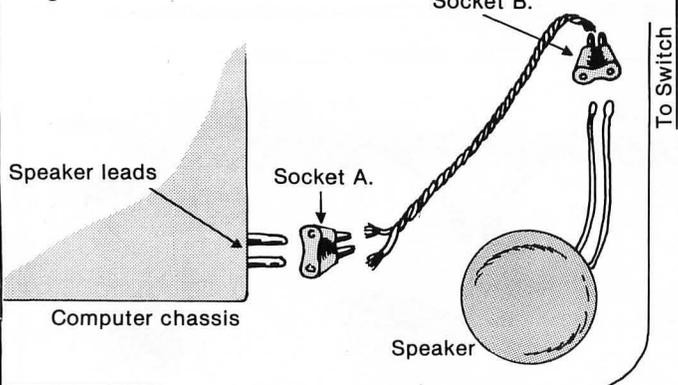
**Figure 2**



**Figure 3**



**Figure 4**



**SOFTSIDE  
ORDERING  
INFORMATION**

## FORM OF PAYMENT

### USA

VISA, MasterCard, certified checks, money orders and personal checks are accepted.

### Canada/Mexico

The preferred method of payment is by VISA or MasterCard. A bank check is acceptable if it has been preprinted for payment in U.S. dollars. No personal or company checks accepted.

### Other Foreign Orders

Payment must either be by a bank check drawn on a U.S. bank payable in U.S. dollars or by affiliated bank credit cards of VISA or MasterCard.

### GUARANTEE

All software is guaranteed to load and run. If you experience difficulties with the product within 30 days, it may be returned for replacement. Send your properly protected tape or disk to the attention of the Customer Service Representative and include your name, address, and the reason it is being returned.

### LIABILITY

All software is sold on an as-is basis. **SoftSide** assumes no liability for loss or damage caused or alleged to be caused directly or indirectly by products sold or exchanged by them or their distributors, including, but not limited to, any interruption in service, loss of business or anticipatory profits or consequential damages resulting from use or operation of such software.

### PRICES

Prices are subject to change without notice. We are not responsible for typographical errors.

Unless otherwise noted in a published advertisement, the following prices are in effect as of this issue:

	USA/Canada APO/FPO	USA/Canada FIRST CLASS Mexico	Other Foreign
SoftSide Magazine (yr)	\$30	\$40	\$62

	USA APO/FPO	Mexico Canada	Other Foreign
CV (year) & magazine (6 mo.)	\$75	\$95	\$125
DV (year) & magazine (6 mo.)	\$39	n/a	n/a
	\$125	\$145	\$175
	\$64	n/a	n/a

### Adventure of the Month Month (6 mo.)

	USA	Mexico	Other Foreign
Cassette	\$29	\$35	\$41
Disk	\$49	\$55	\$61

### BACK ISSUES

Minimum order for magazines only — 3 issues. There is no minimum order for magazine/media combinations. Price includes shipping to the 48 states, APO/FPO only. Alaska, Hawaii, Puerto Rico, and ALL foreign orders — postage is additional.

ALL Foreign orders and all magazine/media combination orders — Order directly from **SoftSide, 6 South St., Milford, NH 03055.**

# ATARI®

Orient the connector so that it matches the view in Figure 2. Using a screwdriver or toothpick, press down on the silver tongue on the top of the plastic connector, as you gently pull the wire from the side. **Don't force anything!** When you've pressed the tongue down far enough, the contact will slide right out. Pull both contacts out of the plastic container.

Next, take the bottom panel you removed earlier and hold it so that the vents are at the bottom, as shown in Figure 3. You will mount the switch in the left-hand vent, where there is room to spare, and nothing nearby that might get shorted out. Insert a flat blade screwdriver between the two vent slots where the switch will be mounted. (It's a good idea to stay over to the left — this will make the switch easier to reach.) Gently twist the screwdriver to spread the slot, then press the neck of the switch through. The plastic will have to bend a bit to accommodate the switch. Put on a washer, then screw on the lock nut to fasten the switch in place.

The final installation will be facilitated by repositioning the back panel so that the computer looks like an open valise. This way, the wire between switch and speaker will not be stretched. First, press the speaker contacts into the middle connector, as indicated in Figure 4. The speaker can now be repositioned in its place. Gently connect the far socket to the speaker leads from which you removed the original connector. Spreading them a bit may insure a tight fit. Finally, tuck the wire away under the keyboard post and away from boards and the speaker itself. There's enough room on that side of the computer to insure that the modification will not interfere with any other hardware.

You may wish to tape the original connector to the wire itself. Then, should you wish to remove the modification, the original connector will be right where you left it.

Screw the back panel on, plug things back in, and run a test. You can easily use the keyboard REPEAT function in memo pad mode to do this.

Listen. You can almost hear a pin drop! 



## GALACTIC CHASE™

The aliens have swept undefeated across the galaxy. You are an enterprising star ship captain—the final defender of space.

As the aliens attack, you launch a deadly barrage of missiles. Flankers swoop down on your position. Maneuvering to avoid the counterattack, you disintegrate their ships with your magnetic repellers.

As your skill improves, the attackers increase their speed. And as a last resort, the aliens use their invisible ray to slow the speed of your missile launcher.

GALACTIC CHASE provides Atari owners with the most challenging one or two person game in the galaxy.



Atari 400/800 16k. Written in machine language. Requires joysticks.

Payment: Personal Checks—allow three weeks to clear.

American Express, Visa, & Master Charge—include all numbers on card. Please include phone number with all orders. 24.95 for cassette or 29.95 for disk plus 2.00 shipping. Michigan residents add 4%.

Check the dealer in your local galaxy. Dealer inquiries encouraged.

Galactic Chase © 1981 Stedek Software.

**SPECTRUM**  
COMPUTERS

Dept S.  
26618 Southfield  
Lathrup Village, MI. 48076  
(313) 559-5252

## Deadline

Reviewed by Eric F. Wolcott

from Infocom, 55 Wheeler St., Cambridge, MA 02138. System requirements: Disk drive and 32K ATARI® 400/800, 32K Apple II+™ or 48K IBM® PC. Suggested retail price: \$49.95.

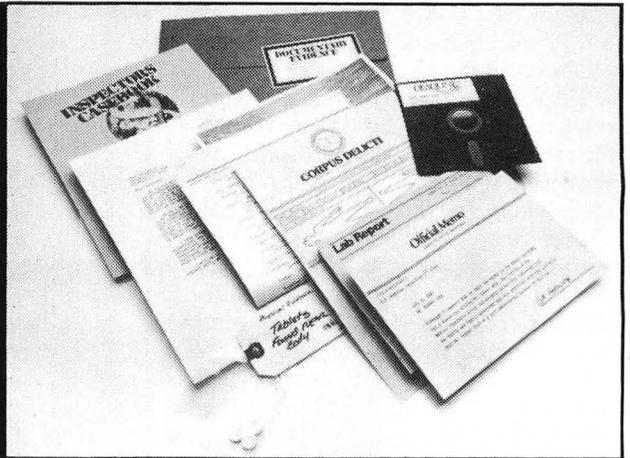
I was in the shower when the phone rang across the room. Cursing and dripping, I stumbled over to it. "Yeah," I growled, thinking this had better be good. It was a man with a problem. Yeah, I know, everybody has problems, but his was different. His was murder.

8:00 AM. I arrived at the Robner house thinking, "Why me?" Mr. Robner had been found dead in a locked room. Official cause of death was self-induced overdose; verdict, suicide. My instincts told me that the officials were wrong in this case. It was murder, but I would have to prove it. I thrive on adversity. This case was mine and I was going to finish it. If I didn't, the murderer would strike again. I was working with a *Deadline*.

*Deadline*, by Infocom, is a fantastic text adventure. Although some people believe that a text adventure cannot provide the involvement of a video adventure, many staunch adventurers believe the text adventure is the only way to go. *Deadline* is the text adventure for both groups. It provides a sense of close involvement, casting you in the role of a police inspector called in to solve a tough case. The novel packaging, good story line, and full sentence input combine to make an enthralling game.

When you get the package, you will be holding the official brown case folder containing the Robner file. Inside, you will find the coroner's death report, the detective's report, police transcripts of interviews with the household, a photograph of the body's placement, Exhibit A — a packet of pills found near the body, and the Inspector's Case Book. All these items

"Deadline provides a sense of close involvement, casting you in the role of a police inspector called in to solve a tough case."



contain necessary background information for solving the case.

The Inspector's Case Book is a well laid out, easy-to-read manual. It explains how the time element works in the game and how to make it work for you. It also describes how to use the police laboratory, handle the evidence, and use the stenographic services to get a transcript of your activities. Never does it refer to the game or player, always it refers to the case and the Inspector. In short, the Case Book is a complete set of rules that remains, at all times, in character with you.

The story line is excellent. It reads like a Micky Spillane mystery. There are full page descriptions of all the major areas on the Robner estate — standard fare for a text adventure. But *Deadline* goes one step beyond. You will not be the only person actively on the scene. Members of the Robner household will continue their daily routines, albeit disrupted somewhat by the demise of Mr. Robner. During your investigation you may interact with any one of these people. Talk to Mrs. Robner, the bereaved widow, as she eats breakfast. Follow Mrs. Rourke, the housekeeper, as she goes on her rounds. If you hurry, you will even find some clues before Mrs. Rourke innocently takes them away. The inhabitants of the Robner home will

also interact with you. This provides a sense of reality not usually found in adventures.

The full sentence input should be heralded as the greatest advance in adventure technology to date. Instead of commands such as USE PENCIL and GET FLASK, you may employ full sentences containing many words. Commands like PICK UP EVERYTHING or PICK UP EVERYTHING EXCEPT THE NEWSPAPER ON THE FLOOR are easily understood. The first command can be found in most adventures, the second is peculiar to *Deadline*. Now, does this mean that to move around you have to say WALK BRISKLY NORTH? No, N, S, U, D all work. Things need not become overly complicated if that is your wish. To further aid in your investigation, many of the necessary common verbs are explained in the Case Book.

Because of the story line, packaging, and sentence input, *Deadline* is an easy-to-use, enthralling game. Instead of a barbarian walking through a dead cavern, I was a police inspector moving through a hostile world. It was like a good book, and I was caught up in the plot. Adventurers from both camps should enjoy *Deadline* for the excitement and involvement it offers. Now excuse me, I have a *Deadline* to meet. 



**SECOND ANNUAL  
PHOENIX COMPUTER EXPO  
ONE PLACE  
TO SEE IT ALL!**

Mini and micro computers, software, graphics, data and word processing, telecommunications, peripheral equipment, supplies and computer services — you'll find it all at the Phoenix Computer Expo. September 30 - October 2 at Phoenix Civic Plaza.

You'll be able to sit down with a computer to evaluate its ability to meet your needs whether business, educational or for your home.

Throughout the three days, Hands-on workshops sponsored by RADIO SHACK, APPLE COMPUTER INC., AND IBM CORPORATION will be conducted.

Attend seminars designed to advise you where computer technology is heading and its influence on your life today and tomorrow.

And you'll enjoy yourself with computer music, Reggie the Robot, computer art and a host of other electronic for home and office equipment.

For further information contact:  
**Judco Computer Expos, Inc.**

P.O. Box 963 • Scottsdale, AZ 85252  
(602) 990-1715 or (800) 528-2355 (outside Arizona)

CO-SPONSORED BY DPMA PHOENIX CHAPTER  
DATA PROCESSING MANAGEMENT ASSOCIATION

**SEE, LEARN, PARTICIPATE  
AND ENJOY!**

---

**PHOENIX COMPUTER EXPO**

---

**Phoenix Civic Plaza**

225 East Adams

Thursday, September 30 - 10 a.m. - 8 p.m.

Friday, October 1 - 10 a.m. - 8 p.m.

Saturday, October 2 - noon - 6 p.m.

Admission \$7.50 per person and \$3.00 for children.

**PHOENIX COMPUTER EXPO**

**SAVE**

**\$2.50**

Present this coupon at the Phoenix Civic Plaza box office and save \$2.50 on one adult admission.

# UP PERISCOPE

by Ron Potkin

*Up Periscope* is a graphics wargame for a TRS-80® Model I or III with 32K RAM and disk drive. It is included as the bonus program on this month's TRS-80® Disk Version.

## PROLOGUE

The following conversation took place between the Admiral of the Fleet and his radio officer in the communications center:

**R.O.:** Signal coming through, Sir.

**Adm:** Read it to me.

**R.O.:** Proceed at once to escort convoy vessels across the sea from Westside to Eastside. Enemy submarines lying in the area. It is essential that at least three get through.

There's a new computer in the operations center — A 16K Level II, Sir. The programmer says we merely have to power it up, put the disk in the drive and select "UP PERISCOPE" from the menu.

**Adm:** What else?

**R.O.:** We can move in six directions, Sir. When it's your turn, each convoy vessel may move three hex in any direction. Each destroyer has 7 moves available. Moving one hex, turning, reversing, sonar and dropping a depth charge will each cost one move. Each destroyer has a complement of 10 depth charges.

**Adm:** Turns? Moves? What is this?

**R.O.:** We have to give the enemy an opportunity to shoot back, Sir. Incidentally, they move first.

**Adm:** What do we know about the submarines?

**R.O.:** We had some luck there, Sir. Apparently the programmer designed those as well. They can lie at three levels: on the surface, periscope depth, or on the bottom. They are each fitted with two tubes and carry 10 torpedoes. Each will have 6 moves. Moving one hex at periscope depth costs two moves. Moving on the surface, diving, rising and firing torpedoes each cost one move. They can only fire at periscope depth.

**Adm:** Sounds pretty dangerous. Is our sonar equipment in order?

**R.O.:** Yes, Sir. But there are snags.

**Adm:** What's that?

**R.O.:** We will not get a sounding if the sub is lying on the bottom and also, it cannot distinguish between a submarine and our own vessels and torpedoes.

**Adm:** How effective are our depth charges?

**R.O.:** I will check, Sir. (pause) The programmer says that dropping a depth charge immediately over a submarine will destroy it. Submarines in any of the hex surrounding that hex will be forced to the surface. And, there is a 50 percent chance that a tube will become inoperable.

**Adm:** That's good; and if they are on the surface can I ram them?

**R.O.:** Yes, but only with the sharp end, Sir. The convoy vessels may not.

**Adm:** O.K.. By the way, what is a hex?

**R.O.:** The sea is divided into 500 imaginary areas. Each is a regular six sided polygon. We don't have to concern ourselves too much with that. The main purpose is to indicate the direction for movement and so on.

**Adm:** How far will the convoy have to go?

**R.O.:** The sea is 32 hex wide from East to West. That means it will take them at least ten turns to get across. The torpedoes could well mean that it will take longer. You may know, Sir, that the Isle Of Radsha lies in this sea. There are many inlets where the subs can hide.

**Adm:** Yes, naturally.

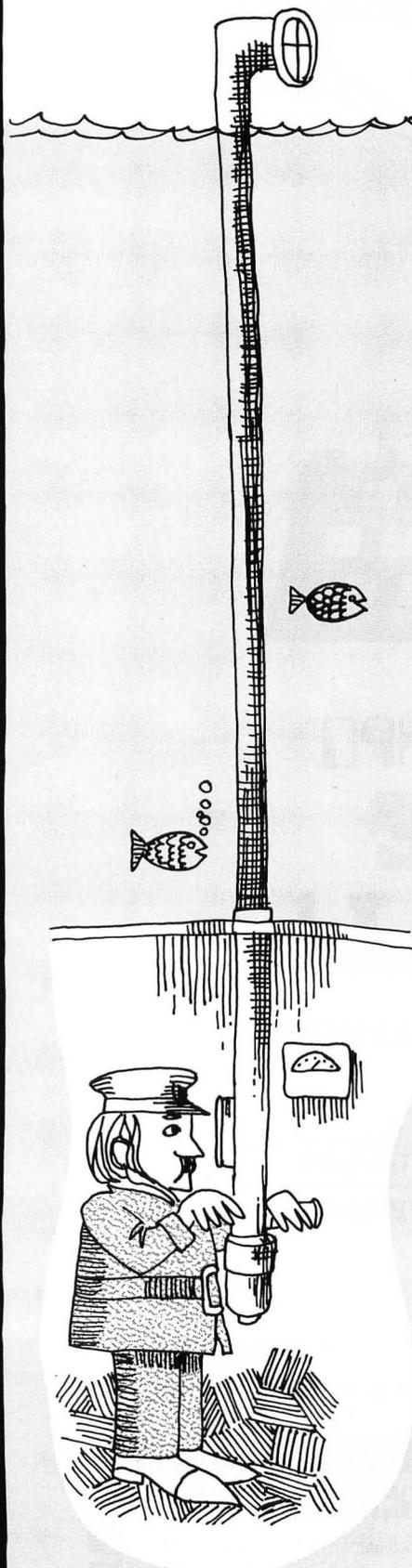
**R.O.:** There is one last point. I hardly know how to tell you this.

**Adm:** Well?

**R.O.:** The programmer says that he had difficulty accommodating all six directions on the screen. If a destroyer lies North-South, it's O.K.. But, if we turn in one of the other four directions, the shapes are not right and would we mind scrunching up a bit! Apparently the monitor can't handle diagonal lines.

**Adm:** It's not like the old days! Pass all the information we have along to the fleet and let's get under way.

**R.O.:** Aye, Aye, Sir.





At this point, a top priority message is received from the Submarine Commander. It reads:

"I understand that I must command my pack using a TRS-80® computer. I do not have one. May I use yours? Under the laws of computerized combat, I trust that you will adhere to the rules and not look when I make my moves."

The next few moments are difficult to describe coherently. However it has since been learned that the Admiral is recovering satisfactorily.

**1. Objective**

The Submarine Commander wins if the number of convoy vessels remaining plus the number that have crossed the sea successfully is less than 3 (or any number up to 6 depending on the difficulty level).

The Fleet Commander wins if all submarines are destroyed or at least 3 convoy vessels cross the sea.

**2. Vessels**

The Fleet Commander has 6 destroyers and 6 convoy vessels. The Submarine Commander has 10 submarines.

**3. Order of Play**

- (a) All submarines appear on the screen.
- (b) The Submarine Commander moves all, some or none of his pieces. This will include the firing of torpedoes at periscope depth.
- (c) All submarines, other than those on the surface, are removed from sight.
- (d) The Fleet Commander now moves all, some or none of his pieces — destroyers first, followed by his convoy.
- (e) All torpedoes move one hex in the direction they were initially fired.
- (f) An update of the number of pieces is made and the winner, if any, determined. Until there is a winner, steps (a) to (f) are repeated.

**4. Starting the Program**

You will be asked "HOW MANY VESSELS MUST GET THROUGH (1-6)?" Press any number from 1 to 6 and then ENTER. The second question will be "HOW MANY SUBMARINES (1-10)?" Press any number from 1-10 and press ENTER.

After a short pause, a statement of the current number of vessels will appear, followed by "SUBMARINE COMMANDER'S TURN". Press ENTER. At the end of his turn, the number of vessels will again be reported, followed by "FLEET COMMANDER'S TURN." Press ENTER again. This is important, as it enables the opponent to turn away before the map appears.

**5. Movement**

Each piece is moved in order. When it is due to be moved, it will flicker and show the number of moves left. A player may, during his turn, move his pieces up to the limit

of moves available (assuming they do not hit a torpedo).

All actions cost one move with the exception of submarines at periscope depth, where a move from one hex to an adjacent hex costs two moves.

To move a piece, other than destroyers, imagine a six-hour clock (see Figure 1).

To move Northeasterly, press "1", Southeasterly, press "2", to move South press "3" and so on.

The same method is used to indicate the direction for sonar, depth charges or torpedoes.

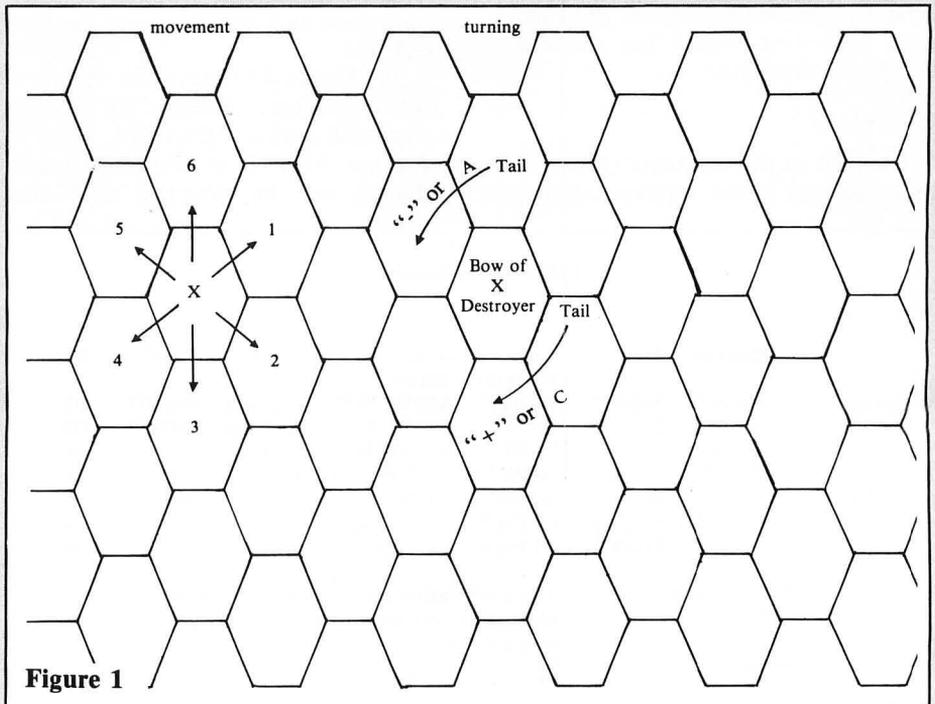
The following options are available to both players:

(a) NEXT: To move on to the next piece before the present piece has expended all its moves ... Press "N"

(b) DONE: If you decide that you have finished your turn even though pieces have not yet moved ... Press (ENTER)

Note that if you have any pieces on "HOLD" you may still move them.

(c) HOLD: The pieces will normally flicker in a pre-defined order.



**Figure 1**

If you wish to move in a different order, so that you can more easily clear a congested area ... Press "H"

The remaining options vary by type of vessel:

### Submarines

(a) Press a number from 1 to 6 to move one hex as described above.

(b) Press "T" to fire a torpedo. The piece will flicker "TPDO". Press 1 to 6 to indicate direction. Note that the torpedo is invisible until the Fleet Commander has finished his turn. (The torpedo was fired at periscope depth...It takes a moment to reach the surface.)

(c) Press "U" to move up one level.

(d) Press "D" to move down one level.

The piece will flicker "HOW?" If you press "U" and it is on the surface, or if you press "D" and it is on the bottom.

### Convoy Vessels

The only option available is to move from one hex to another. It is the only piece that may move off the board. This occurs only on the East edge indicating that it has made a successful crossing.

### Destroyers

The bow of the destroyer (the part that flickers) is the reference point

when indicating directions. The clock method is not used for movement. Instead, a number from 1 to 6 will indicate the number of hexes to move in the direction it is presently pointing. Any submarines lying in its path are rammed and destroyed. Other options are:

(a) Turns: When turning, the bow remains stationary and the stern moves to bring the ship around into the new direction. Port and starboard, left and right have been avoided. Instead, it makes either "C" (clockwise) or "A" (anticlockwise) turns. If this is confusing, use "+" (without shift key) or "-". So, if the stern is at 4 o'clock then "+" or "C" moves it to 5 o'clock. Similarly, if the stern is at 3 o'clock then "-" or "A" moves it to 2'clock. (See Figure 1)

(b) Reverse: To move the destroyer full astern one hex...Press "R".

(c) Sonar: To search for submarines or torpedoes, press "S". The piece will flicker "SONAR". Press a number from 1 to 6 to indicate direction. The bottom of the screen will indicate whether contact was made and how far away the object lies.

(d) Depth-Charges. To drop a depth charge, press "D". The piece will flicker "DCHG". Press a number from 1 to 6 and a depth charge will be dropped one hex

away from the bow of the ship. Any submarine or torpedo in that hex will be destroyed. Any submarines in surrounding hexes will be forced to the surface.

### 6. Special Feature

No two pieces may occupy the same hex. When a destroyer or convoy vessel enters a hex occupied by a submarine lying below the surface, the submarine is moved to a new location. Using the sub hex as the reference point, a search is made for an empty hex starting at 1 o'clock. If 1 through 6 are occupied, hexes two spaces away are searched. If all are occupied, the submarine is destroyed. This could happen around the island or at the edge of the board.

### 7. Table of Outcomes

The table below left may be consulted to check all outcomes when one object contacts another. The column indicates the piece being moved; the row is the piece contacted:

#### DESIGN NOTES

It is difficult to know who has the advantage in this game. It may be that the submarines have the edge; hence the reason for being able to change the level of difficulty.

The destroyers are fairly powerful. The sonar and the ability to bring submarines to the surface with a near miss does give them plenty of clout. Their best defense is to find and destroy, not waiting for the submarines to move in. Once the submarines close in on the convoy, they will normally cause considerable damage.

The submarines have a considerable advantage. Their strategy should be to move slowly, always ending their turn on the bottom. Above all, they must remain well apart and not fire torpedoes unless they are confident they can get well away. The appearance of a torpedo is, of course, their greatest giveaway.

Table of Outcomes

	Convoy	Dest'r	Sub. Surface	Sub. Below	Torp.	Edge	Space
Convoy	HOW?	HOW?	HOW?	MOVE SUB	CLEAR	HOW?	OK
Des(move)	HOW?	HOW?	RAM	MOVE SUB	CLEAR	HOW?	OK
Des(rev)	HOW?	HOW?	HOW?	MOVE SUB	CLEAR	HOW?	OK
Sub(Surf)	HOW?	HOW?	HOW?	MOVE SUB	HOW?	HOW?	OK
Sub(Below)	HOW?	HOW?	HOW?	HOW?	HOW?	HOW?	OK
Torpedo	CLEAR	CLEAR	CLEAR	MOVE SUB	CLEAR	CLEAR	OK
Dchge	HOW?	HOW?	CLEAR	CLEAR	CLEAR	HOW?	OK

MOVE SUB	Move submarine and continue as for Space
CLEAR	Both pieces are destroyed
OK	Normal move
HOW?	Error condition

## Alien Defense

Reviewed by Andre Chen

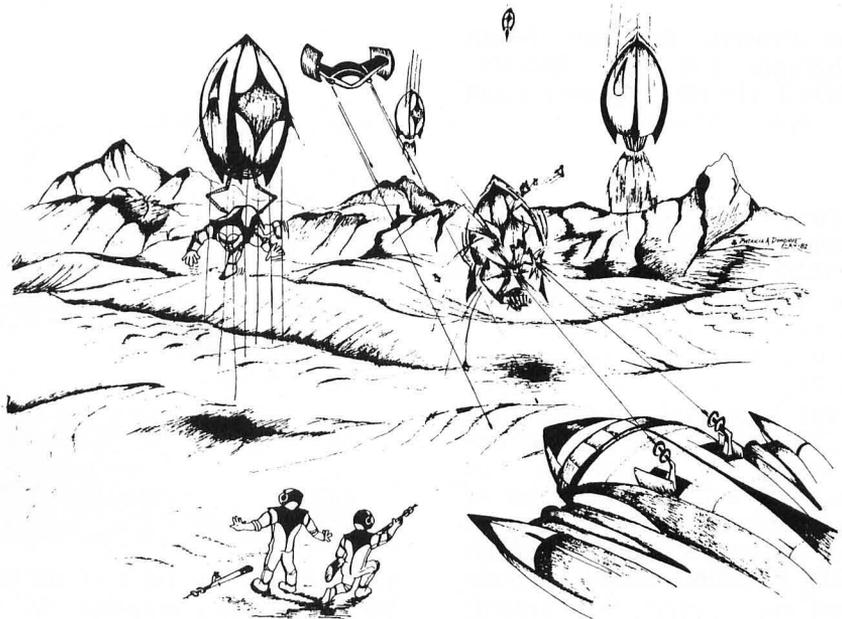
By Larry Ashman (Soft Sector Marketing, Inc., 6250 Middlebelt Road, Garden City, MI 48135.)  
 System requirements: TRS-80® Model I or III with 16K (cassette) or 32K (disk). Suggested retail price: Cassette — \$15.95, Disk — \$19.95.

*Alien Defense* is a fast-paced Machine Language game that plays much like the arcade game, *Defender*. In *Alien Defense*, you thrust across a landscape in search of aliens to obliterate or humans to rescue.

You propel your ship horizontally by thrusting and also control the vertical movement in order to avoid various obstacles. As you move across the landscape, any of six different kinds of aliens may attack you with missiles and bombs, or just try to collide with you. The landscape wraps around so if you pass by an alien, it will appear again should you later pass over the same location.

Each variety of alien has its own special characteristic and style of attack. The "Landers" attempt to land, pick up your humans, and carry them off into space where they become mutants. "Mutants" bounce wildly up and down, making them very difficult to hit. "Bombers" drop rows of stationary bombs which impede your movement across the landscape. "Cruisers" maneuver around your path and, without warning, turn around and attack head-on. They may follow you for a long time, forcing you to stay on full thrust, until you either hit another alien, or try something foolish like reversing direction and firing. This maneuver requires guts, concentration, and a fast firing finger. "Pods," which are the most dangerous aliens, send missiles at you rapid-fire. Blasting the "Pod" won't solve the problem because when hit, it bursts into several fast little "Swarms" that are almost impossible to evade.

As if avoiding aliens weren't enough of a challenge, another duty



of a good defender is to save the humans. The "Landers" continually try to kidnap the humans. Your job, as a defender, is to shoot the "Lander," catch the falling human, and return him to the surface. This is a tricky operation, especially when being tailed by a "Cruiser," but if you succeed, you will be generously rewarded with points. If all fifteen humans are gone, you are transferred to space duty where you face an unbelievable number of aliens.

To succeed in *Alien Defense* (you never win), the most important goal is to keep moving. Stopping or making frequent reversals leaves you vulnerable to being rammed by a "Cruiser" or hit by one of the alien's homing missiles.

The controls used are: 1) — Change Direction, 2) — Move UP, 3) — Move DOWN, 8) — Thrust, 9) — Fire, SPACE BAR) — Smart Bomb, ENTER) — Hyperspace. Operating seven controls skillfully takes practice, but they are conveniently located and designed for maximum efficiency and minimum fatigue.

*Alien Defense* is one of the first arcade games to use Model III

special characters extensively. It uses a combination of graphic characters for the aliens and humans. The disk version includes voice effects in addition to the sound effects which come on the cassette version. Its extensive vocabulary includes the name of the game, author, and publisher during the attract mode, "Coward" when you use hyperspace, and "You're in trouble now!" when all humans are gone. The disk version also saves all of the current high scores onto disk if the user wishes. (Some games save the high score after every game, which necessitates keeping the disk in the drive at all times, increasing the risk of disk damage during a power outage.)

*Alien Defense* is an addictive game. It's never boring. The difficulty, the very fast pace, the different kinds of aliens, and the many control options make it exciting, challenging, and totally engrossing. It goes beyond the "move and shoot" type of arcade game. You'll be playing it for a long time, getting better every time, but never quite mastering it. Anyone who enjoys arcade games can make a sure bet on this one. 5

## NewScript Version 7.0

Reviewed by Harry Temple

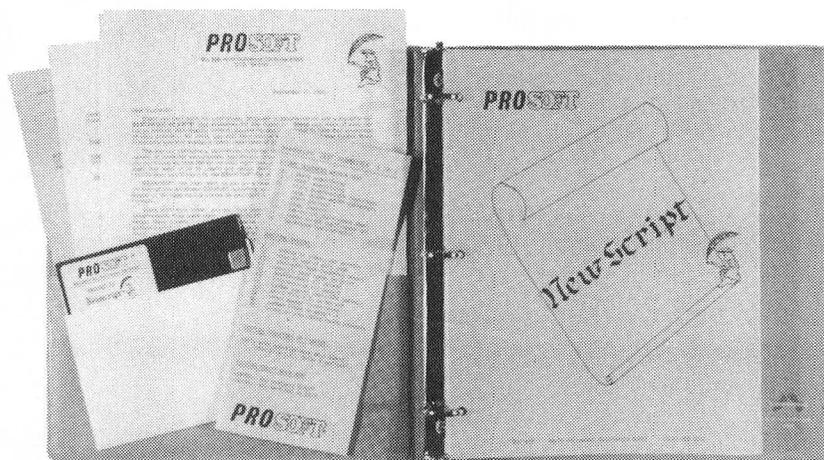
From Prosoft, Box 839, North Hollywood, CA. For TRS-80® Models I and III. Suggested retail price: disk — \$124.95.

*NewScript 6.1* was reviewed by Joseph A. Breton in the April, 1982 *SoftSide*. The following is a review of an updated version of the word processing package.

You probably won't believe what *NewScript 7.0* can do, but you should! This word processor takes the TRS-80® to a new level of text handling. It organizes and remembers indexes and tables of contents; scrolls right or left, up or down, by line or page; has an instantly executing editor; supports almost every printer; does proportional printing, form letters and mass mailings; accepts graphics from the keyboard; has typeahead and printahead buffers, and runs a cassette recorder for dictation. It also inserts and deletes, does global change and replace, runs on a Model I or a Model III without changes, is integrated to a spelling checker and graphics editor, has a useful HELP command, superb documentation and user support. (Pant, pant.)

When you initialize *NewScript 7.0*, you tell it about your printer and disk operating system. *NewScript 7.0* then automatically adjusts itself. It will even load its own serial printer driver. It asks what you want to do: edit, print, run a spelling checker or advanced graphics editor, print mailing labels, check directories, reinstall *NewScript*, or exit to DOS or BASIC. (The spelling checker, advanced graphics editor and mailing labels programs are optional add-on modules.)

If you want to edit, *NewScript 7.0* loads EDIT, a Machine Language program with a BASIC front end. You can't type too fast for EDIT. There is no keybounce. It scrolls up or down, left or right. If you press



CLEAR (the Control key), the cursor changes to a graphics "C". If you press the left or right arrow, the text moves horizontally. If you press "F" or "B" while the "C" cursor is displayed, the text immediately moves forward or backward one page. Forget the special keys needed with other word processors. Want to insert? Type Clear, "I". Delete a character? Type Clear, "D". Delete a word? Type Clear, "W". Editing functions are a combination of the Clear key and one other, easily remembered key.

Format commands have default settings, but you can alter them. *NewScript 7.0* is very flexible. Besides freedom of cursor movement, block moves, global deletion or insertion, forward or backward searching, and automatic wraparound, you can mark entries for an index or table of contents. A utility program lets you enter index keywords and marks your text files.

*NewScript 7.0* uses imbedded control words to format text. For paragraphs, type ".p" at the beginning of a line. To adjust line length, type ".ll", followed by the desired line length.

*NewScript 7.0's* approach to text formatting is logical. Rather than measuring in variable lines and spaces, it measures in standard

inches — a common sense solution to a problem that has troubled other word processors.

*NewScript 7.0* overcomes memory limitations. You can create documents larger than memory by chaining files together, and even link files between disks. EDIT works in memory and stores your text on disk when you press Clear and "S" at the end of the session, or after a specified number of text changes. The text formatter processes text files. If a file gets too big to edit, a utility chops the file into segments and chains them together.

Since *NewScript's* author, Chuck Tesler, has modules to perform various functions (EDIT does editing, SCRIPT does printing), there is room for additional features within each module. When you print, you don't need the editor. If you use the memory the editor occupied, you can write a super text printer.

SCRIPT's video display looks almost like a printed page, so you can check your "print" without using paper. If you revise a lot, do it on video. Then, when your document is perfect, print it. You'll find you'll print text on paper less than with other programs.

*NewScript 7.0* supplies a print routine that shows you what will be



printed, so mistakes can be corrected before printout. If you need to insert text during printing, at the end of printing, or both, *NewScript 7.0* can meet your needs. You can also edit a document on its way to the printer and enter text from the keyboard while printing.

*NewScript 7.0* can number text lines as it prints them, or make a number of copies. It can use sheets or continuous forms. *NewScript 7.0* can triple space. SCRIPT allows you to begin printing at any page of a document. The printer spooler improves throughput. There are optional proportional spacing packages for *Diablo*, *Starwriter*, *Qume* or *Spinwriter* printers. (Some Daisy wheel printers are already supported.)

---

**“It not only tells you what to do when things work right, but gets you out of trouble when they don’t.”**

---

*NewScript 7.0* is very user-friendly. You always have the option of bailing out. If you invoke EDIT when you really want SCRIPT, it's easy to get where you want to be. If you give the wrong filename, *NewScript 7.0* asks for the right one and then straightens things out. If you make a mistake while editing, the “Whoops” command gets you back to square one.

*NewScript 7.0* is great at error trapping. If you inadvertently hit the BREAK key while editing, the screen tells you how to recover. If the system reboots and your text is still in memory, type “BASIC\*” (on most operating systems), press BREAK and type “Go To 999”. Text in memory is then saved under a temporary filename. You use EDIT to recover it. This is just one example of the thought that went into *NewScript 7.0* — it not only tells you what to do when things work right, but gets you out of trouble when they don't.

*NewScript 7.0* takes a little time to learn because it's so powerful. You can do tabs, super and subscripts, and half line feeds in forward or reverse directions. You can hyphenate, alter the proportional spacing method, use double width or tiny characters, italics — almost anything you can think of, providing that your printer allows it. The user's manual is superbly written, and sets a new standard in the industry for documentation. Chuck Tesler also responds quickly to letters for help. When you buy *NewScript 7.0*, you not only get programs and documentation, you get support from the author himself.

In summation, *NewScript 7.0* has the features you want in a word processor: ease of use, error trapping, super printing, a print spooler, superb documentation, and adaptability to many printers and operating systems. *NewScript 7.0* is a standard against which other TRS-80® word processing programs will be judged. ☺

## ATTENTION AUTHORS

**SoftSide Publications** is actively seeking program, article and review submissions for the TRS-80®, Apple™ and ATARI® home computers.

● **Programs** — **SoftSide** has always been the leader in the field of BASIC software. BASIC remains our specialty. However, with the advent of Disk Version (DV), we can now also offer an outlet for Machine Language and multiple language programs which do not lend themselves to printed versions. Games, utilities and educational software, as well as any other applications for the home computer user are preferred, although we will consider virtually any type of program. Hybrid mixes of articles and programs are also welcomed.

Please be sure to include **full** documentation of subroutines and a list of variables, also a brief article describing the program.

● **Reviews** — Well written, informed reviews of all software for the systems we cover are a regular feature of **SoftSide**. Reviewers should take into consideration all aspects of a particular software package, from speed of execution to programming creativity to the estimated length of time that the product will hold the customer's interest.

● **Articles** — We welcome article submissions of all types, but prefer those specifically geared to the home computer market. We give our readers information as a first priority, but vary our content to include some humor and commentary.

All text, including documentation and descriptive articles for programs, should be typewritten and double-spaced. Extra monetary consideration will be given to articles and reviews submitted on machine-readable media (Scriptsit, Super-Text II, etc.). Programs should be submitted on a good cassette or disk. TRS-80® BASIC programs should function under both Level II and Disk BASIC.

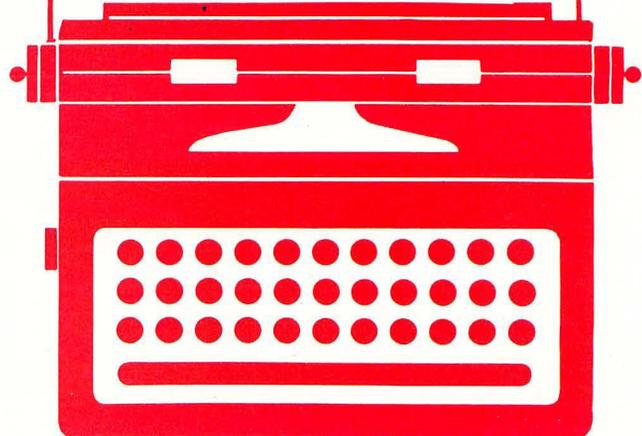
Please be sure to pack your cassettes and disks carefully and to include your return address and phone number.

Send to: **SoftSide Publications**  
**SUBMISSIONS DEPARTMENT**  
6 South Street  
Milford, NH 03055

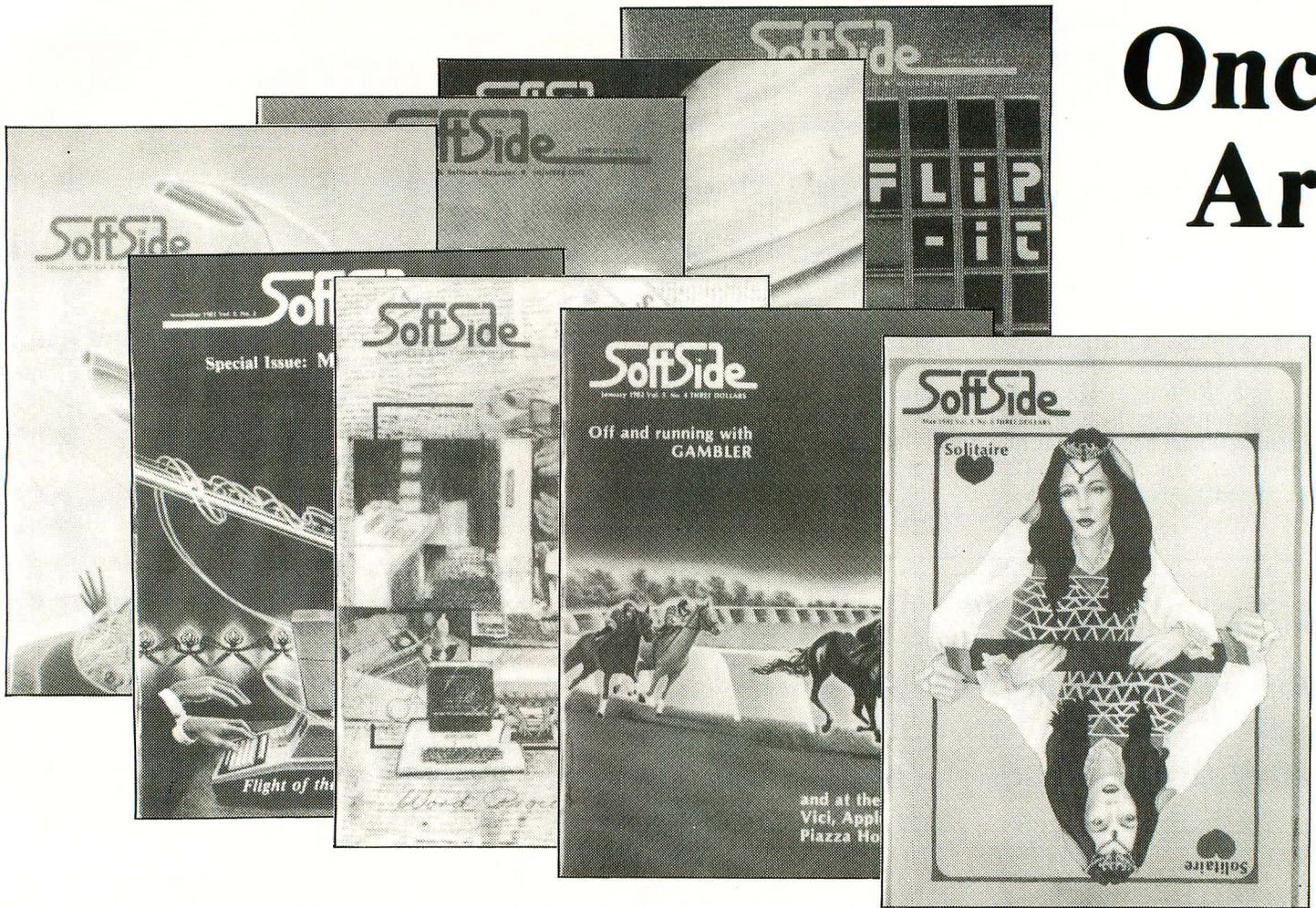
We regret that due to the volume we receive, we are unable to return submissions which do not include return postage.

Be sure to send for our **FREE AUTHOR'S GUIDE**. It further outlines the specifics of our submission procedure.

TRS-80 is a registered trademark of Tandy corporation.



# Once Are



## Back Issues

## from

# SoftSide™

### August 1980

"You Can Have Sound" — Apple  
"Rom the Robot" — Apple  
"Masters Golf" — ATARI®  
"Grand Prix" — TRS-80®

### September 1980

"Developing Data Base I" — All Systems  
"Goal" — Apple  
"Barricade" — ATARI®  
"Concentration II" — TRS-80®

### October 1980

"Developing Data Base II" — All Systems  
"Moonlanding" — Apple  
"World Series" — ATARI®  
"Earth-Port II" — TRS-80®

### November 1980

"Developing Data Base III" — All Systems  
"Collision" — Apple  
"Trench" — ATARI®  
"Kriegspiel" — TRS-80®

### December 1980

"Developing Data Base IV" — All Systems  
"Baseball" — Apple  
"Speedello" — ATARI®  
"Kidnapped" — TRS-80®

# They're Gone... They Gone Forever?

If you like the programs, reviews, and programming information in this issue of *SoftSide*...think of what's waiting for you in past issues!

Exciting games like *Defense...Mini-Golf...Micro Man...*  
Great Graphics like *Old Glory...Titan...Flight of the Bumblebee...*  
PLUS...databases, utility programs, educational programs, and more!

It's all here in the Back Issues of *SoftSide* magazine. And many of these issues are still available for your enjoyment. But not all. Several issues are SOLD OUT...others are available but supply is limited.

So check out the issues and features listed below and order today!

- Back Issues ..... \$3.50 ea.\*
- Back Issue on cassette ..... \$9.95 ea.\*
- Back Issue on disk ..... \$14.95 ea.\*
- Back Issue w/ Enhanced Disk Version  
(contains an additional program) ..... \$19.95 ea.\*

Use our bind-in order card provided elsewhere in this issue to order your Back Issue copies, cassettes, disks and enhanced disk versions of *SoftSide*.

For magazine orders only (minimum 3 issues) send order card and payment to:  
**SoftSide Magazine 100 Pine Street, Holmes, PA 19043**

For magazine plus cassette or disk orders...and enhanced disk orders...send order card and payment to: **SoftSide Magazine, 6 South St., Milford, NH 03055**

\*Prices good for USA orders only — for foreign pricing see page 80 for information.

## January 1981

- "Developing Data Base V"  
All Systems
- "Convoy" — Apple  
and TRS-80®
- "Angle Cannon" — ATARI®
- "Ship Destroyer" — TRS-80®

**Disk and tape only — available**

## February 1981

- "Developing Data Base VI" —  
All Systems
- "Miner" — All Systems
- "Mini-Golf" — ATARI®  
and TRS-80®
- "Long Distance" — TRS-80®

## March 1981

- "Developing Data Base  
VII" — All Systems
- "Strategy Strike" — Apple  
and TRS-80®
- "Flags" — ATARI®
- "Volcano" — TRS-80®

## April 1981

- "Battle At Sea" — Apple
- "Convoy" — ATARI®
- "Dominoes" — TRS-80®

## May 1981

- "Galaxia" — Apple
- "Dodge" — ATARI®
- "Orienteering At Jacque's  
Coulee" — TRS-80®

## June 1981

- "Old Glory" — All Systems
- "Word-Search Puzzle  
Generator" — All Systems
- "Anallist" — TRS-80®

## July 1981

- "Chemistry Drill" — All Systems
- "Volleyball" — Apple
- "Space Lander" — ATARI®
- "Word Wars" — TRS-80®

**Disk and tape only — available**

## August 1981

- "Quest 1" — All Systems
- "Shark" — Apple
- "Dairy Farming" — ATARI®
- "Compu-Sketch" — TRS-80®

## September 1981

- "Flip-It" — All Systems
- "Orienteering at Jacques  
Coulee" — Apple
- "Exterminate" — TRS-80®

**Enhanced Disk Version**  
"NEWBASIC" —  
TRS-80® Mod-I

## October 1981

- "Leyte" — All Systems
- "Developing Data Base" — Apple
- "Character Generator" —  
ATARI®
- "Envyrn™" — TRS-80®

### Enhanced Disk Versions

- "Super Dairy Farming" — Apple
- "Gameplay" — TRS-80®

## November 1981

- "Flight of the Bumblebee" —  
All Systems
- "Music Machine" — Apple
- "Music Programmer" — ATARI®
- "Music Editor" — TRS-80®

### Enhanced Disk Versions

- "National Anthems" — Apple
- "Volleyball" — ATARI®
- "Mean Checkers Machine" —  
TRS-80®

## December 1981

- "Titan" — All Systems
- "Aircraft Commander" — Apple
- "Developing Data  
Base" — ATARI®
- "Electronics  
Assistant" — TRS-80®

### Enhanced Disk Versions

- "Bobsledding" — Apple
- "Survive" — ATARI®
- "Konane" — TRS-80®

## January 1982

- "Gambler" — All Systems
- "Microtext 1.1" — All Systems
- "Apple Capture" — Apple
- "Piazza Hotel" — ATARI®
- "TRS-Man" — TRS-80®

### Enhanced Disk Versions

- "Nuclear Submarine Adventure" —  
Apple, TRS-80®
- "Death Star" — ATARI®

## February 1982

- "Space Rescue" — All Systems
- "Rubcube" — Apple
- "Defense" — ATARI®
- "Maze Sweep" — TRS-80®

### Enhanced Disk Versions

- "Andorra" — Apple
- "Kismet II" — ATARI®
- "Help Package" — TRS-80®

## March 1982

- "Hexapawn" — All Systems
- "Magical Shape Machine" —  
Apple
- "Outer Space Attack" —  
ATARI®
- "Killer Cars" — TRS-80®

### Enhanced Disk Version

- "PEEKER/POKER" — Apple
- "Curse of the Pharaoh" —  
ATARI®
- "Warpath" — TRS-80®

**Disk and tape only — available**

## April 1982

- "Microtext" — All Systems
- "Poster Maker" — Apple
- "Banner Machine" — ATARI®
- "Database" — TRS-80®

### Enhanced Disk Versions

- "Semaphore" — Apple
- "Renumbering for the  
ATARI®" — ATARI®
- "Screen Print" — TRS-80®

## May 1982

- "Solitaire" — All Systems
- "Micro-Man" — Apple
- "Cross Reference" — ATARI®
- "Ladders" — TRS-80®

### Enhanced Disk Versions

- "List Formatter" — Apple
- "Robot Battle" — ATARI®
- "Breakthru" — TRS-80®

## #30

- "Escape from the Dungeons of  
the Gods" — All Systems
- "SWAT" — All Systems
- "Code Breaker" — Apple
- "Sabotage" — ATARI®
- "Piazza Hotel" — TRS-80®

## #31

- "C.A.T.S. Computer Assisted  
Testing System Part I" — All  
Systems
  - "Puzzle Jumble" — Apple
  - "Tunein" — ATARI®
  - "DOTS" — TRS-80®
- Enhanced Disk Versions**  
"Applesoft Extensions" — Apple  
"Menu Plus" — ATARI®  
"Starbase Gunner" — TRS-80



**GRAND OPENING**

**SPECIAL — SUPER SALE**

VAULTS OF CYMARRON  
(100 ROOM GRAPHIC ADVENTURE)

REG PRICE	OUR PRICE	W/THIS AD
39.95	29.95	20.00

	Reg. Price	Our Price
ADVENTURE INTERNATIONAL		
ELIMINATOR	19.95	15.95
STAR SCOUT	14.95	12.25
ADVENTURES 1-12	ea. 19.95	15.95
BIG 5		
SUPER NOVA	15.95	12.95
ATTACK FORCE	15.95	12.95
ROBOT ATTACK	15.95	12.95
MED SYSTEMS		
DEATHMAZE 5000	14.95	12.25
LABYRINTH	14.95	12.25
ASYLUM	19.95	16.00
COMPUTHINGS		
WIZARD'S MOUNTAIN	14.95	12.00
DARK STAR	24.95	17.50
TOWER OF ORLANDOR	19.95	14.50
ROMAN CONQUEST	14.95	12.00

AND MANY MORE  
SEND FOR FREE CATALOG/PRICE LIST

470 CASTRO SUITE #207-#3359  
SAN FRANCISCO CA 94114  
415-861-8966



**Oasis West**

**OUT OF THE BLUE**

The **SoftSide** family is growing. We will soon be covering the IBM Personal Computer and we cordially invite PC people everywhere to join the ranks of **SoftSide** contributors. If you have a program, a review, or just a story to tell, let us have a look at it. Send submissions to:

**SoftSide Publications, Inc.**  
Department Z  
6 South Street  
Milford, NH 03055

**Continental Adventures**

for ATARI® Computer Owners

Each adventure takes up to an hour and is programmed with random variables to change the game every time you play.

● **THE TALISMAN OF POWER**  
A search through many obstacles for the Four Keys of Gremlock. Will you get out alive?  
16K-\$17.95

● **THE GHOST TOWER**  
Find a Magical Gem in a tower haunted with Orcs, goblins, etc. 16K-\$16.95.

● **SUPER SHAPE BUILDER**  
A graphics game mainly for children from age five. "Draw" your own pictures with your joystick, in color. 8K-\$9.95.

Send Check or C.O.D. — Discs \$4.00 extra

**SATISFACTION GUARANTEED**

Send for Complete Software List

**DEALER INQUIRIES INVITED**  
CONTINENTAL ADVENTURES, 4975 Brookdale  
Bloomfield Hills, MI 48013 (313) 645-2140

**APPLE PHYSICS**

11 Disks - 75 Programs - \$203

These programs contain extensive graphics. Each diskette has 5 to 10 programs requiring 48K memory with Applesoft.

V 1 Vectors & Graphing	\$10.00
V 2 Statics	\$12.00
V 3 Motion	\$12.00
V 4 Conservation Laws	\$12.00
V 5 Circular Motion	\$15.00
V 6 Thermodynamics	\$20.00
V 7 Electricity	\$12.00
V 8 Optics	\$20.00
V 9 Atomic Physics	\$30.00
V10 Solar System Astronomy	\$30.00
V11 Stellar Astronomy	\$30.00

Ask for Atari, IBM information

**AQUARIUM:** This is an aquarium simulation in which the fish swim, breed, eat and interact. The full disk includes the Community Aquarium plus 4 games. 48K \$25.

**DINOSAURS:** 6 games and demos. Includes Dinosaur Matching, Dinosaur Hangman and Paddle Graphics. 48K \$15.

**CROSS EDUCATIONAL SOFTWARE**

P. O. Box 1536  
Ruston, LA 71270  
318-255-8921



Write today for a FREE Catalog.

**THE KWIK CURE**

For S-80 Cassette Blues!



**KWIKOS:** (Model I, Level II, 4k-48k)  
A PROGRAMMER'S PROGRAM... FOR NOVICE OR EXPERT, NOT JUST A SIMPLE SPEED-UP (Select 2x to 6x)... BUT ALSO MANY EASY-TO-USE ENHANCEMENTS TO SUPPORT TAPING, FEATURES: SAVE, LOAD, VERIFY, SEARCH, CHAIN LOAD, CATALOG, AND TEST-READ OF BOTH 'BASIC' AND 'SYSTEM' PROGRAMS, + PASSWORDS + LONG TITLES + DEBOUNCE + SELF 'BACKUP', and MORE... \$24 ppd US

**KOSS:** ("Kwikos" for Model III, 4k-48k)  
ALL kwikos FEATURES AT 2200 BAUD, + KWIK SET OF: CASB HI/LO, I/O ROUTING, TIME, DATE, BREAK KEY, LIST SCROLL SPEED, etc..... \$24 ppd US

**KWIKKIT:** (specify Model I or Model III)  
mini-SYSTEM FOR "BASIC" ONLY. EASY-LOAD 1000 BAUD FOR MOD I, 2200 BAUD FOR MOD III. MANY kwikos FEATURES AND UP-COMPATIBLE TO FULL SYSTEM... (add \$2 for debounce)... \$50 ppd US

**KWINK:** Model I 'system' DUPLICATOR PLUS!  
MAKES STAND-ALONE FAST-LOADING (2x to 6x) kwik COPY OF ANY MOD I 'SYSTEM' PROGRAM. (3 min tape in 44 sec.)..... \$12 ppd US

**KLOAN:** Model III 'system' DUPLICATOR PLUS!  
MAKES STANDARD 500 OR 1500 BAUD COPY OF ANY 500/1500 BAUD 'SYSTEM' PROGRAM... \$10 ppd US

(...COMING SOON...)  
**KLOAN:** Model I 'basic' DUPLICATOR PLUS!  
MAKES STAND-ALONE FAST-LOADING COPY OF ANY 'basic' PROGRAM. DOES FOR 'basic' PROGRAMS WHAT KWINK DOES FOR 'system'..... \$12 ppd US



**KWIK SOFTWARE**

P. O. Box 328  
Bolivar, Mo. 65613-0328

Phone (417) 326-7154

**WANTED!**

TRS-80 — APPLE — ATARI — VIC

- ★ PROGRAMMERS  
20% Royalties paid monthly
- ★ SALES REPS  
High Commissions on Sales
- ★ DEALERS WANTED  
contact: FRANK BOTTA

**Gompu-Things**

SUITE #2  
270 BROADWAY  
REVERE, MA 02151  
(617) 289-8866

**BUY! SELL! TRADE!**

COMPUTER & HAM EQUIPMENT

**COMPUTER® TRADER**

**PERMANENT SUBSCRIPTION \$10.00**

Low Ad Rates — Mailed Monthly

**COMPUTER TRADER®**

Chet Lambert, W4WDR  
1704 Sam Drive • Birmingham, AL 35235  
(205) 854-0271

Please include your Name, Address, Call Sign or Phone Number

**Two Of The Best Programs From SoftSide**

**Quest 1** An exciting journey into an underground maze in search of treasure and adventure. Armed only with sword and bow, you must rely on quick thinking to survive.

**Just \$4.95**

**Flip-It** is an excellent implementation of the board game Othello™. Match wits with a formidable opponent: your computer.

**Just \$4.95**

MAIL YOUR ORDER TODAY TO:

**SoftSide, 6 South Street  
Milford, NH 03055**



**ARE YOU INTO SPORTS**

DO YOU... Jog? Swim? Walk? Cycle? Run? Play Racquet Sports?  
The **Aerobics Master** for the Apple® is for you. A full year exercise diary that computes the improvements and values of your exercises day-by-day, all year long.

**\$24.95** plus \$1.00 postage and handling  
Mich residents add 4% sales tax.

**FREE LANCE INK**

1806 Wickham • Royal Oak, MI 48073  
REQUIRES APPLE II WITH AppleSoft in ROM • 48K Memory  
1 or 2 Disk Drives • dos 3.5

Apple is a registered trademark of Apple Computer, Inc.

**TAZUMI SOFTWARE  
INTERNATIONAL, INC.**  
8 North Grosvenor Avenue  
Burnaby, B.C., CANADA  
V5B 1J2



*Horse Racing Classic* is a graphics horse race simulation game for a 48K Apple II™ (DOS 3.3). It allows up to nine bettors to participate in a day at the race track. Each bettor has access to information on the seven horses entered in each race, allowing him to expertly choose his horse. The bets are placed and the race begins, with the Thoroughbreds traversing an oval track, constantly jockeying for the best positions. The payoffs are computed, distributed to the winners, and graphically displayed to show player standings...then the next race begins.

*Horse Racing Classic* is available on disk at a retail price of \$36.95.

**KWIK SOFTWARE**  
P.O. Box 328  
Bolivar, MO 65613



*KWICOS*, for the TRS-80® Model I, Level II with 4 to 48K of memory, is a machine-code program that adds disk-like commands to Level II BASIC. This provides a new and improved Cassette Operating System to supplement the standard CSAVE/CLOAD/SYSTEM routines. It is a software-only enhancement, and requires no hardware modifications or add-ons. All operations are effected by simple commands in the Level II "Immediate" mode. With all its features, it is only about 1700 bytes long, while a KILL reduces the length to 1150 bytes.

This system was initially developed to allow 1000 baud operation even with the Model I XRX CPU modification active, and at this speed (2x), the volume range is extremely broad. Without the XRX modification, however, even higher baud transfer rates can be used. Advantages of programs taped by *KWICOS* are:

1) Selectable data-rate (baud) for cassette storage: 2x-6x.

2) A graphics display which guides you to the correct volume setting for loading each program.

3) Machine code programs can be saved and loaded as easily as BASIC programs.

4) The length of each BASIC program is automatically inserted in its Fileheading.

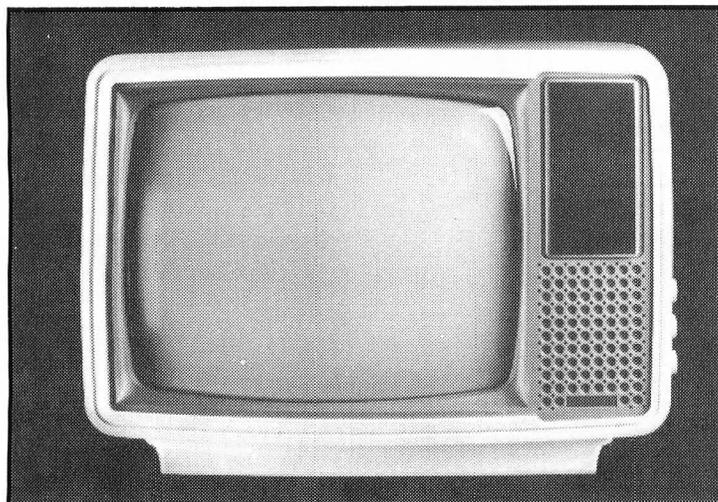
5) "Catalog" feature reads and displays the Fileheading for each KWIK program on a tape.

*KOS3*, for the TRS-80® Model III, Level II, 4K to 48K, has the same features as *KWICOS* except:

1) Transfer rate is fixed at 2250 baud for KWIK functions.

2) Simple control of: BREAK key activation/deactivation; Cassette high/low set; Input/Output routing and initialization; Time set; Date set; Clock display on/off.

Offered as the reasonable alternatives to disk, *KWICOS* and *KOS3* are available on cassette from KWIK Software for \$24.00 postage paid.

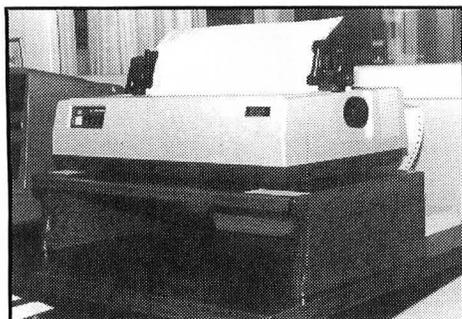


## LEADING EDGE PRODUCTS

225 Turnpike Street  
Canton, MA 02021  
(800)343-6833

*Mean Green* is a new, 12 inch, green screen monitor which economically solves the problem of television tie-up for the at-home computer user. Also an ideal monitor for small businesses, it is a 13 pound CRT boasting a composite video signal and a display format of 1920 characters (80 characters x 24 lines). It measures 40 cm wide, 28.5 cm high and 32 cm deep.

*Mean Green* carries a one year, no questions asked return/replacement plan should the unit fail. It is offered at a very reasonable suggested retail price of \$99.00.



**OAK KIT HARDWARE**  
8689 North 63rd Street  
Brown Deer, WI 53223  
(414)354-8516

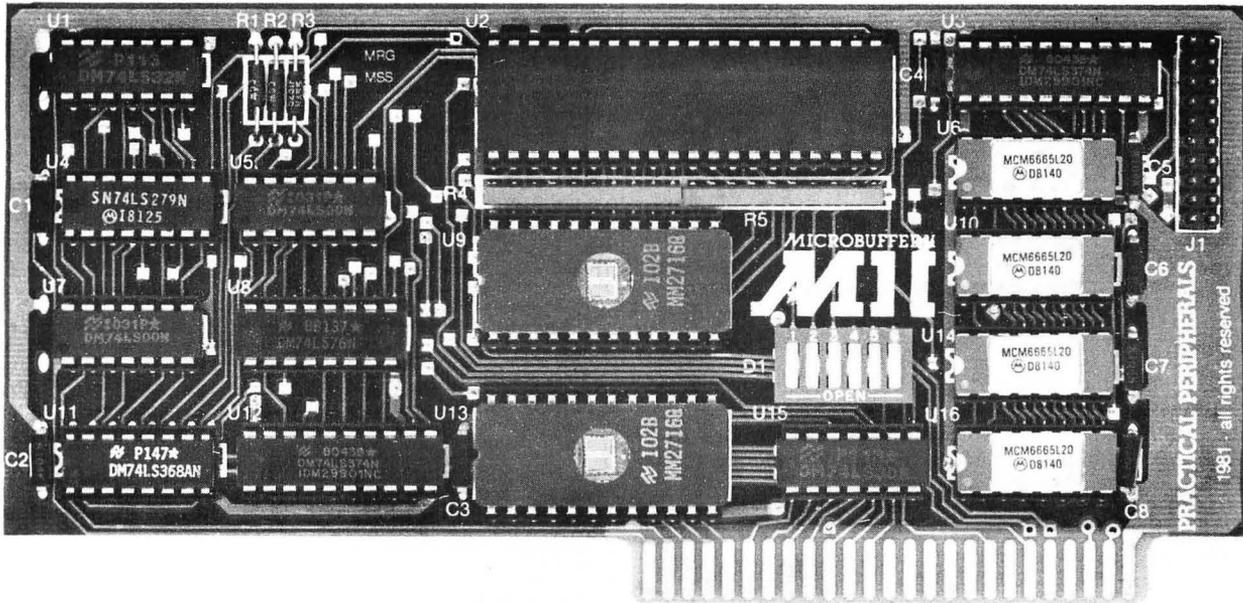
The *Oak Printer Stand* is for those who enjoy the natural beauty of oak and a fun, rewarding challenge at low kit prices. It also offers the practical benefits of organizing continuous paper below your printer, saving space and bringing peripherals within easy reach.

All the kit's parts come pre-cut and pre-shaped, and the stand may be

stained or finished "natural." (Stain and varnish are not included with the kit.) Instructions and hardware are included.

The stand kit comes in two sizes. The Standard kit comfortably holds MX-80, Okidata 82A, etc. size printers and its generous width also allows use as a TV/monitor stand. The King kit holds MX-100, Okidata 83A, etc. size printers with ease.

Price of the Standard kit is \$27.95; the King kit is \$31.95. When ordering either kit, include an additional \$3 for shipping and handling.



**PRACTICAL PERIPHERALS, INC.**  
 31245 La Baya Drive  
 Westlake Village, CA 91362  
 (213)991-8200



The *Microbuffer II™* is an intelligent Centronics-compatible parallel interface for the Apple II™ with up to 32K of on-board RAM for data buffering as well as on-board firmware for text formatting. For graphics type printers, the firmware

also provides advanced graphics dump routines.

The buffering capability of the *Microbuffer II™* increases your data processing efficiency by freeing you from the wait normally experienced while printing. With *Microbuffer II™*, you can print and process simultaneously. It accepts data as fast as your Apple™ can send it, returning control of the computer to you while it handles the printing. Additional output may be dumped to the buffer without waiting

for the first to be completed. Use of *Microbuffer II™* breaks the computer-waiting-for-printer/printer-waiting-for-computer bottleneck.

*Microbuffer II™* is compatible with all parallel printers that use the Centronics-type interface. The full buffering capability and text formatting functions are available for all compatible printers; graphics functions are supported for certain hi-resolution graphics printers. Suggested retail price for the 32K is \$299.00; for the 16K it is \$259.00.



**ATARI® HOME COMPUTER DIVISION**  
 P.O. Box 427  
 Sunnyvale, CA 94086



*My First Alphabet* gives pre-school children an enchanting introduction to letters and numbers. The touch of a key summons any one of 36 high-resolution graphics representing a particular character. *My First Alphabet* uses association and pattern recognition to help children master objects, songs and shapes. This approach sparks children's imagination and gives them basic tools for the all-important process of conceptual learning.

*My First Alphabet* requires an ATARI® Home Computer with 32K of memory, one disk drive, and an ATARI® BASIC cartridge. Suggested retail price — \$29.95.

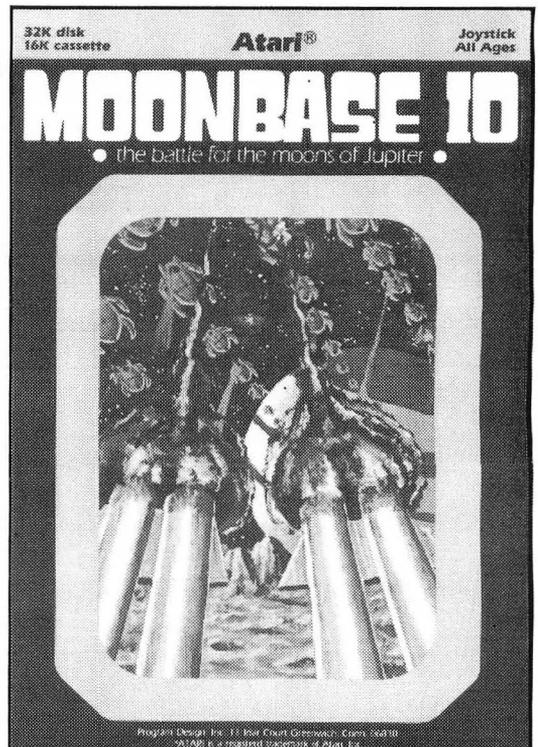
**PROGRAM DESIGN, INC.**  
 11 Idar Court  
 Greenwich, CT 06830  
 (203)661-8799



*Moonbase IO*, available for use on ATARI® 400/800 computers with memory of at least 24K, is the first game on the market to combine arcade and adventure formats with voice narration. It promises to challenge even the most experienced arcader.

Someday, according to *Moonbase IO*, the solar system will be governed by the Earth Federation. The Federation will have bases on IO and two other Jovian moons, Europa and Ganymede. As *Moonbase IO* begins, aliens have entered the solar system and have destroyed several ships from Earth. Earthbase Control asks you, the player, to secure the three Jovian bases.

The game consists of a voice cassette and a disk. It is available at computer stores, or directly from Program Design, at \$29.95. A cassette version will be available at a later date.





# CRYPT OF THE UNDEAD

**SUDDENLY** you awaken in a coffin surrounded by a vast graveyard. **YOU MUST GET OUT BEFORE DAWN!**

**HOW?** There's no easy way, the walls are too high—the gates are locked. Vampires, Zombies, and Werewolves constantly attack as you frantically search musty crypts, catacombs, and dead tombs for treasures and clues. **ONLY TWELVE HOURS TO ESCAPE OR REMAIN FOREVER!**

CRYPT OF THE UNDEAD, designed by Marc Benioff, Offers...

- Both Puzzle-Solving & Role-Playing Intrigue!
- Superb Graphics, Sound & Color Animation!
- Hours of Challenging Terror!

Requires...

- ATARI 400/800 & One Disk Drive
- One Player & Joystick Controller

Comes with...

- Game Program & Complete Instructions
- EPYX 30/FOREVER WARRANTY

Now Available At Your Favorite Dealer ... \$29.95  
For the name of your nearest EPYX dealer write:  
"CRYPT OF THE UNDEAD"

EPYX/Automated Simulations, Inc.  
P.O. Box 4247, Mountain View, CA 94040

**EPYX**  
COMPUTER GAMES  
THINKERS PLAY

For the ATARI\*  
A New EPYX  
Graphic-Adventure  
from Automated Simulations

EPYX Temple of Apshai was the very first computer game ever to win the Hobby Industry award for excellence. EPYX pledges you that same excellence in every game you purchase from us ... the VERY BEST in entertainment!

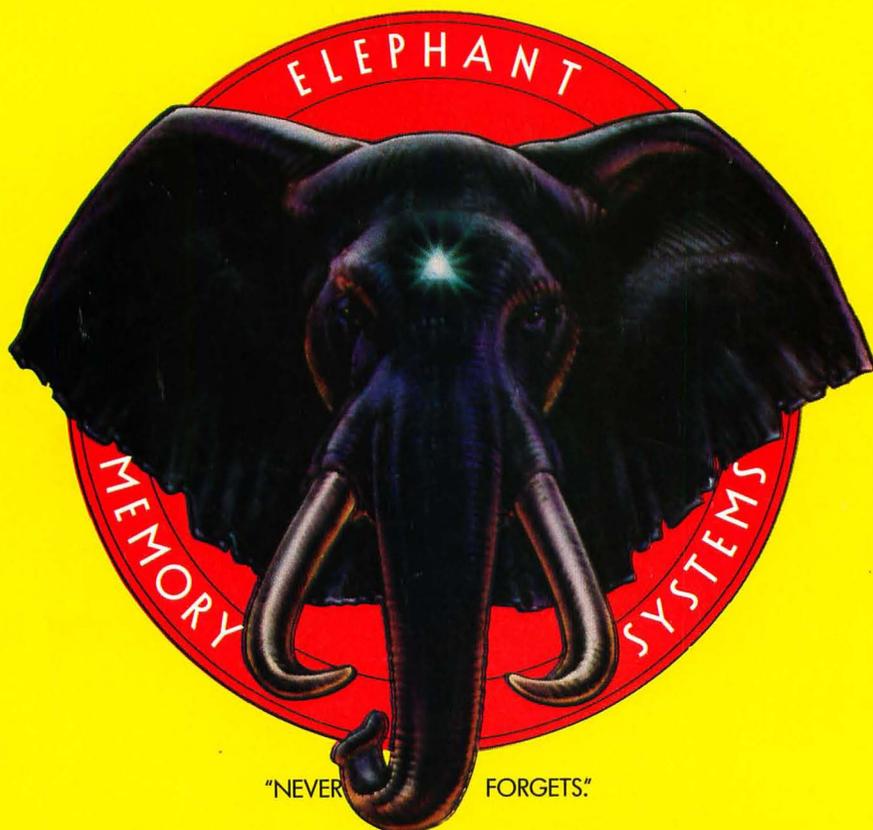
#### EPYX 30/FOREVER WARRANTY

\*Our 30-day Unconditional Guarantee: If your EPYX Game has any defect whatsoever within 30 days of purchase, return it to us or your dealer and we will replace it free.

\*Our Forever Warranty: If anything happens to your disk at any time after 30 days, for any reason, just send it back with \$5.00 for shipping, and we will send you a replacement.

\*ATARI 400/800 is a trademark of ATARI, INC.

# REMEMBER:



## MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to

make life miserable for everyone in the disk-making business.

How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

## ELEPHANT.™ HEAVY DUTY DISKS.

For a free poster-size portrait of our powerful pachyderm, please write us.

Distributed Exclusively by Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021

Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.