A Publication of Compupress Inc. • May/June 1984

$2.95

Programs You Can Type

Hi-Res MAGAZINE™

Specifically for Atari and Commodore Users

DELL 04316

Painless Computing: Step-by-Step Guide

The 264: Right Stuff at the Right Time?

Voyage to the Outer Reaches of the Universe

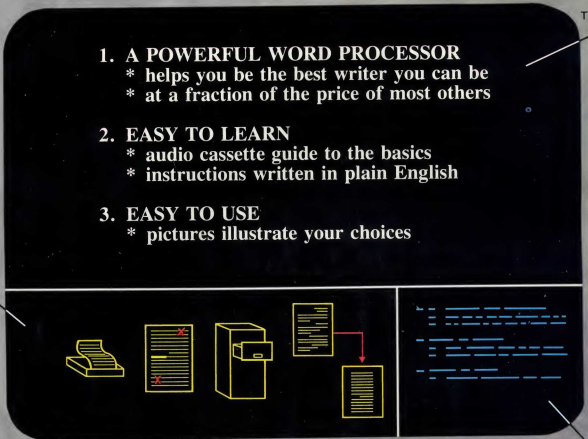
Plus: Hardware, Software Reviews

'84 Beholds Cautious Optimism for Industry Giants

Introducing the first word processor made especially for the home

HOMEWORD™

The Personal Word Processor



1. A POWERFUL WORD PROCESSOR
   * helps you be the best writer you can be
   * at a fraction of the price of most others
2. EASY TO LEARN
   * audio cassette guide to the basics
   * instructions written in plain English
3. EASY TO USE
   * pictures illustrate your choices

HomeWord makes writing easy. Changing what you write is even easier. Since the commands are illustrated right on the screen, you don't have to memorize any complex codes.

Yet HomeWord's simplicity is deceptive. Although easier to use, it offers you the power of more expensive word processors!

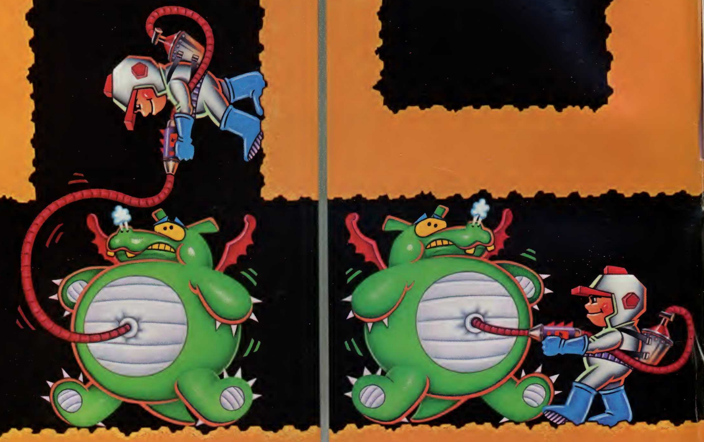
Yes, HomeWord is the best buy for your money. . .and the best money can buy! HomeWord is available for only $69.95 on the Apple II, II+, IIe and Commodore 64. Coming soon on the Atari!

Features Include

* add, move and erase blocks of text
* automatic outline indents
* underline, boldface, upper and lower case
* reliable storage and retrieval of all your files
* automatic page numbering
* print documents of unlimited length
* universal search and replace
* easy view of movement through your files

Sierra ON-LINE TM

Sierra On-Line Building • Coarsegold, CA 93614 • (209) 683-6858



Which is the best way to inflate your score?

Better find out. We've done our groundwork on DIG DUG so well, it plays just like it does in the arcade. So don't dig yourself a hole. Like the player on the left. Sure he'll score points for blowing up the Fygar™ in a vertical path. But blow up the Fygar in a horizontal path, like the player on the right, and score twice as many points. A landslide.

You can dig up even more points by uncovering a bonus veggie. But you get only one on each level. All you have to do is drop two boulders and you'll see it. Buried treasure disguised as a veggie.

Only Atari makes DIG DUG for the ATARI® 2600™ Game, Sears Video Arcade† systems, and versions exclusively for the ATARI 5200™ Super-System and all ATARI Home Computers.

So get to your nearest store and dig into your pockets. For DIG DUG.

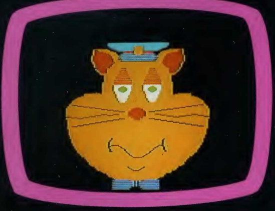
Here comes Dig Dug™* from Atari.®

A Warner Communications Company

†Trademark of Sears, Roebuck and Co. © 1983 Atari, Inc. All rights reserved.

*DIG DUG is created and designed by Namco Ltd., manufactured under license by Atari, Inc. Trademark and © Namco 1982.

SOFTWARE MOVIES™



NEW!

SOFTWARE DESIGNS FOR DEVELOPING YOUNG MINDS™

MAXIMUS, Inc. is excited to present two new educational software movies for kids of all ages . . . and more are on the way!

STORYLINE™ makes bedtime, or anytime, a warm and friendly funtime. Clover the Clown is your tourguide for two fairytales: The Ugly Duckling and Rumpelstiltskin. Through computer magic you become a part of each story!

SAFETYLINE™ combines fun with important lessons about safety. Max the Cat shows you how to cross the street safely going to school and what to do when lost at the zoo.

Both software movies have interactive games that reinforce the fun and learning. Both have real human voices; you don't need a voice synthesizer!

Features of the software include:

- Sophisticated Face Animation
- Lip-Synchronized Voices
- Original Musical Scores
- Superior Full-Color Graphics
- Optional Joystick Control
- Machine Language Programs

Relax and enjoy yourself. Let Max, Clover, and their colorful friends guide you through a new world of fun and learning. You'll be amazed at how real they are!

Four games are included with each Software Movie™. STORYLINE™ and SAFETYLINE™ currently operate on any Atari computer with 48K, a disk drive, and a program recorder. Call toll-free 1-800-368-2152 for our free brochure or for the dealer nearest you. Retail price $39.95.

MAXIMUS, Inc. 6723 Whittier Avenue, McLean, VA 22101

ORDER TOLL-FREE 1-800-368-2152

MAXIMUS ***...Where Excellence is the Standard***

Commodore 64 Version Coming Soon!

Contents

Hi-Res Magazine May/June 1984 Vol. 1 No. 4

Cover Story

SHAKEOUT 20
by T. Lee Kidwell

What does the crystal ball hold for the home computer arena? Many are called — few survive. The tournament continues!

Features

264 22
by T. Lee Kidwell

The most comprehensive article to date on the elusive 264: enter the decade of productivity with a businesslike attitude.

Music Theory 26
by Duane Tutaj

Measure your skill, note your progress and rest assured that you know the score in this latest test on music theory drills.

Font Editor 33
by Steven Reed

Discover how to change the video character generator of your Commodore 64 to display fun fonts and custom characters.

Lords and Ladys 40
by Steven Reed

Conservation and cunning are crucial if you're to rule your own kingdom in this stimulating game of medieval economics and strategy.

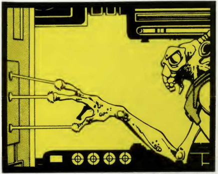
Video Art 76
by Sol Guber

Two articles on computer art that will transform your monitor into a video canvas bounded only by your imagination.

20

22

76



86

Truehart 84
by Léo G. Laporte

A beautiful blonde and hairy cannibals? An installment called "Kind Hearts and Carrot Juice?" Does this compute? Answer Y or N.

Space Knights 86
by David L. Heller

Hold on tight as you rocket into the future with Jake, Lisa and CLIDE to battle deadly Gammalon pirates. First of a series.

Departments

Editor's Letter 6
Publisher's Letter 9
Backspace 10
Perspectives 12
Painless Computing 47
Cryptics 52
The Family Place 54
Academic A 56
Commodore On-Line 58
Strolling Forth 60
Some Assembly Required 62

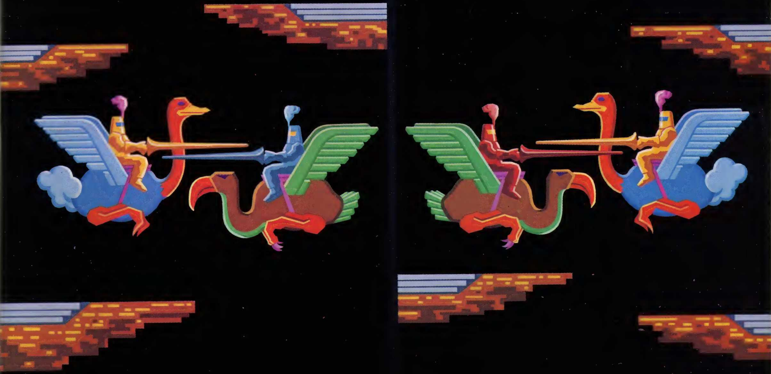
Cover illustration:
Melissa Slimick

As an added convenience with HI-RES' new format, Atari and Commodore logos have been provided to designate coverage on certain pages.

HI-RES Magazine (ISSN 0742-6747) is published bi-monthly by Compupress Inc., 208 West Canton Ave., Suite 310, Winter Park, Fla. 32789. HI-RES Magazine is a trademark of Compupress Inc. The contents of HI-RES are copyrighted by Compupress Inc. Vol. 1, No. 4, May/June 1984. All rights reserved. Advertisers call 305/629-4567. HI-RES is distributed in the United States and internationally by Dell Distributing Co., 245 East 47th St., New York, N.Y. 10017. Subscriptions are $24 per year in the United States, $34 in Canada and $44 foreign. All subscriptions should be sent to HI-RES.

HI-RES welcomes submissions from programmers and authors. When submitting manuscripts, double-space copy and submit a disk containing program. Include a self-addressed stamped envelope for their return. HI-RES assumes no responsibility for unsolicited material that is not accompanied by adequate return postage. Address material in care of the editor.

All letters are considered the property of HI-RES and unconditionally submitted for publication. Atari, Atari Home Computer, Video Computer System (VCS) are trademarks of Atari Inc. Commodore, Commodore Computers, VIC 20, Commodore 64, are trademarks of Commodore Electronics Ltd.



Which player is winging his way to victory?

When Joust* flies into your living room you might think you're in the arcade. Because, just like the arcade, this Joust is a duel to the finish. But remember that some jousts are worth more than others. The knight on the right, for example, is about to score 500 points for lancing a red Bounder. But the knight on the left will score three times as many for skewering a blue Shadow Lord. That ought to needle his opponent.

You can spear yourself a lot more points. Especially in the Egg Wave. Pick up as many eggs as you can before they hatch. The first egg's worth 250 points, the second 500, the third 750, and the fourth 1,000 points.

Only Atari makes Joust for the ATARI® 2600™ Game, Sears Video Arcade† systems, and versions exclusively for the ATARI 5200™ SuperSystem and all ATARI Home Computers.

So mount your ostrich and descend on your nearest store. For Joust.

ATARI®

A Warner Communications Company

HI-RES Magazine

Publisher: Anthony J. Nicholson

Editor: John Babinchak II
Assistant to the Editor: Denise R. Snyder
Art/Production Director: Lee Silbert

Advertising Manager: Craig L. Malcolm
Publisher's Representative: David L. Heller

Circulation Manager: Andrew C. Jones
Editorial/Circulation Assistant: Remora Di Salvo
National Sales Consultant: Sam Taylor

Assistant to the Publisher: Teri L. Turner

Technical Editor: T. Lee Kidwell
Technical Advisers: Jim Ponce, Stephen Reed, Russ Wetmore

Contributing Writers: Bill Haslacher, David L. Heller, Dorothy Heller, Pat Henderson, Leo Laporte, Vern L. Mastel, Robert A. Peck, Lloyd R. Prentice, Stephen Reed, Evan Rosen & Steve McGuire, Pam Sexton, Sallie Stephenson, Duane Tutaj, Russ Wetmore, Jim Wooding

Illustrators: Daijo Akutagawa, Billy Davis, Eric Kleinsmith, David Mitchell, Bruce Sconzo, Dick Thiot

Typographer: Typo-Graphics, Orlando, Fla.
Lithographer: Diversified Graphics, Orlando, Fla.
Printer: Foote & Davies/Mid-America, Lincoln, Neb.
Distributor: Dell Distributing Co., New York, N.Y.

HI-RES Magazine is a division of Compupress Inc.
Offices at 280 W. Canton Ave., Suite 310
Winter Park, Fla. 32789
305/629-4567

Compupress Inc.
Chairman of the Board: Anthony J. Nicholson

From the Editor

Editor:

I subscribed with the understanding HI-RES was to be Atari only. Now you are adding Commodore. Don't bother sending a subscription renewal.

— R. B. Gordon,
Everett, Wash.

Dear R.B. Gordon:

While I am sincerely sorry we have lost a bit of your respect, I must say we continue to forge ahead with great pride and excitement over the addition of Commodore to the HI-RES formula.

But this is not to say we intend to slight Atari in the least.

I am sure you will agree the past few months have shown a staggering evolution in the home computer arena. Once prominent software and hardware giants can't even meet their bills; others have found a happy road to prosperity.

Enter Atari and Commodore.

Commodore enjoys success unmatched in the industry while Atari's future is best described as debatable. One thing certainly is apparent to us at HI-RES Magazine — Atari is, and will continue to be, a powerful force in the microcomputer industry.

The competition between these two is fierce. Commodore's 64 beyond a doubt is a wonder machine selling like — well — like Atari is selling the 800.

Atari's XL series is "selling well," according to industry analysts, and the company's new 1450XLD is due for release very soon. Commodore's answer? The 264 series.

The yet-to-be-released 264 certainly will hail one of the biggest changes to the microcomputer field. The aim of this new machine is not the video game junkie, but word processing and home/small business finance, along with a multitude of other duties.

About our editorial direction:

HI-RES will be highlighting Atari and Commodore line listings while we evolve from game orientation to applications, software and hardware reviews, and how the home computer touches all of us in everyday life.

Granted, the emphasis of the $500 and under computer market is on game software. So we won't abandon coverage entirely — but we will cater to a more mature audience and seek to increase our page count accordingly.

We will not, however, allow our magazine to be diluted to the point of "general interest."

Sincerely,

John Babinchak II
Editor

Turn your Atari into a Ferrari.

Introducing the all-new 1984 Indus GT™ disk drive. The most advanced, most complete, most handsome disk drive in the world.

A flick of its "Power" switch can turn your Atari into a Ferrari.

Looks like a Ferrari.

The Indus GT is only 2.65" high. But under its front-loading front end is slimline engineering with a distinctive European-Gran flair.

Touch its LED-lit CommandPost™ function control AccuTouch™ buttons. Marvel at how responsive it makes every Atari home computer.

Drives like a Rolls.

Nestled into its soundproofed chassis is the quietest and most powerful disk drive power system money can buy. At top speed, it's virtually unhearable. Whisper quiet.

Flat out, the GT will drive your Atari track-to-track 0-39 in less than one second. And when you shift into SynchroMesh DataTransfer,™ you'll increase your Atari's baud rate an incredible 400%. (Faster than any other Atari system drive.)

And, included as standard equipment, each comes with the exclusive GT DrivingSystem™ of software programs. World-class word processing is a breeze with the GT Estate WordProcessor.™ And your dealer will describe the two additional programs that allow GT owners to accelerate their computer driving skills.

Also, the 1984 Indus GT is covered with the GT PortaCase.™ A stylish case that conveniently doubles as a 80-disk storage file.

Parks like a Beetle.

The GT's small, sleek, condensed size makes it easy to park.

And its low $449 price makes it easy to buy.

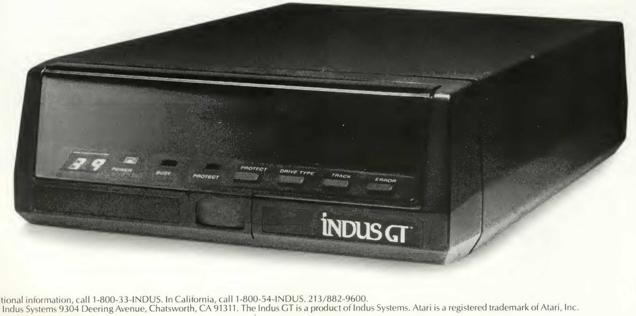
So see and test drive the incredible new 1984 Indus GT at your nearest computer dealer soon.

The drive will be well worth it.

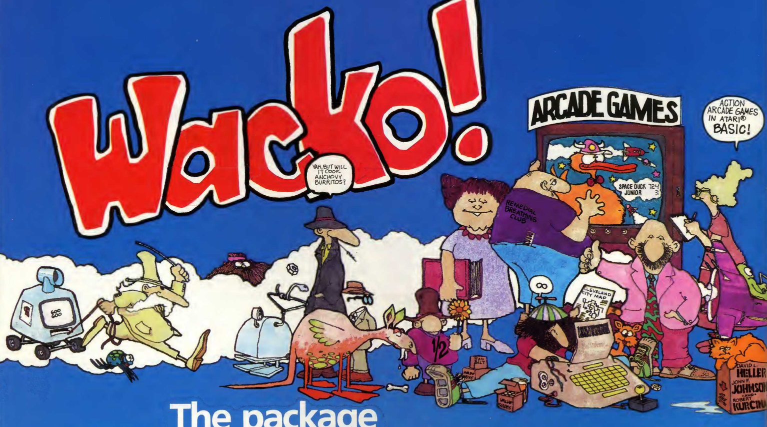
INDUS™

The all-new 1984 Indus GT Disk Drive.

The most advanced, most handsome disk drive in the world.



For additional information, call 1-800-33-INDUS. In California, call 1-800-54-INDUS. 213/882-9600.


Wacko!
YAH, BUT WILL IT COOK ANCHOVY BURRITOS?
ARCADE GAMES
ACTION ARCADE GAMES IN ATARI® BASIC!
SPACE DUCK 724 JUNIOR 3
REMEDIAL BREATHING CLUB
CLEVELAND CITY MAP
1/2
DAVID L. HELLER JOHN F. JOHNSON •AND• ROBERT KURCINA

The package every Atari gamer has been waiting for!

DR. C. WACKO'S MIRACLE GUIDE TO DESIGNING AND PROGRAMMING YOUR OWN ATARI ARCADE GAMES.

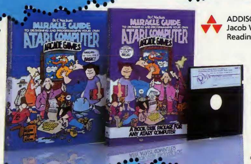
Voyage through Dr. Wacko's action-packed universe of Atari arcade game programming as the master gamesman himself unveils the little-known secrets of Atari BASIC game design. You can create games that sizzle! Games that ZAP . . . POW . . . BZZZZRK!

Now you can be an expert at combining all the elements necessary for your own sure-fire, super-snazzy games: graphics, animation, playing fields, colors, sounds, plots, characters, and unnamed "Things that Bounce in the Night." The 5¼" disk sold with the book saves you hours of typing in Wacko's Monster Maker, Animation Machine, and other ready-to-go programs.

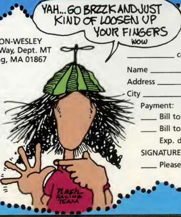
Dr. Wacko and his motley gang of live-in cartoon characters (created by David Heller, John Johnson, and Robert Kurcina) explain everything Atari BASIC programmers need to design games on *any* Atari computer. But wait . . .! Atari owners *without* disk drives need not worry—DR. WACKO is also available in book-only form (you'll just have to type in the programs yourself!).

Sounds good? You can run out now and buy DR. WACKO at your local bookstore or computer store—or send in this coupon today!

Addison-Wesley
Reading, Massachusetts 01867


Dr. C. Wacko's MIRACLE GUIDE TO DESIGNING AND PROGRAMMING YOUR OWN ATARI COMPUTER ARCADE GAMES
A BOOK/DISK PACKAGE FOR ANY ATARI COMPUTER

ADDISON-WESLEY
Jacob Way, Dept. MT
Reading, MA 01867



Please send me

__________ copies of DR. WACKO Book/Disk Package 11490 $24.95

__________ copies of DR. WACKO Book ONLY 11488 $12.95

Name ____________________________

Address ____________________________

City __________ State __________ Zip __________

Payment: ____ Check/Money Order enclosed.

____ Bill to VISA # __________ Exp. date __________

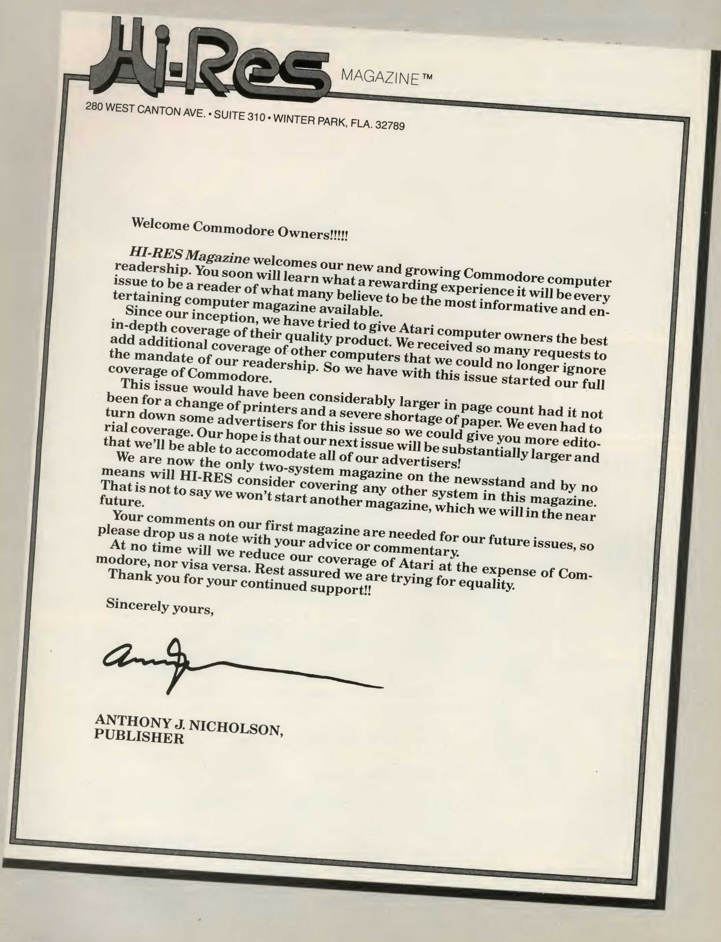
____ Bill to MasterCard # __________ Interbank # __________

Exp. date ____________________________

SIGNATURE: ____________________________

____ Please send me your free catalog.

39006



Hi-Res MAGAZINE™

280 WEST CANTON AVE. • SUITE 310 • WINTER PARK, FLA. 32789

Welcome Commodore Owners!!!!!

HI-RES Magazine welcomes our new and growing Commodore computer readership. You soon will learn what a rewarding experience it will be every issue to be a reader of what many believe to be the most informative and entertaining computer magazine available.

Since our inception, we have tried to give Atari computer owners the best in-depth coverage of their quality product. We received so many requests to add additional coverage of other computers that we could no longer ignore the mandate of our readership. So we have with this issue started our full coverage of Commodore.

This issue would have been considerably larger in page count had it not been for a change of printers and a severe shortage of paper. We even had to turn down some advertisers for this issue so we could give you more editorial coverage. Our hope is that our next issue will be substantially larger and that we'll be able to accomodate all of our advertisers!

We are now the only two-system magazine on the newsstand and by no means will HI-RES consider covering any other system in this magazine. That is not to say we won't start another magazine, which we will in the near future.

Your comments on our first magazine are needed for our future issues, so please drop us a note with your advice or commentary.

At no time will we reduce our coverage of Atari at the expense of Commodore, nor visa versa. Rest assured we are trying for equality.

Thank you for your continued support!!

Sincerely yours,

ANTHONY J. NICHOLSON,
PUBLISHER

Back-Space

Not Happy Now

I was very disappointed in issue number 3. On the basis of issue 2 I subscribed, but now I feel I have been had.

So now *HI-RES* is a Commodore magazine, and next month, let me guess, Peanut or Mac or whatever bandwagon comes along?

— Dwight Brown,
Cleveland Heights, Ohio

I feel betrayed. I won't renew my subscription.

— Keith E. Wilcox,
Clearfield, Utah

As soon as I saw the first issue of *HI-RES* I decided to subscribe. I even wrote a favorable review in our user group's newsletter. I saw some potential.

I would suggest that you drop your coverage of Atari altogether so that true Atari supporters can receive the recognition that's due.

— Greg Peck,
Mexico, Mo.

Too much Atari! More Commodore! Put good Commodore 64 programs and software reviews into magazine. Explain your programs.

— Shawn Dulaney,
Starksville, Miss.

I have no major complaints about your magazine also supporting Commodore, and although it is my understanding that its future is somewhat clouded, it certainly must have dominated Christmas home computer sales by Atari's default.

— Thomas E. Ihle,
Murphysboro, Ill.

A-O.K.

Your mag is really great. Not too general, or too technical. Fantastic!

— John Taylor,
Melbourne, Australia

The two best things of computing are Atari and Hi-Res Magazine.

— Keith Chapman,
Hudson, N.H.

Enjoying your magazine very much. *HI-RES* is better than *Antic* by far. Keep it both for beginners and more advanced. Would like to see more computer, less game machine coverage.

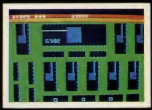
— James D. Beatty,
Massillon, Ohio

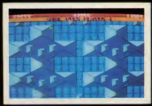
I would like to see articles on business applications for the Atari — eg. a review of accounting programs and 80-column boards used with Atariwriter.

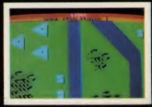
— David Adalian,
Visaha, Calif.

For Apple, Atari & Commodore 64

THE ULTIMATE FLIGHT EXPERIENCE







OVER TWO DOZEN GAME SCREENS!

Put yourself in the pilot's seat of an advanced AGX hypersonic fighter! The year is 2096 and the time is now. Your mission is to destroy COMPUTER CONTROL, a renegade CPU out to take over the universe. You'll fly in low over awe-inspiring scenery whose beauty just might be your end. Guarding the approach to its fortress is an endless array of computer controlled FLAK batteries. They always seem to know where you are, where you're going and where you'll be. You fire, evade, fire again ... Your hand tightens around the flight control, you push harder on the fire button, the tension builds ... Can you take it? Will you survive the FLAK?

BY: Y. Lempereur
and A. Marsily

SEE YOUR DEALER
or write or call for more information. If unavailable from your local dealer, send $34.95 plus $2.00 for shipping. Foreign orders add $5.00 Mastercard/Visa accepted.

FUNSOFT

28611B Canwood St.
Agoura, CA 91301
818-991-6540
TELEX 181161

Available now for:

APPLE II/IIE &
ATARI 800 (XL)/1200 - 48K,
COMMODORE 64
on DISK
$34.95

FLAK is a trademark of FUNSOFT INC.

Apple, Atari and Commodore 64 are trademarks of Apple Computer, Atari, and Commodore Int., Inc. respectively.

—Perspectives—

Duplication Prevention

Louisiana Secretary of State Jim Brown announced new legislation that would make his state the first in the United States to prevent the unauthorized duplication and distribution of computer software.

The bill will be introduced in the current session by Sen. William Atkins and Rep. Al Ater, both democrats.

Industry estimates peg losses to publisher wholesale revenues in 1983 at $360 million for unauthorized duplication of computer software. At least one half-billion dollars are expected to be lost this year.

Nolan Bushnell Returns

Back for a return engagement, Atari has signed an agreement to license and market products designs by Androboy Inc., a company controlled by Atari founder Nolan Bushnell.

Sources said Atari would market a line of products that include *Androman*, a one-foot robot designed to work with Atari's basic video game console, the VCS 2600.

By using 2600 controls, players would be able to use the television screen as a map to steer the robot around obstacles, a source said. Terms of the agreement were not released.

Mattel Backs Out

Mattel Inc., the Hawthorne, Calif., toy maker, is getting out of video games.

Mattel will sell its Intellivision video game business, inventory and related assets for $20 million in cash to a new company formed by Mattel Electronics senior vice president of marketing and sales, Terrence Valeski.

145OXLD Coming Soon?

Atari Inc. says it is reviewing the design and features for its previously announced 1450XLD, a higher priced computer, but still plans to introduce the model this year. Atari originally announced the 1450 XLD last summer.

A Boost at Warner

Warner Communications expects to post a 1983 fourth-quarter profit of about $5 million, but said nine months of losses at its Atari unit resulted in a loss of $420 million for the year.

A $536.4 million loss was recorded by Atari.

However, Steven J. Ross, chairman and chief executive officer, said in an interview with the Wall Street Journal that Warner was able to "turn around" the Atari video game and home computer unit in 1983.

During 1983, Atari cut costs 40 percent through staff cuts and by moving most Atari manufacturing operations overseas.

Activision Suffering

Activision Inc. suffered sharply lower sales in the third quarter of 1983.

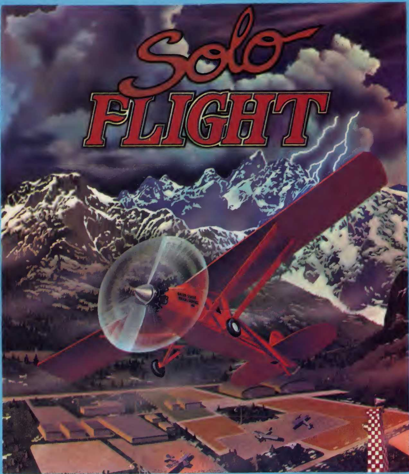
The company attributed the bad news to home computer and video game problems.

In the third fiscal quarter ended Dec. 31, 1983, Activision had net sales of $10 .2 million and a net loss of $8.1 million.

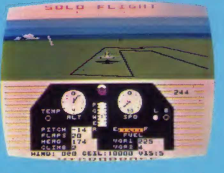
AT LAST!!! A REAL FLIGHT SIMULATOR!!!
— From The Creators of HELLCAT ACE!

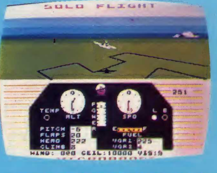
A real flight simulator allows you the pilot to accomplish Takeoffs, Landings, Cross-Country Navigation, and Emergency Procedure practice under Clear (VFR), Cloudy (IFR), and Cross-Wind conditions. SOLO FLIGHT is such an advanced simulation that uses realistic three dimensional terrain graphics, actual configuration instrument panel, multiple air navigation maps, and accurate aircraft performance characteristics to provide the challenge, thrill, excitement, and joy of the flight experience!!!

- Three Dimensional Terrain Graphics
- Multiple Airports and Runways
- Dual Radio Navigation Instruments
- Full Feature, Actual Configuration Instrument Panel
- Multiple Air Navigation Charts
- Instrument Approaches



- Takeoffs
- Landings
- Cross-Country Navigation — Day, VFR and IFR Flying
- Emergency Procedures Practice
- Air Mail Delivery Game for 1-4 Players
- Smooth Landing & Good Navigation Scoring
- For All Atari and Commodore 64 computers, disk or cassette . . . $34.95
- Coming Soon for IBM-PC





Look for MicroProse's two new exciting real time combat simulations — MIG ALLEY ACE & NATO COMMANDER. Write or call for our Free Catalog.

If you cannot find our games at your local store, you can order by MasterCard or VISA, Money Order, COD or Check. Add $2.50 for Postage and Handling. Maryland Residents add 5% Sales Tax.

MicroProse Software

DEALER INQUIRIES INVITED.

10616 Beaver Dam Road, Hunt Valley, MD 21030
(301) 667-1151

Reviews

Airline
Adventure Int.
Longwood, Fla.
Atari 400/800/1200
Commodore 64
Disk or Cassette
$24.95

Airline is a financial strategy game for one to four players by Phillip Case. Upon starting the game you are asked how many turns you want the game to last. This determines the end of the game. The player with the most money at the end of the game is the winner.

The game is played with a joystick and will support either human or computer generated players and, for a change, the computer opponents play a very good game.

The goal of the game is to build a profitable airline and in these days that's not easy. To do this you have to make decisions, such as the types of aircraft to purchase, the most profitable cities to fly between and how to thwart your opponents plans. One interesting aspect of this game is there are not any random factors in determining the players success, therefore, the skillful players win, not the lucky ones.

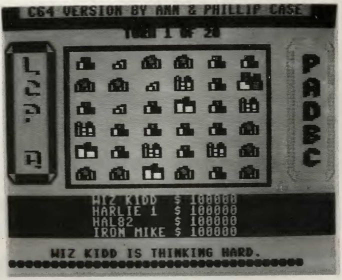
The game is played on a game board of 36 U.S. cities layed out in hi-resolution color graphics, with each player having a different color to represent his cities. (If you are using a black-and-white set, you probably will not be able to tell who is who.)

The game is packaged in a cardboard fold-around box and comes with a well written manual that gives some strategy hints as well as basic playing instructions.

Rating Scale

Game Play	07
Packaging	10
Documentation	10
Graphics	08
Cost Factor	05
Total Rating	**08**

— Stephen Reed



```
C64 VERSION BY ANN & PHILLIP CASE
TURN 1 OF 20
WIZ KIDD   $ 100000
HARLIE 1   $ 100000
HAL82      $ 100000
IRON MIKE  $ 100000
WIZ KIDD IS THINKING HARD.
```

Airline

ARCHON
Electronic Arts
San Mateo, Calif.
Atari 400/800/1200
Commodore 64
32K Disk
$40.00

I'll never forget it. They were aboard the Millenium Falcon, traveling at warp speed through the western arm of the galaxy. To pass the time, the droid, R2D2, had challenged the wookie, Chewbacca, to an animated game of chess. Tiny chess men stomped and crushed each other on the chess board in a miniaturized mimicry of real life.

I'd been playing tournament chess for about five years when I first saw that scene from *Star Wars*, and it left a lasting impression. Wouldn't it be wonderful if you could somehow animate the game of chess? Chess lacks nothing as a mental challenge, of course, but even I saw the appeal of spicing up the game with some holographic histrionics. But, alas, such wizardry seemed more a part of the *Star Wars* future than the Earth-bound present. The future was closer than I thought.

ARCHON, the game I imagined during that brief scene from *Star Wars*, is a chess like strategy game, but with some differences. The game is played by two opposing sides on a board measuring 9x9 squares. It's a battle to the death between the forces of Light and Dark. Even action-addicts could get into this one.

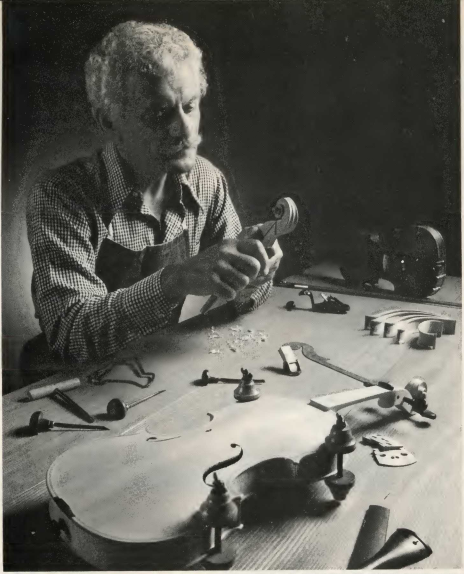
The game begins with opposing sides moving alternately in an attempt to control the arena. Your goal is to occupy five power points distributed around the board or to wipe out the opposing force. Each side has eight different kinds of pieces, each with its own method of travel and with varying degrees of strength on the battlefield.

Battlefield? Yes, this is where *ARCHON* differs from any other game. When opposing forces meet on the same square, they will join battle for possession of that square. And it's a battle to the death.

Once a challenge is fielded, the screen shifts from the chessboard-like strategy screen to the combat arena. Here the two combatants will fight, each using his preferred weapon.

This phase is more like an arcade game than a test of wits. Fast reflexes will usually determine the winner, although a good strategist can compensate for a sluggish trigger finger by choosing his fights carefully.

After the battle, the victor will return to the strategy screen to claim the square he has won. He may be weakened by wounds or fatigue, however, and that will make holding the



Precision Software Tools
The Right Tools For The Right Job

OSS, the leader in software tools for home computers, introduces a new line of application tools that you can use at home yet are versatile enough for the professional.

The Write Tool™

Get all the power you need in a word processor. Compare our features. Compare our value. Buy The Write Tool at the right price. Use with cassettes or disks. On cartridge, $49.95.

Move Over Atari Writer!

OSS
Precision Software Tools
1173D S. Saratoga/Sunnyvale Rd.
San Jose, CA 95129 • (408) 446-3099

The Postal Tool™

Manage your list like never before. Add and change names easily. Prepare bulk mailings. Organize your lists. Select and sort by multiple categories.

The Sort Tool™

Easily sort almost any file. Produce key files, tag files, or fully-sorted files. Sort lines, fixed-length records, or variable-length records. Try The Sort Tool, fast and efficient.

Both The Postal Tool and The Sort Tool on disk only, $59.95.

The Print Tool™

Give your current word processor professional features. Indexing. Outlines. Footnotes. Table of contents. 35mm slides. And much more. Make your documents appear their best. Use The Print Tool. On disk only, $59.95.

See your dealer first. MC/VISA, C.O.D. accepted.

All products available for computers made by Atari, Inc.

The Write Tool, The Print Tool, The Postal Tool, The Sort Tool and OSS are trademarks of Optimized Systems Software, Inc.

square more difficult.

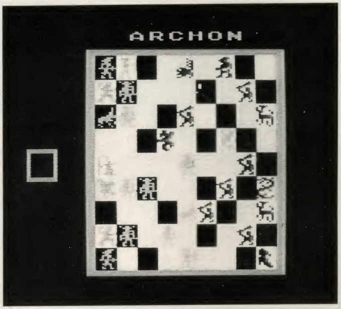
During battle, the Light side is heavily favored on the white squares and the Dark on the black squares. But the luminosity of the squares is not static — a number of them rotate through a cycle of light to dark to light again several times during the course of the game. The fortunes of each side seem to ebb and flow with this changing of luminance. A careful player will harbor his resources when the squares are against him and attack when they are in his favor.

Magic is another factor which can suddenly and radically change the course of the game. Each side has a magician — the Dark has a sorceress and the Light a wizard — each are capable of casting a number of spells. As you might imagine, the judicious use of these spells can have a telling effect on the outcome of the game. But each spell may be used only once, so wisdom is required when invoking these powers.

ARCHON may be played by two players or you may oppose the computer. The computer is a good player, although its chief strength seems to be its agility in the combat arena. After you gain some prowess in the arena, you should start to beat the computer, and in a few months you will require a human opponent for a challenge.

Despite the mechanical simplicity of chess, there is no limit to its strategic complexity. An even simpler game, Go, has at least as much depth. So it always surprises me that people designing new games seem to think complex rules are required for strategic depth. Unfortunatley, *ARCHON* suffers from the same thinking.

ARCHON is a very complex game. It takes some time to learn the rules and there are many things to think about while you play. Also, because there are 16 different kinds of pieces — as opposed to six in chess and one in Go — it requires some concentration to sort them out. Nevertheless, it is possible to sit down and play a game in short order, provided you have the manual by your side.



Archon

Despite its complexity, *ARCHON* is a very good game. It may seem overwhelming at first, but after a while you'll get the hang of it. The authors have struck a fine balance between a strategy game and an arcade shoot-'em-up. It should appeal to fans of both games.

— Léo G. Laporte

C64-Link
Richvale Telecom
Ontario, Canada
Commodore 64
Disk
$120.00

The *C64-Link* is a multipurpose interface for the Commodore 64 computer. Developed by David Foster, this adapter provides your C64 with an IEEE interface for use with all the older Commodore peripherals, such as the 80 32 printer and the high speed 4040 disk drives. Also incorporated into the *C64-Link* are BASIC 4.0 disk I/O commands, a machine language monitor and a terminal software package, all in ROM.

The interface, which measures 3x3x1 inches, fits into the ROM cartridge expansion slot on the back of the Commodore. Using the *C64-Link*, throughout on 40 40 drives, increases about five times normal 1541 speeds. Used in conjunction with the HESMODEM, I've been on Compuserve and several BBS systems without a hitch. The built in machine language monitor, always online, is very useful for saving object files to disk.

The greatest drawback to the *C64-Link* is where it resides in memory: (8000H-9FFFH). This is a poor place for firmware because it creates a compatability problem with many products. To reduce this problem, a *C64-Link* relocater is provided on tape which moves the *C64-Link* software elsewhere in memory. But even with this utility, I've had problems with a number of software packages. I am told by Richvale that a new Compatability Board is being developed which will make the *C64-Link* transparent to most software.

On the back of the *C64-Link* is a connector for IEEE peripherals and with special cables sold by Richvale, you can operate any centronics parallel printer. The 46 page manual provided with the *C64-Link* covers usage of the *C64-Link* adaquately.

At a retail price of more than half the cost of a new C64, this is one of the best peripherals available today for your Commodore. I recommend this device to anyone who is seriously developing software on the Commodore 64.

— Stephen Reed

Easy Spell
Commodore
Westchester, Penn.
Commodore 64
$20.95

When I purchased my Commodore 64 my objective was to build an inexpensive, full-featured word processor. A pipe dream? Well, with a little time and investigation I was able to build a powerful system at a very reasonable price.

As I began to use the system, I was quite pleased with myself for having assembled such an adequate word processor at such an affordable price. However, the system had one major flaw — it couldn't spell!

No matter how much I yelled or how cautious I was (typing at a lightning speed of 15 mistakes-per-minute) the system insisted on misspelling words.

Well, the folks at Commodore must have heard me yelling, because they released *Easy Spell*, a spelling checker for use with Com-

WE SPECIALIZE IN ATARI COMPUTERS SO WE KNOW WHAT WORKS BEST!

COMPUTER PALACE

CALL US AT (503) 683-5361 FOR INFORMATION ON THE LATEST AND BEST FOR ATARI

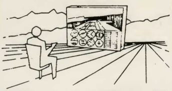
- 2-disk set!
- A lot of game for your money!

$52.50

New from Origin Systems.

Here is the latest in the famous series of Fantasy/Role-Playing games. This one is the best yet!

Flight Simulator II



So what if you haven't gotten your pilot's license yet . . . You can take off with this super-scrolling flight simulator and fly to your heart's content! The disk even contains a WW I flying ace dog fight game so that you can brush up on your aerial combat!

48K Disk From Sublogic **$44.90**

DIMENSION-X

What has unlimited power . . . Scrolling perspective . . . and super excitement? THIS DOES!!!

32K Disk/Tape from Synapse Only **$31.50**

FREE Catalog

With any order, or send $1 (refundable with purchase).

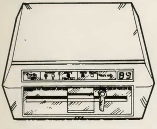
Send us your name & address for FREE FLYERS with Special Offers!

Use Your Credit Cards!

We honor VISA, American Express, & MasterCard.

"IF YOU DON'T HAVE OUR CATALOG... YOU'RE MISSING OUT!"

TRAK DISK DRIVE



- DOUBLE DENSITY
- BUILT-IN PARALLEL PRINTER INTERFACE & BUFFER

$398.00 Plus Frt.

INDUS-GT DISK DRIVE

NEW

New Low Price **$398.00** + Frt.

- Quiet & Reliable
- Faster I/O than Atari 1050 Drive
- True Double Density Operation
- Includes Free Software: DOS—Word Processing—Data-Base—Spread Sheet
- Includes Carrying Case

ONE YEAR WARRANTY!!!

Now You Can Run CP/M On Your Atari!

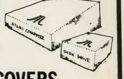
Priced from **$299.00** Plus Freight (16K model, not CP/M compatible)

ATR-8000 Expander

- 64K, Z80 computer
- Runs any combination of 5¼" and 8" drives in single or double density
- Built-in serial and parallel ports
- Built-in printer buffer

64K CP/M model	$449.00
2-DDDS 5¼" add-on drives	595.00
1-DDDS 5¼" add-on drive	379.00

Protect Your Investment with these durable and attractive



DUST COVERS

Custom Fitted For:

- ATARI 400,800,600XL,800XL,1200XL, 410,810,1010,1050,1025 and 1027
- EPSON MX-80 and FX-80
- GEMINI 10 ● PROWRITER 8510
- PERCOM ● RANA

Please specify **Only $8.95** Each

SPECIAL! ***2 for $16.95***

BOOKS & MAGAZINES

COMPUTE PUBLICATIONS:	
1st Book of Atari	12.95
2nd Book of Atari	12.95
1st Book of Atari Graphics	12.95
1st Book of Robots & Computers ...	12.95
1st Book of Atari Games	12.95
Atari Basic Source Book	12.95
Inside Atari DOS	19.95
Mapping the Atari	14.95
Machine Language for Beginners	14.95
Machine Language Programming:	
6502 Assembly Language Prgm.	18.95
6502 Assembly Language Subroutines	17.95
The Atari Assembler	14.95
De Re Atari (~~All~~ About Atari)	17.90
Machine Language for Beginners	14.95
Basic Programming Applications:	
Games Atari's Play	14.95
Kids and the Atari	19.95
Your Atari Computer	17.95
Magazines:	
ANTIC-vol. 2, no. 11	2.50
Back issues-vol2, no.2,3,4,9,10 ea.	2.50
ANALOG-no.16	3.00
Back issues-no.13,14,15 ....ea.	3.00
HiRes-no.3	2.95
Back issues-no. 1, 2 ....ea.	2.95

SUPER SPECIALS!

LIMITED QUANTITIES—WHILE SUPPLIES LAST!

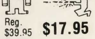
(PLEASE LIST YOUR SECOND-CHOICE WHEN ORDERING)

MOUNTAIN KING

16K cartridge From CBS

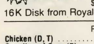
Reg. $39.95 Only **$17.95**

K-Razy Shoot Out



Reg. $39.95 **$17.95**

cartridge from CBS



METEOR STORM

Reg. $29.95 **$9.95**

16K Disk from Royal Software

Boulders & Bombs

From CBS Reg. $39.95

$17.95 16K CARTRIDGE

	Regular	SPECIAL		Regular	SPECIAL
Chicken (D, T)	35.00	**$19.95**	**Protector II (D, T)**	35.00	**$19.95**
Guardian of Gorn (D)	35.00	**9.95**	**David's Midnite Magic (D)** .	35.00	**17.95**
Genetic Drift (D)	30.00	**14.95**	**Race In Space (T)**	30.00	**19.95**
Claim Jumper (D, T)	35.00	**19.95**	**Megalegs (T)**	35.00	**9.95**

SOFTWARE

Joust (C)	44.90
Robotron (C)	43.50
Blue Max (D, T)	31.50
Odesta Chess (D)	59.50
Pole Position (C)	44.90
Donkey Kong (C)	44.90
Monkey Wrench II (C)	54.00
Ultima II (D)	52.50
Dig Dug (C)	44.90
B/Graph (D)	89.90
The Tax Advantage (D)	62.50
Letter Perfect (C)	89.90
Spell Perfect (D)	69.50
Home Accountant (D)	67.50
Q-Bert (C)	44.90
Valforth (D)	54.50

HARDWARE (plus freight)

Astra 1620 Dual Drive	529.00
Rana 1000 Drive	349.00
Super-5 Printer	329.00
Atari 1027 Printer	319.00
Gemini & Epson Printers	CALL
Atari Graphics Tablet	79.10
Koala Pad Tablet	87.90
Power Pad Tablet	87.90
Axiom Printer Interface	99.00

COMPUTER PALACE

OPEN Monday - Saturday, 9 to 6

2160 W. 11th Avenue Eugene, Oregon 97402

USE YOUR CREDIT CARD & CALL

Toll Free 1-800-452-8013

★ ORDERS ONLY, PLEASE ★

There's never a penalty for using your credit card!

For Information, Call (503) 683-5361

Shipping Note: UPS & Parcel Post-$2.90. UPS Air-$4.75
Shipping on hardware & paper supplies-extra. Call.
24 & 48 hour shipping available—Call (503) 683-5361

We specialize in *ATARI Compatible Products*
So we know what works best!!!
Call for the latest info... (503)683-5361

modore's *Easy Script* word processing.

After a file is created on *Easy Script*, *Easy Spell* can be LOADed without interfering with *Easy Script*. When *Easy Spell* is RUN, the user selects a file to check for words that are not present on the 2Ø,ØØØ word dictionary provided.

Once the file is checked (a task which takes from 3Ø seconds to 3 minutes) the program will display statistical data about the file which includes a total number of words, unique words, sentences, paragraphs and average word length. For the writers interested in analyzing their writing, or the educators wishing to check the progress of student writers, this information is invaluable. The program will then display the unrecognized words and offer the option to edit the file.

When the edit option is chosen, the file will be displayed one screen at a time. Unrecognized words will be highlighted in reverse characters. The user has the option to correct the unrecognized word (if it is truly misspelled) or add it to the "user dictionary."

The user dictionary allows the program to be customized to meet the needs of the specialized writer. For example, a technical writer can add technical terms and a writer who uses phonetics can add words like "ya'll" and "youse guys." The program would then accept these words as correctly spelled.

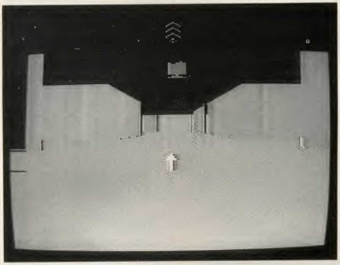
Once the file is edited, the program file will be updated on the disk and an option to return to *Easy Script* is displayed.

Easy Spell is a valuable addition to Commodore's line of word processing software. The manual is well written and easy to understand. Anyone familiar with *Easy Script* can learn to use *Easy Spell* in approximately 3Ø minutes and teach their word processor to spell.

— T. Lee Kidwell

London Blitz
Avalon Hill
Baltimore, Md.
Atari 26ØØ
Cartridge
$3Ø.ØØ

LONDON, October 1940: After massive German bombing raids, the city is a smoldering pile of destruction. While most of the bombs dropped on



London Blitz

London exploded on impact, others failed to detonate. These unexploded bombs (UXBs) now sit like time bombs around the streets of London. Before reconstruction can begin each UXB must be found and defused. That's where you come in.

Assigned to the British Army Royal Engineers, your job is to search the London Streets for UXBs. Once located, each UXB must be disarmed by matching a three-digit combination using three slides. There are three different types of bombs, and each type requires a different disarming strategy. Successfully defuse enough bombs to work your way up to the rank of Captain and you retire a hero. Make too many incorrect attempts to disarm a UXB and you become another casualty.

When the game begins you will be shown a blue and green street map of London. Your present location is shown by a red X, and the locations of the UXBs are shown as blinking white bombs. The map conveniently scrolls in all directions to show you the entire city.

Once you determine which bomb to disarm first, press the joystick button and you will be given the view at street-level. An arrow indicates which direction (in relation to the original map) you are facing. Move the stick forward or backward to travel along the street and to the left or the right to take corners. You can view the map again by simply pushing the button.

When you reach a bomb, you will easily be able to tell which of the three types it is by the color of its slides. Basic Bombs have green slides, Automatic Bombs have blue slides and Random Indicator Bombs have orange slides.

Below the three slides are three indicator lights. After arranging the slides into a possible combination and testing that combination (all done with the joystick), the indicator lights tell you whether each slide is in its proper position (shown by a black light), if it should be moved to the right (green light), or if it should be moved to the left (red).

The difference between the disarming techniques for each of the three bomb types is as follows.

Basic Bombs are defused by entering a combination with the slides and checking the indicator lights. The leftmost indicator corresponds to the top slide, the middle to the middle and the right indicator to the bottom slide.

Automatic Bombs cycle through 16 possible slide combinations automatically; one of which is the correct one. It may be tested at any time to see which, if any slides are in the correct position. If, for example you try a combination that shows the correct position for the top slide to be at the far left, then only try future combinations with the top slide at the far left. In many ways, this is the easiest type of bomb to defuse.

Random Indicator Bombs can be tricky because before you can determine the correct disarming combination, you usually have to figure out which indicator light goes with each slide. This is easier than it sounds. By moving one slide at a time, you will know that a changed indicator corresponds to the last slide moved.

No matter which type of bomb you are attempting to defuse, you will only have a limited amount of time to work on it. This is shown by a green timer below the indicator lights. If you take too long disarming a bomb, the timer will turn yellow and you will only have 33 seconds left. The amount of time to defuse each bomb and the number of incor-

Continued on Page 66

RESTON GUIDES YOU THROUGH THE WORLD OF ATARI.®

Reston takes you step-by-step through the wonders of Atari® Home Computers. **Inside ATARI BASIC ®**, by Bill Carris, is the complete guide for beginners learning Atari BASIC®, or for those who know BASIC but haven't used an Atari® before. It's also a quick reference for color, GTIA and other graphic modes. **Designs From Your Mind with Atari® Graphics** is Tom Rowley's new tutorial book which allows you to realize your imaginative visions with the computer, just as an artist uses canvas and paint. Part One introduces you to shapes, colors, and screen design, and Part Two covers advanced graphics. For the novice and experienced programmer. **Space Knights**, by David Heller and Robert Kurcina, is the book/disk game for sci-fi fans. But it's no simple arcade shoot-'em-up. Instead, it links adventure, science fiction and the computer's color, sound and graphic capabilities for a captivating game of role-playing for the ATARI® 400/800 Home Computer with 24k or 48k memory. Reston can help you discover the worlds of knowledge, excitement and adventure hidden in every Atari® computer.

Space Knights is A Creative Pastime™ from Reston Publishing Company ™

A Prentice-Hall Company
11480 Sunset Hills Rd.
Reston, VA 22090
Available at your local bookstore and computer retailer or call us at
(800) 336-0338.

Atari is a registered trademark of Atari, Inc.

For the ATARI 400/800 Home Computer with 24k or 48k memory and disk drive.

Shakeout i

In the Year of our Lord, Nineteen Hundred and Eighty-Three, the stage was set for a jousting contest of grand scale. As the home computer established itself as a viable product for the mass market, many noble players had their sights set on the championship.

Word spread of the royal competition, and many optimistic jousters mounted their steady steeds preparing to enter the arena of the mass market. Even the Great Blue One (IBM) was rumored to be a possible competitor for the low-end home market. The worthy opponents included the likes of Timex, Sharp Electronics, Radio Shack, Heath Co., Panasonic, Texas Instruments, Sinclair Research Ltd., Apple Computer, Coleco, Atari, Commodore Business Machines and many others.

These competitors all proposed to bring products to the market that would revolutionize the lives of the masses at affordable prices.

Early in the competition it became apparent some of the less-known competitors' steeds were not steady enough, nor their lances sharp enough, for the level of competition about to ensue.

These weaker contestants were defeated early and the real battle of champions began. The survivors included Timex/Sinclair, Coleco, Atari, Commodore and, of course, IBM. The masses viewed the battle with great interest and anticipation — on their television sets.

From January 1983 to September 1983, the competitors spent $101.6 million advertising on television trying to convince potential consumers they were the best bet to win the joust. IBM and Commodore rode in force. In 1983, IBM increased its television advertising purchases by 147 percent over its 1982 expenditures, while Commodore increased its purchases by 83 percent.

In the early championship rounds the jousting became ferocious, lances meeting in fury as the jousters went head-to-head. It was questionable whether the Great Blue One would remain part of such a barbaric battle. Rumors hinted the Great Blue One would wait for the others to wear each other down and then arrogantly enter the arena to capture the final victory.

The defending champion, Atari, had established its place in the low-end home market with the 2600 game system and well-known 400 and 800 computers. Atari was confident and felt well-prepared to deal with whatever the competition might produce. It seemed unconcerned about the rumors and rode proudly into the arena.

Upstart Coleco boisterously announced the coming of its mysterious, but incredibly capable, jouster — the Adam. Adam was touted to possess capabilities far beyond those of any of the competitors', at an un-

n Camelot

t. Lee kidwell

matched price. Adam also was said to be fully compatible with Coleco's game system — which was enjoying great success.

Timex, in cooperation with Sinclair Research Ltd., produced a product of adequate capability at an affordable price — the ZX-80. However, its abilities and configuration matched its price. This isn't to say it was not a good buy at less than $100, at the time it was a very good buy.

Texas Instruments entered the arena with a 16-bit wonder at a price more restrictive than the Timex, but TI rationalized the higher price by proclaiming its power to be unbeatable.

The TI99/4A was totally the creation of Texas Instruments, which refused to reveal any of its inner workings for fear of piracy.

Commodore took action to prove it meant business: putting a 5K computer in the mass market for, as William Shatner told us so many times, "less than $300."

The VIC-20 quickly proved to be the one to beat in 1983 as it began to out-joust all others in the low-end market.

In some respects, it resembled many of the other home computers on the market. However, with a little investigation, it was clear the VIC-20 offered a superior keyboard for word processing and applications requiring extensive text entry. The VIC-20 also sold for $299 retail, as much as $100 less than its nearest competitor.

Commodore had proven itself to be a formidable foe.

Final Rounds

With the jousters' strategies in the open, they began preparation for the final round, shining their armor to a blinding sheen and sharpening their lances to lethal, piercing points.

Commodore, which previously had been involved in a joust with Texas Instruments in the calculator market and lost miserably, paired off with its archrival with revenge in its eyes.

Commodore dropped the price of the VIC-20 to a low that caused its competitor's head to spin. To further gain the upper hand on its archrival, Commodore introduced the very powerful 64K Commodore 64 through authorized dealers, promising the new computer soon would be released to the mass market. The masses anticipated more price cuts on this new machine and waited anxiously for its appearance at local department stores.

To match Commodore's aggressiveness, TI responded by dropping its price on the TI99/4A, but it made critical mistakes in strategy and evaluation of the foe.

As the TI99/4A began to sell, the company still refused to release the secrets of how it worked, making it impossible for third-party vendors to

Continued on Page 64

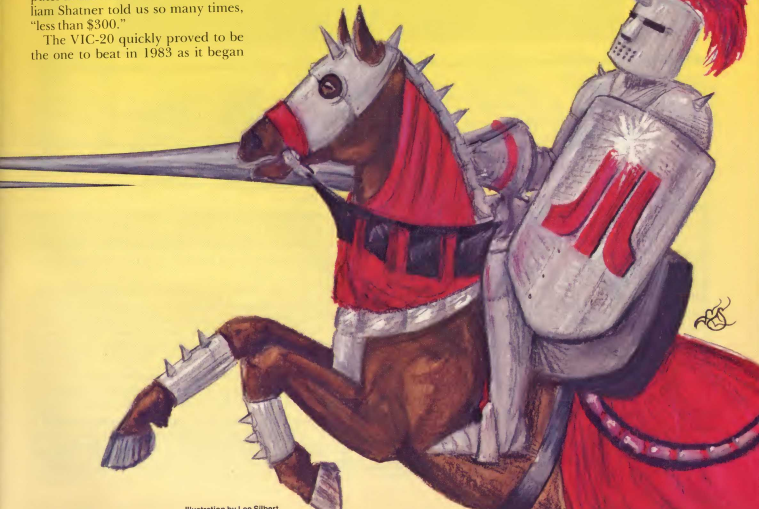
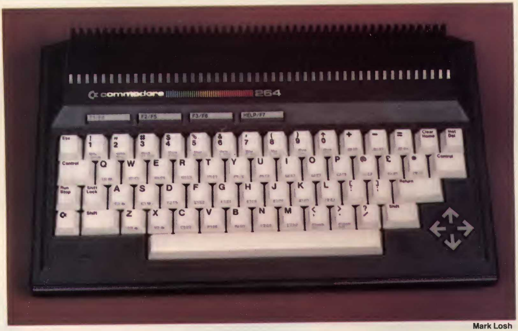


Illustration by Lee Silbert

The 264

Practicality vs. Pac-Man?



Mark Losh

by T. Lee Kidwell

Two days before his resignation as Commodore Business Machines' president, Jack Tramiel proudly posed for photographers at the January 1984 Consumer Electronics Show. In his hands were two of the latest Commodore home computers: the 264 and 364. Tramiel knew how these new products should be marketed, what they should cost, and when they should be put on the market.

But, the man of whom it has been said ruled over Commodore personnel so tightly that "people didn't go to the bathroom without asking Jack for permission," decided to end his reign on Jan. 13, 1984.

Before Tramiel's resignation, Commodore had announced the 264 series of home computers would be available in April. But the company announced the product would be delayed, or as Jim Gracely of Commodore put it, "put on hold."

"When the Commodore 64 was released to the mass market exactly one year ago, we anticipated it would be very well received," said Commodore chairman Irving Gould. "Its reception and continuing strength to date has exceeded our most optimistic projections.

" The rate of demand continues to exceed our production in the post-Christmas quarter. As for our plans related to the 264, this new microcomputer is planned to be introduced in a year and time when our capacity permits both a continuing increase in Commodore 64 production as well as large-scale production of the 264."

Neil Harris, a Commodore communications executive said of the 264: "Its place in the market has been made unclear by many events in the last few months. There has been some internal talk about the proper configuration for the 264.

"When that product line was originally conceived, it was supposed to be a low-end product. The engineers got carried away and designed a 64K machine with all this other stuff in it. I suspect they are going to be rethinking some of that," Harris said.

"I can't really tell for sure what the final outcome will be. The graphics chip and BASIC (a new version called 3.5) will certainly be coming out in some machine this year. But, I wouldn't be surprised myself to see a 16K version of that machine before a 64K version comes out," Harris added.

The 16K version of the 264 is rumored to be called the 116. The, as yet fictional, 116 included all the features of the 264 with the exception of the 64K random access memory and full-stroke keyboard. The 116 keyboard is "chiclet" style similar to the IBM PCjr. Other sources inside Commodore said the 116 will not be released nor will there be any 16K version of the 264 put on the market.

Commodore has received some very negative press and

The Computer Book Club®

Recognized as the authoritative source for micro users!

Superior Performance! Practical Price!

More programs, projects, ways to use your micro for home, hobby, education, and business!

Select 5 Books for Only $2.95

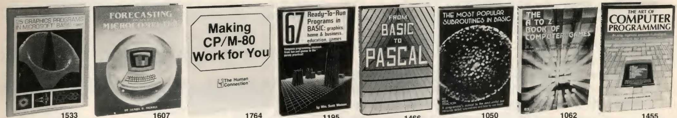
worth up to $105.75

ADVANCED PROGRAMMING TECHNIQUES FOR THE ATARI
ATARI PROGRAMMING WITH 55 PROGRAMS
PROGRAMMING YOUR ATARI COMPUTER
ATARI

1545 List $19.95

1485 List $21.95

1453 List $16.95


25 GRAPHICS PROGRAMS IN MICROSOFT BASIC
FORECASTING ON YOUR MICROCOMPUTER
Making CP/M-80 Work for You
67 Ready-To-Run Programs in BASIC: graphics, home & business, education, games
FROM BASIC TO PASCAL
THE MOST POPULAR SUBROUTINES IN BASIC
THE A TO Z BOOK OF COMPUTER GAMES
THE ART OF COMPUTER PROGRAMMING

1533 List $17.95 · 1607 List $21.95 · 1764 List $16.95 · 1195 List $13.95 · 1466 List $18.95 · 1050 List $13.95 · 1062 List $14.95 · 1455 List $16.95

1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER
ROBOTS AND ROBOTOLOGY
PASCAL
HOW TO CREATE YOUR OWN COMPUTER BULLETIN BOARD
8-BIT & 16-BIT MICROPROCESSOR COOKBOOK
55 ADVANCED COMPUTER PROGRAMS IN BASIC
DATABASE MANAGER IN MICROSOFT BASIC
THE COMPLETE BOOK OF HOME COMPUTERS

Going On-Line with Your Micro
The First Primer of Microcomputer Telecommunications
MAKING MONEY WITH YOUR MICROCOMPUTER
MACHINE & ASSEMBLY LANGUAGE PROGRAMMING
INVESTMENT ANALYSIS WITH YOUR MICROCOMPUTER
THE MICROCOMPUTER BUILDER'S BIBLE

1746 List $17.95 · 1688 List $14.95 · 1506 List $7.95 (paper) · 1389 List $15.95 · 1479 List $19.95 · 1473 List $19.95

Free guide to BASIC Statements & Commands

Plus FREE For Joining ⟶

7 very good reasons to join The Computer Book Club®

- **Big Savings.** Save 20 % to 75 % on books sure to increase your computer know-how
- **No-Risk Guarantee.** All books returnable within 10 days without obligation
- **Club News Bulletins.** All about current selections—mains, alternates, extras—plus bonus offers. Comes 13 times a year with hundreds of up-to-the-minute titles you can pick from
- **"Automatic Order."** Do nothing, and the Main selection will be shipped automatically! But . . . if you want an Alternate selection—or no books at all—we'll follow the instructions you give on the reply form provided with every News Bulletin
- **Bonus Books.** Immediately get a Dividend Certificate with every book purchased and qualify for big discounts of 60 % to 80 %
- **Extra Bonuses.** Take advantage of added-value promotions, plus special discounts on software, games, and more
- **Exceptional Quality.** All books are first-rate publisher's editions selected by our Editorial Board and filled with useful, up-to-the-minute information

The Computer Book Club®

P.O. Box 80, Blue Ridge Summit, PA 17214

Please accept my membership in The Computer Book Club® and send the 5 volumes circled below, *plus* my FREE copy of *BASIC Statements, Commands and Functions,* billing me $2.95 plus shipping and handling charges. If not satisfied, I may return the books within ten days without obligation and have my membership canceled. I agree to purchase 3 or more books at reduced Club prices (plus shipping/handling) during the next 12 months, and may resign any time thereafter.

1050 1062 1160 1195 1205 1295 1389 1423
1453 1455 1466 1473 1479 1485 1506 1533 1545
1567 1607 1633 1643 1673 1688 1746 1764

Name ______________________

Phone ______________________

Address ______________________

City ______________________

State ______________________ Zip ________

Valid for new members only. Foreign applicants will receive special ordering instructions. Canada, must remit in U.S. currency. This order subject to acceptance by The Computer Book Club®.

HIRS-484

comments on the 264 series — possibly the result of the fact the 264 and 364 will run only a minimal amount of Commodore 64 software, a real sore spot with market analysts.

However, the market this machine is aimed at is different from that of the Commodore 64.

The C64 has a sound synthesizer and sprite graphics which make it a superb game computer, in addition to its capabilities as a small business computer. The 264, on the other hand, has less sound capability and does not have sprite graphics. The 264 has two sound generators and BASIC commands (CHAR and LOCATE), which allow STRING of characters to be defined and moved around the screen using X-Y coordinates.

A Commodore spokesman said the 264 is aimed at serious home users, writers, financial analysts and small-business operators.

"The key area we're emphasizing in software for the Commodore 264 is productivity, covering such areas as household management, word processing, calculation, business accounting and education," said Sig Hartmann, president of Commodore Software.

"We believe we are in a decade of increased productivity and microcomputers like the Commodore 264 are going to make an important contribution, helping us to get more things done, more things learned, and more things enjoyed in less time than ever before — without sacrificing quality or efficiency," said Hartmann.

The machine is truly a more business-oriented computer with its optional built-in "integrating" software and "screen window" capability. Imagine working with a word processor and data base or electronic spread sheet simultaneously on the screen. This allows writing on the word processor while viewing data from the data base or spread sheet (i.e., addresses, recipes, dates to remember, inventory control data, financial analysis data, etc.).

With "integrating" software, data can be exchanged from one program to the other. Data from the data base or spread sheet easily can be inserted into a document on the word processor.

"The Commodore 264 is the first personal computer offering a selection of productivity software built into the machine," said Hartmann. "In other words, by choosing a Commodore 264 with a particular software package built in, you can tailor the computer to your own needs.

"If you use your computer to do mostly word processing, you can buy the Commodore 264 with professional word processing built in. If you need financial calculation, you can have a built-in electronic spreadsheet...plus...you can use standard software on cartridge, disk, or tape."

The optional built-in software for the Commodore 264 also will be available on plug-in cartridge. For example, if the machine is purchased with a word processor built in and the owner later decides to purchase the electronic spread sheet, the spread sheet can be purchased on an add-on cartridge.

"One such package is called "3-PLUS-1." "The name 3-PLUS-1 really describes the product," Hartmann said. "There are three essential types of productivity software — word processing, electronic spreadsheet and data-base file management. We have added a PLUS which is business graphics, so the user can visually represent spreadsheet data in chart or graph form."

The fact the 264 can be purchased in different configurations is another sore spot with market analysts. They believe this feature will force retailers to stock various versions of the system, overloading their inventories. It is unknown how Commodore will handle this problem.

The new 3.5 BASIC included in the 264 and 364 is greatly enhanced over the 2.0 BASIC in the Commodore 64. The 3.5 BASIC has several high-resolution graphics commands such as COLOR, DRAW, BOX, CIRCLE AND PAINT. These commands will aid the programmer interested in business applications, which call for the drawing of pie charts and graphs. The 264 and 364 also include the full set of PET keyboard graphics.

The 264 and 364 also have eight programmed, reprogrammable function keys. These keys are preprogrammed to perform such functions as: DLOAD — to easily load a program from disk; DIRECTORY — to load and display the disk directory without erasing a program in memory; SCNCLR — to clear the screen; DSAVE — to easily save a program to disk; RUN, LIST and HELP. The HELP key is very useful when that all too familiar message, "syntax error!" appears.

Some of the more significant features of the Commodore 264 include:

- 64K RAM (60K available for BASIC programming).
- Optional built-in software.
- 32K ROM, 48K in the 364.
- 7501 HMOS microprocessor — .89 to 1.76 MHz clock (A version of the 6510).
- ROM cartridge and parallel disk drive port.
- Two joystick ports (special 264 joysticks only).
- Works with existing C1541 disk drive, C1526 dot-matrix printer, and C1702 color monitor.
- Split screen text high resolution graphics.
- 128 colors (16 colors; eight luminance levels).
- Screen window capability (window can be user defined).
- Four separate cursor control keys.
- Four graphics modes from BASIC including high resolution.
- Warm RESET button.
- PET keyboard graphics.
- Reverse and flashing characters.
- Automatic insert, delete, cursor movement, scrolling and erase functions.
- 25 rows by 40 characters, 320 X 200 resolution.
- Auto line numbering.
- Built-in machine language monitor with 13 commands
- Trace function debugging aid.
- 3.5 BASIC with more than 75 commands including: high-resolution graphics commands (i.e., CIRCLE, BOX); TRAP, for automatic error trapping; CHAR and LOCATE, for moving characters on the screen; DO/UNTIL looping; GETKEY, waits for key to be pressed; PRINT USING, to format string and numeric output; Full complement of disk operating system commands; SSHAPE and GSHAPE, to save and restore multicolor or high resolution characters as BASIC string variables.

The question which remains is how much the 264 and 364 (the 364 includes all the features of the 264 plus a built-in speech synthesizer and numeric keypad) will sell for. As of this writing, the only information Commodore has released is the price will be under $1,000. Sources said, however, the machines will sell from $350 to $500 depending on selected options.

The "PILL" is the most advanced CARTRIDGE BACKUP device available in the WORLD today and is now selling in 21 different countries! Insist on the original . . . insist on THE PILL!

ATARI® ONLY $69.95

ATARI® ONLY $69.95

THE "PILL"!®

WORKS WITH ANY ATARI COMPUTER HAVING 48K or more (400/800/1200XL/600XL/800XL).Saves cartridges to disk OR cassette! Works with DOUBLE DENSITY drives for even greater storage capacity! No installation required.

The "PILL" allows you to store the contents of cartridges designed for any ATARI computer onto disk or cassette (up to twenty 8K programs or ten 16K programs each with file names on a single disk!) simply and instantly!

The "PILL" allows you to select and EXECUTE any of the stored CARTRIDGE programs with equal ease and simplicity!

- Transfers your cartridges to disk or cassette.
- Stores up to 20 programs on a single disk (requires only 7 seconds for 8K programs or 14 seconds for 16K programs).
- Allows you to EXECUTE and run programs which were transferred to disk or cassette.
- All files can be transferred using standard DOS.
- Free software is included with the purchase of THE "PILL" containing several useful utility routines.
- DOUBLE DENSITY menu.
- Works with ALL Atari computers (please specify).
- Available with DISK or CASSETTE (please specify).
- Immediate delivery!

DISTRIBUTOR/DEALER inquiries welcome.

Mastercard-Visa-Money Order or Cashiers Check. Phone orders: **(716) 467-9326.**

Atari is a TM of Atari Inc. The "PILL" is a TM of Computer Software Services (division of S.C.S.D., Inc.)

Send $69.95 plus $4 shipping and handling (N.Y.S. residents please add 7% for sales tax) to:

COMPUTER SOFTWARE SERVICES
P.O. Box 17660
Rochester, New York 14617

Vastly SUPERIOR to any translation programs available! FOR ATARI 1200XL/600XL/800XL with 64K.

ATARI® ONLY $49.95

ATARI® ONLY $49.95

THE XL "FIX"!®

The Atari XL series computers represent power, sophistication, and flexibility virtually unrivalled in todays Home Computer Market.

With "approximately" 30-40% of existing software being "incompatable", a real, and serious problem exists. Because of this we have developed THE XL "FIX"!

ADVANTAGES over cheaper "translation products":

1. The XL "FIX"! is capable of fixing more software . . . an estimated **30% more software!**
2. The XL "FIX"! is available in **DISK** or **CASSETTE** versions.
3. Either XL "FIX"! version fixes ALL THREE types of software (Disk - Cassette - and Cartridges!).
4. The XL "FIX"! adds **OVER 4K** of usable RAM to your computer (anyone using Data bases or Word processors will **really** appreciate this feature!).
5. You never have to hold the OPTION button down on 600XL or 800XL computers!
6. VERY IMPORTANT! You need to load the XL "FIX"! only **once**. . .you can **change** disks, cassettes, or cartridges **without** rebooting the XL "FIX"! each time!

The XL "FIX"! . . . . another SUPERIOR product! **64K required!**

DISTRIBUTOR/DEALER inquiries welcome.

Mastercard-Visa-Money Order or Cashiers Check. Phone orders: **(716) 467-9326.**

Atari is a TM of Atari Inc. The XL "FIX!" is a TM of Computer Software Services (division of S.C.S.D., Inc.)

Send $49.95 plus $4 shipping and handling (N.Y.S. residents please add 7% for sales tax) to:

COMPUTER SOFTWARE SERVICES
P.O. Box 17660
Rochester, New York 14617

DISK to CASSETTE or CASSETTE to DISK Transformations!

ATARI® ONLY $29.95 (or **$49.95**)

ATARI® ONLY $29.95 (or **$49.95**)

THE "METAMORPHOSES"!®

METAMORPHOSIS ONE is really **two** separate and extremely useful utility programs for ALL "PILL" owners. Now you can store all your backup "PILL"! cassette programs onto the much faster and convenient disk format! Store up to 20 "PILL"! cassettes on a single disk!

or

Transfer any "PILL"! programs you have saved on disk to a cassette . . . quickly and simply! Both "PILL"! programs only $29.95.

METAMORPHOSIS TWO is really two extremely FAST machine code programs that allows you to transfer most BINARY disk files (games, utilities, etc.!) to a cassette very quickly!

or

Transfer BINARY file cassettes to a disk (convenient MENU program is provided) and store up to 10 cassette programs per disk! Both BINARY programs only $29.95

SAVE EVEN MORE! Get all 4 "METAMORPHOSES"! programs for only $49.95! Works with all ATARI computers.

DISTRIBUTOR/DEALER inquiries welcome!

Mastercard-Visa-Money Order or Cashiers Check. Phone orders: **(716) 467-9326.**

Atari is a TM of Atari Inc. The "METAMORPHOSES"! is a TM of Computer Software Services (division of S.C.S.D., Inc.)

Send $29.95 or $49.95 plus $4 shipping and handling (N.Y.S. residents please add 7% for sales tax) to:

COMPUTER SOFTWARE SERVICES
P.O. Box 17660
Rochester, New York 14617

For years they said it couldn't be done . . . IMPOSSIBLE they claimed!

ATARI® ONLY $99.95 estimated

ATARI® ONLY $99.95 estimatea

THE "IMPOSSIBLE"!®

HOW WOULD YOU LIKE TO:

1. BACKUP any (or almost any) disk you own with an **UNMODIFIED** drive (works with **ANY** disk drive!)?
2. Compact all program data so you may store **several** programs on the same disk?
3. Give a file name to each game or program and have it become a standard **DOS file**?
4. Work with DOUBLE DENSITY?
5. Interface with CASSETTE to DISK or DISK to CASSETTE programs?

PLUS MUCH MORE!

Here's the story! **THE "IMPOSSIBLE"!** will allow you to make an **"unprotected"** backup copy of virtually any disk currently available and will do it with an UNMODIFIED disk drive. We are in the process of refining and adding further capabilities to our product. At present, we anticipate the selling price to be $99.95 for the main program (Compactor, Cassette to Disk and other auxiliary programs will probably sell for $29.95). Prices are estimated because this ad is being prepared 2 months prior to the actual newstand availability and we still haven't found a method of copy protecting THE "IMPOSSIBLE"! from itself!!! Order early! We expect sales of this product to be brisk!

DISTRIBUTOR/DEALER inquiries welcome.

Mastercard-Visa-Money Order or Cashiers Check. Phone orders: **(716) 467-9326.**

Atari is a TM of Atari Inc. The "IMPOSSIBLE"! is a TM of Computer Software Services (division of S.C.S.D., Inc.)

Send $99.95 (estimated) plus $4 shipping and handling (N.Y.S. residents please add 7% for sales tax) to:

COMPUTER SOFTWARE SERVICES
P.O. Box 17660
Rochester, New York 14617

Music

Illustration by Lee Silbert

Theory Drills

by Duane Tutaj

Most music students can memorize the name and length of each type of rhythmic symbol. However, many have difficulty in combining the various symbols into rhythmic groups.

The third part of this series tests students by having them decide whether the value length of four notes adds up to a complete measure of music.

Previous installments set the groundwork for this segment with coverage of the names of lines and spaces as well as note and rest identification.

In order to make this drill more exciting, several graphics routines are employed to flip the rhythmic symbols in a slot machine-type action.

To succeed in this lesson, the student should not only know how to recognize the various kinds of notes and rests, but also how to determine the length of each note or rest in various time signatures.

Student input is very simple and requires only typing Y or N during the course of the drill. There are four levels of difficulty and the student may choose to have 10, 15 or 20 questions in each drill. If a student needs help in recalling the correct amount of beats for a given note or rest, a press of the H key will generate a series of help screens.

The program begins by displaying the two title screens and playing another variation of the arcade song. The music is followed by a short delay while machine language routines are read into memory. The instructions include a display of all the notes and rests that are used in the drill as well as the help screens that contain the various rhythmic values of the different notes and rests in the time signatures of 4/4, 2/2, and 6/8.

The next screen gives the user the choice of how many questions will be in the drill. This is followed by the main menu.

There are four levels in this drill: level one is 4/4 time; level two is 2/2 or cut time; level three is slow 6/8 time; level four is a combination of the previous three levels. Selections are entered by pushing the SELECT key until a choice appears on the screen. After choosing a level, the game will begin by holding down the START key.

A jump to a graphic screen will take place and the appropriate time signature will be drawn along with four rhythm symbols. The four rhythm symbols will flash in slot machine fashion and generate various clicks. Slowly, each symbol will stop until all four are displayed.

At this time, the drill will ask for the student to identify a complete measure with a Y or N depending on whether a complete measure is displayed. If a key other than Y or N is pressed, a short message will appear, followed by the chance to again enter an answer. If the answer is correct, a pleasant beep is heard. An incorrect answer brings a buzz.

The answer is tabulated and a check is made to see if there are any more questions left in the drill. If there are more questions, another four symbols will appear and begin flipping. If there are no more questions, a new screen showing the score will appear followed by a short message.

Many additions can be made to this program in order to test students on other aspects of rhythmic values. Having the student input the amount of beats that show on the screen is possible. Another choice could be flipping the symbols until the student presses the right key to match a symbol with its value. Correct and incorrect messages could be added, as well as a timed input routine. Also

3/4, 2/4 and fast 6/8 time could be added.

Obviously, a educational-type program is rarely finished. Just as teachers constantly update their lesson plans each week, educational programmers must strive to cover all aspects of a subject. This program is simply the start of that search.

Program Description

The program starts by initializing several string variables and an array for the arcade music. Values also are set for several counters. Line 45 jumps over line 50 to 59. Lines 50 to 59 were placed at the beginning because they are called up so often. If they were in the main loop at line 600, there would be slower execution of the program.

Line 60 jumps to line 2000, which calls the title screens and arcade music at lines 20000 to 26000.

The data in lines 65 to 81 contain the notes for the music and must be placed in the program to give the correct speed of the play loop. Experiment with this concept by retyping lines 65 to 81 elsewhere in the program. Set the RESTORE statement in line 25010 and run the program. At this point, the music slows. Since there is no delay loop built into the arcade music section, it is necessary to place the data in the correct spot to achieve the desired speed.

After the title screens are displayed and the music stops, there is a short delay while the machine language routines are read into various string variables. Lines 200 to 480 contain these routines. Line 2300 clears the screen and sets the background color. Lines 2310 to 2350 ask for the instructions. If Y is pressed, a jump to the instructions at line 10000 is made. If N is pressed, then the amount of questions is asked for. The variable ATTEMPTS is set and all the counters are cleared in line 2350.

Remember, always POKE 764 with 255 before a GET command is executed. If this is not done, the program will place a value in the GET variable if a key had been inadvertantly depressed since the last GET command. Take out the POKE 764,255 and experiment.

Lines 2360 to 2465 contain the Menu screen. In lines 2465 to 2490, the program reads the console keys and changes the level whenever SELECT is depressed. If the START key is depressed, the program goes on to line 2500 — which call the graphics screen and jumps to the main loop in line 608.

Lines 600 to 713 contain the main loop where all the action takes place. The first call starts at line 608. Here, GOSUB to line 3000 and get the correct time signature drawn to the screen from lines 3000 to 3059.

On the return, the amount of symbol flipping is selected by a random number. Each of the four symbols has a different number count and this is decreased in line 610. A check is made in line 611 to see if the first symbol has stopped flipping. If not, the program checks to see if this is 6/8 time and if so, it limits the kinds of symbols that can be selected. If it is not 6/8 time, any of 14 choices can be made. A short GOSUB to line 1000 gets the correct symbol and draws it to the screen with a USR command. A click is made and then a check of the next symbol.

This loop repeats until all four symbols have been selected and printed to the screen. Lines 1500 to 1980 contain the section where the last symbol is chosen and printed. This routine had to be added since the odds favored that there was usually not a full measure. By careful selection of the last symbol, the odds of a complete or incomplete measure is now about 50-50.

Next, the symbols are given beat values and added. The program jumps to line 600 and asks if this is a full measure. There is a check for only the H, Y or N keys. Upon input, the program jumps to the correct section and checks the answer or prints the HELP screens.

Line 606 checks for end of drill at lines 4000 to 4520. Line 607 goes to line 950 and clears the symbols and returns.

Line 900 is the end sound of the flipping symbols. It also turns off all the other voices.

Lines 4000 to 4520 check for end of drill and display the final screen with an appropriate message. Lines 10000 to 10180 contain the instructions while lines 10200 to 10320 display each symbol and its name.

Lines 10400 to 10680 are the HELP screens and print the correct beat values of each type of note in the various time signatures.

Lines 20000 to 25320 are the title screens and arcade music loop discussed in the previous installments.

Editor's note: If you have enjoyed the past three articles and wish to see other areas of music theory or sound covered, please write to Duane Tutaj in care of Hi-Res Magazine, 280 W. Canton Ave., Suite 310, Winter Park, Fla., 32789.

Variable Table

Temporary variables	Z,I,A,WW4,V,T1-T4,W1-W4
Input variables	K,KEY,FM,ATTEMPTS,MENU,TS
Screen variables	X,Y
Score variables	C,ATT,W,SCORE,TOTAL,BEAT,H
Sound variables	YY(4)
Message variables	F$,F1$,BL$
Image variables	I1$-I7$,I1-I7,E$,E

Music Theory Drills

```
10 REM ******************************
20 REM * MUSIC THEORY DRILLS PART 3 *
30 REM * DUANE TUTAJ                *
35 REM * COPYRIGHT 12/83            *
40 REM *******************************
41 DIM F$(9),F1$(9),BL$(29),YY(4):BL$=
"                             "
42 TOTAL=0:BEAT=0:MENU=1:TS=0
45 GOTO 60
50 REM *** BEAT CONVERSION ***
```

```
51 IF I=1 THEN BEAT=2
52 IF I=2 THEN BEAT=1.5
53 IF I=3 OR I=8 OR I=13 THEN BEAT=1
54 IF I=4 OR I=9 OR I=14 THEN BEAT=0.5
55 IF I=5 OR I=10 OR I=12 THEN BEAT=1
56 IF I=6 OR I=11 OR I=7 THEN BEAT=0.5
57 TOTAL=TOTAL+BEAT
59 RETURN
60 GOTO 2000
65 DATA 53,0,0,162,0,108,128,162,64,0,
0,217,60,108,128,217
66 DATA 53,0,0,162,40,108,128,162,0,0,
0,217,53,108,128,217
67 DATA 42,0,0,162,0,108,128,162,47,0,
0,217,0,108,128,217
68 DATA 53,108,128,162,53,0,0,217,0,0,
0,193,0,0,0,173
69 DATA 53,0,0,162,0,108,128,162,64,0,
0,217,60,108,128,217
70 DATA 53,0,0,162,40,108,128,162,0,0,
0,217,53,108,128,217
71 DATA 60,0,0,173,0,121,144,173,64,0,
0,217,0,121,144,217
72 DATA 72,121,144,173,72,0,0,217,0,0,
0,193,0,0,0,173
73 DATA 53,0,0,162,0,108,128,162,64,0,
0,217,60,108,128,217
74 DATA 53,0,0,162,40,108,128,162,0,0,
0,217,53,108,128,217
75 DATA 42,0,0,162,0,108,128,162,47,0,
0,217,0,108,128,217
76 DATA 53,108,128,162,53,0,0,217,0,0,
0,193,0,0,0,173
77 DATA 40,0,0,162,0,108,128,162,53,0,
0,217,0,108,128,217
78 DATA 64,0,0,162,0,108,128,162,81,0,
0,217,0,108,128,217
79 DATA 64,0,0,162,81,108,128,162,72,0
,0,217,85,121,144,217
80 DATA 81,108,128,162,81,108,128,162,
81,108,128,162,0,0,0,0
81 DATA 9999
0198 RESTORE 210
0199 REM *** IMAGES ***
0200 DIM I1$(60):FOR I=1 TO 60:READ A:
I1$(I,I)=CHR$(A):NEXT I:I1=ADR(I1$)
0210 DATA 0,0,1,64,0,0,0,1,64,0,0,0,1,
64,0,0,0,1,64,0,0,0,1,64,0,0,0,1,64,0
0220 DATA 0,5,85,64,0,0,20,5,64,0,0,80
,1,64,0,0,80,1,64,0,0,20,5,0,0,0,5,84,
0,0
0230 DIM I2$(60):FOR I=1 TO 60:READ A:
I2$(I,I)=CHR$(A):NEXT I:I2=ADR(I2$)
0240 DATA 0,0,10,0,0,0,0,10,0,0,0,0,10
,0,0,0,0,10,0,0,0,0,10,0,0,0,0,10,0,0
0250 DATA 0,42,170,0,0,0,170,170,0,0,2
,170,170,10,0,2,170,170,10,0,0,170,168
,0,0,0,42,160,0,0
0260 DIM I3$(60):FOR I=1 TO 60:READ A:
I3$(I,I)=CHR$(A):NEXT I:I3=ADR(I3$)
0270 DATA 0,0,60,0,0,0,0,60,0,0,0,0,60
,0,0,0,0,60,0,0,0,0,60,0,0,0,0,60,0,0
0280 DATA 0,63,252,0,0,0,255,252,0,0,3
,255,252,0,0,3,255,252,0,0,0,255,240,0
,0,0,63,192,0,0
0290 DIM I4$(60):FOR I=1 TO 60:READ A:
I4$(I,I)=CHR$(A):NEXT I:I4=ADR(I4$)
```

```
0300 DATA 0,0,21,0,0,0,0,21,64,0,0,0,2
0,80,0,0,0,20,80,0,0,0,20,80,0,0,0,20,
0,0
0310 DATA 0,21,84,0,0,0,85,84,0,0,1,85
,84,0,0,1,85,84,0,0,0,85,80,0,0,0,21,6
4,0,0
0320 DIM I5$(60):FOR I=1 TO 60:READ A:
I5$(I,I)=CHR$(A):NEXT I:I5=ADR(I5$)
0330 DATA 0,32,0,0,0,0,40,0,0,0,0,10,1
68,0,0,0,0,40,0,0,0,2,160,0,0,0,10,128
,0,0
0340 DATA 0,40,0,0,0,0,42,168,0,0,0,0,
40,0,0,0,0,160,0,0,0,2,128,0,0,0,10,0,
0,0
0350 DIM I6$(60):FOR I=1 TO 60:READ A:
I6$(I,I)=CHR$(A):NEXT I:I6=ADR(I6$)
0360 DATA 0,0,0,0,0,0,0,0,0,0,240,3,
0,0,0,60,15,0,0,0,15,255,0,0,0,0,15,0,
0
0370 DATA 0,0,60,0,0,0,0,240,0,0,0,3,1
92,0,0,0,15,0,0,0,0,60,0,0,0,0,240,0,0
,0
0380 DIM I7$(60):FOR I=1 TO 60:I7$(I,I
)=CHR$(0):NEXT I:I7=ADR(I7$)
0400 REM *** ROUT FOR IMAGES IN GR. 7
0410 DIM E$(123):FOR I=1 TO 123:READ A
:E$(I,I)=CHR$(A):NEXT I:E=ADR(E$)
0420 DATA 104,104,104,133,195,104,104,
133,186,104,133,213,104,133,212,169,0,
133,187,165,186
0430 DATA 10,38,187,10,38,187,10,38,18
7,133,186,164,187,10,38,187,10,38,187,
24,101,186,133,186,152
0440 DATA 101,187,133,187,24,165,186,1
01,195,144,2,230,187,24,101,88,133,186
,165,187,101,89,133,187
0450 DATA 169,12,133,195,162,5,160,0,1
77,212,145,186,24,165,212,105,1,133,21
2,144,2
0460 DATA 230,213,24,165,186,105,1,133
,186,144,2,230,187,202,208,227,24,165,
186,105,35,133,186,144,2
0470 DATA 230,187,198,195,208,208,96
0480 RETURN
0590 GRAPHICS 7:POKE 752,1:GOTO 608
0596 GOTO 608
0599 REM *** MAIN LOOP ***
0600 POKE 656,0:PRINT "        IS THIS
A FULL MEASURE?":FM=0:POKE 764,255:GET
 #1,FM
0601 IF FM=72 THEN GOTO 11000
0602 IF FM=89 OR FM=78 THEN GOTO 606
0603 POKE 656,0:PRINT "        PLEASE T
YPE Y OR N      ":FOR I=1 TO 400:NEXT I
:GOTO 600
0606 GOSUB 4000:TOTAL=0
0607 GOSUB 950:POKE 712,0
0608 GOSUB 3000:T1=INT(RND(0)*10+5):T2
=T1+INT(RND(0)*5):T3=T2+INT(RND(0)*7):
T4=T3+INT(RND(0)*9)
0610 T1=T1-1:T2=T2-1:T3=T3-1:T4=T4-1
0611 IF T1<0 THEN GOTO 616
0612 IF MENU=3 OR TS=3 THEN X=7:Y=15:I
=INT(RND(0)*3+1):W1=I:GOSUB 1000
0613 X=7:Y=15:I=INT(RND(0)*14+1):W1=I:
GOSUB 1000
0614 SOUND 0,20,10,10:SOUND 0,0,0,0
```

```
0615 IF T1=0 THEN GOSUB 900
0616 IF T2<0 THEN GOTO 650
0620 IF MENU=3 OR TS=3 THEN X=13:Y=15:
I=INT(RND(0)*3+2):W2=I:GOSUB 1000
0621 X=13:Y=15:I=INT(RND(0)*14+1):W2=I
:GOSUB 1000
0630 SOUND 1,40,10,10:SOUND 1,0,0,0
0640 IF T2=0 THEN GOSUB 900
0650 IF T3<0 THEN GOTO 675
0655 X=19:Y=15:I=INT(RND(0)*14+1):W3=I
:GOSUB 1000
0660 SOUND 2,60,10,12:SOUND 2,0,0,0
0670 IF T3=0 THEN GOSUB 900:GOSUB 710:
GOTO 1500
0675 IF T4<0 THEN GOTO 700
0680 X=25:Y=15:I=INT(RND(0)*14+1):W4=I
:GOSUB 1000
0685 SOUND 3,80,10,12:SOUND 3,0,0,0
0690 IF T4=0 THEN GOSUB 900:GOTO 705
0700 IF T1>0 OR T2>0 OR T3>0 THEN 610
0705 I=W4:GOSUB 50:GOTO 600
0710 I=W1:GOSUB 50
0711 I=W2:GOSUB 50
0712 I=W3:GOSUB 50
0713 RETURN
0899 REM *** CLANK- WHEEL STOP ***
0900 SOUND 0,150,14,10:FOR I=1 TO 10:N
EXT I:FOR I=0 TO 3:SOUND I,0,0,0:NEXT
I:RETURN
0950 PRINT CHR$(125):A=USR(E,7,15,I7):
A=USR(E,13,15,I7):A=USR(E,19,15,I7):A=
USR(E,25,15,I7):A=USR(E,2,12,I7)
0951 A=USR(E,2,19,I7):RETURN
0999 REM *** PRT IMAGES ***
1000 IF I=1 THEN A=USR(E,X,Y,I1)
1010 IF I=2 THEN A=USR(E,X,Y,I2)
1020 IF I=3 THEN A=USR(E,X,Y,I3)
1030 IF I=4 THEN A=USR(E,X,Y,I4)
1040 IF I=5 THEN A=USR(E,X,Y,I5)
1050 IF I=6 THEN A=USR(E,X,Y,I6)
1060 IF I=7 THEN A=USR(E,X,Y,I6)
1070 IF I=8 THEN A=USR(E,X,Y,I3)
1080 IF I=9 THEN A=USR(E,X,Y,I4)
1090 IF I=10 THEN A=USR(E,X,Y,I5)
1100 IF I=11 THEN A=USR(E,X,Y,I6)
1110 IF I=12 THEN A=USR(E,X,Y,I5)
1130 IF I=13 THEN A=USR(E,X,Y,I3)
1140 IF I=14 THEN A=USR(E,X,Y,I4)
1150 RETURN
1500 REM *** FULL MEASURE SELECTION
1502 V=INT(RND(0)*3)+1
1508 IF MENU=1 OR TS=1 AND TOTAL<4 THE
N GOTO 1700
1510 IF MENU=2 OR TS=2 AND TOTAL<4 THE
N GOTO 1700
1520 IF MENU=3 OR TS=3 AND TOTAL<6 THE
N GOTO 1900
1530 T4=0:GOTO 680
1700 WW4=4-TOTAL
1710 IF WW4=2 AND V<3 THEN WW4=1:GOTO
1980
1720 IF WW4=1 AND V<3 THEN WW4=3:GOTO
1980
1730 IF WW4=1.5 THEN WW4=2:GOTO 1980
1740 T4=0:GOTO 680
1900 WW4=TOTAL
1910 IF WW4=2 THEN WW4=3:GOTO 1980
1920 IF WW4=1 THEN WW4=1:GOTO 1980
1930 IF WW4=1.5 THEN WW4=2:GOTO 1980
1935 IF WW4=2.5 THEN WW4=4:GOTO 1980
1940 T4=0:GOTO 680
1980 X=25:Y=15:I=WW4:GOSUB 1000:GOSUB
900:W4=WW4:I=W4:GOSUB 50:GOTO 600
2000 OPEN #1,4,0,"K:":GOSUB 20000
2250 GRAPHICS 0:POKE 710,160:POKE 752,
1
2260 POSITION 10,10:PRINT "ONE MOMENT
PLEASE":GOSUB 198
2300 PRINT CHR$(125):SETCOLOR 2,4,0
2310 POSITION 7,3:PRINT "DO YOU WISH I
NSTRUCTIONS?  ":K=0:GET #1,K
2320 IF K=89 THEN GOSUB 10000:GRAPHICS
 0:GOSUB 2341:GOTO 2355
2330 IF K=78 THEN GOSUB 2341:GOTO 2355
2340 POSITION 7,3:PRINT "  PLEASE TYPE
 YES OR NO   ":FOR I=1 TO 50:SOUND 0,4
0,2,10:NEXT I:SOUND 0,0,0,0:GOTO 2310
2341 SETCOLOR 2,4,0:POKE 752,1:POSITIO
N 5,5:PRINT "HOW MANY PROBLEMS DO YOU
WANT"
2342 POSITION 10,7:PRINT " (TYPE 1, 2,
 OR 3)     "
2343 POSITION 12,9:PRINT "1) 10 PROBLE
MS"
2344 POSITION 12,11:PRINT "2) 15 PROBL
EMS"
2345 POSITION 12,13:PRINT "3) 20 PROBL
EMS"
2346 GET #1,K:IF K<=48 OR K>=52 THEN P
OSITION 10,7:PRINT "TYPE 1 ,2, OR 3 PL
EASE":FOR I=1 TO 150:NEXT I:GOTO 2341
2347 IF K=49 THEN ATTEMPTS=10
2348 IF K=50 THEN ATTEMPTS=15
2349 IF K=51 THEN ATTEMPTS=20
2350 TOTAL=0:BEAT=0:C=0:W=0:H=0:SCORE=
0:ATT=0:RETURN
2355 PRINT CHR$(125):POKE 710,32
2360 POSITION 8,1:PRINT "*************
***********"
2370 POSITION 10,3:PRINT "MUSICAL SLOT
S MENU"
2380 POSITION 10,6:PRINT "1)  4/4 TIME
 ONLY"
2410 POSITION 10,9:PRINT "2)  2/2 TIME
 ONLY"
2420 POSITION 10,12:PRINT "3)  6/8 TIM
E ONLY"
2430 POSITION 10,15:PRINT "4)  ALL TIM
E SIGNATURES"
2440 POSITION 8,18:PRINT "************
************"
2450 POSITION 4,20:PRINT "USE SELECT K
EY TO PICK SELECTION"
2460 POSITION 8,21:PRINT "PUSH START T
O BEGIN GAME"
2465 MENU=1
2470 KEY=PEEK(53279)
2472 IF KEY=6 THEN GOTO 2500
2473 IF KEY=5 THEN MENU=MENU+1
2474 IF MENU>=5 THEN MENU=1
2480 POSITION 17,22:PRINT "LEVEL ";MEN
U
2490 FOR I=1 TO 20:NEXT I:GOTO 2470
2500 GRAPHICS 7:POKE 752,1:GOTO 608
3000 IF MENU=1 THEN GOTO 3014
3004 IF MENU=2 THEN GOTO 3040
```

```
3005 IF MENU=3 THEN GOTO 3050
3006 IF MENU=4 THEN TS=INT(RND(0)*3)+1
3007 IF TS=1 THEN GOTO 3014
3010 IF TS=2 THEN GOTO 3040
3011 IF TS=3 THEN GOTO 3050
3014 COLOR 1:PLOT 9,14:DRAWTO 9,18:DRA
WTO 13,18:PLOT 13,14:DRAWTO 13,21
3015 PLOT 9,23:DRAWTO 9,27:DRAWTO 13,2
7:PLOT 13,23:DRAWTO 13,30
3019 RETURN
3040 COLOR 1:PLOT 9,14:DRAWTO 13,14:DR
AWTO 13,16:DRAWTO 9,20:PLOT 9,15:PLOT
9,21:DRAWTO 13,21
3045 PLOT 9,23:DRAWTO 13,23:DRAWTO 13,
25:DRAWTO 9,29:PLOT 9,24:PLOT 9,30:DRA
WTO 13,30
3049 RETURN
3050 COLOR 1:PLOT 10,14:DRAWTO 12,14:P
LOT 13,15:PLOT 9,15:DRAWTO 9,21:DRAWTO
 13,21:DRAWTO 13,18:DRAWTO 9,18
3055 PLOT 9,23:DRAWTO 9,30:DRAWTO 13,3
0:DRAWTO 13,23:DRAWTO 9,23:PLOT 9,26:D
RAWTO 13,26:PLOT 9,27:DRAWTO 13,27
3059 RETURN
4000 IF MENU=1 OR TS=1 THEN GOTO 4100
4003 IF MENU=2 OR TS=2 THEN GOTO 4140
4004 IF MENU=3 OR TS=3 THEN GOTO 4150
4007 RETURN
4100 IF FM=89 AND TOTAL=4 THEN GOTO 42
00
4102 IF FM=78 AND TOTAL<>4 THEN GOTO 4
200
4104 GOTO 4300
4140 IF FM=89 AND TOTAL=4 THEN GOTO 42
00
4142 IF FM=78 AND TOTAL<>4 THEN GOTO 4
200
4144 GOTO 4300
4150 IF FM=89 AND TOTAL=3 THEN GOTO 42
00
4152 IF FM=78 AND TOTAL<>3 THEN GOTO 4
200
4154 GOTO 4300
4200 SOUND 0,150,10,10:FOR I=1 TO 50:N
EXT I:SOUND 0,0,0,0
4201 C=C+1:ATT=ATT+1
4202 IF ATT=ATTEMPTS THEN GOTO 4400
4203 RETURN
4300 SOUND 0,150,2,10:FOR I=1 TO 50:NE
XT I:SOUND 0,0,0,0
4301 W=W+1:ATT=ATT+1
4302 IF ATT=ATTEMPTS THEN GOTO 4400
4303 RETURN
4400 GRAPHICS 0:POKE 710,227:POKE 752,
1
4410 POSITION 8,1:PRINT "YOUR FINAL RE
SULTS WERE:"
4420 POSITION 14,3:PRINT "CORRECT=";C
4430 POSITION 14,5:PRINT "WRONG  =";W
4440 POSITION 14,7:PRINT "HELPS  =";H
4441 C=C-H:SCORE=INT((C*100)/ATTEMPTS)
4450 POSITION 14,9:PRINT "SCORE      ="
;SCORE;"%"
4460 FOR I=1 TO 150:NEXT I
4461 IF SCORE=100 THEN GOTO 4470
4462 IF SCORE>=90 THEN GOTO 4480
4463 IF SCORE>=70 THEN GOTO 4485
4464 IF SCORE<=69 THEN GOTO 4490
```

```
4470 F$="FANTASTIC":F1$=[INVERSE CHARA
CTERS]"FANTASTIC"
4471 FOR I=1 TO 15:POSITION 16,12:PRIN
T F$
4472 SOUND 0,55,10,10:FOR Z=1 TO 3
4473 POSITION 16,12:PRINT F1$:NEXT Z:S
OUND 0,0,0,0:NEXT I:GOTO 4500
4480 POSITION 10,12:PRINT "VERY GOOD":
GOTO 4500
4485 POSITION 3,12:PRINT "YOU SEEM TO
UNDERSTAND THIS LEVEL"
4486 POSITION 3,13:PRINT "BUT TRY IT A
GAIN SO THAT YOU CAN":PRINT " MASTER I
T!":GOTO 4500
4490 POSITION 3,12:PRINT "YOU NEED MOR
E WORK ON THIS LEVEL"
4491 POSITION 3,13:PRINT "GO BACK AND
STUDY THE INSTRUCTIONS":GOTO 4500
4500 POSITION 8,20:PRINT "DO YOU WISH
TO PLAY AGAIN"
4501 K=0
4502 POKE 764,255:GET #1,K
4504 IF K=89 THEN GOTO 4510
4506 IF K=78 THEN GOTO 4520
4508 POSITION 10,20:PRINT "PLEASE TYPE
 Y OR N               ":FOR I=1 TO 100:N
EXT I:GOTO 4500
4510 TOTAL=0:BEAT=0:MENU=1:TS=0:PRINT
CHR$(125):SETCOLOR 2,4,0:GOSUB 2341:GO
TO 2355
4520 GRAPHICS 0:CLOSE #1:END
010000 PRINT CHR$(125):POKE 710,64:POK
E 752,1
010010 POSITION 2,3:PRINT [INVERSE]"MU
SICAL SLOTS teaches note values in  se
veral time signatures by randomly"
010020 PRINT "printing various notes o
r rests to    the screen."
010030 POSITION 2,8:PRINT "The object
is to add up the beats and see if the
total number of beats add"
010040 PRINT "up to a complete measure
 of music."
010050 POSITION 2,13:PRINT "If the tot
al is correct-type Y or     if incorre
ct type N."
010060 PRINT "Your score is based on t
he amount of  correct answers versus y
our attempts."
010070 POSITION 2,18:PRINT "When the a
ctivity is finished your    score will
 be shown with an option to play again
."
010075 POSITION 2,22
010080 PRINT [INVERSE]"<<<     HIT ANY
 KEY TO CONTINUE   >>>":K=0:POKE 764,25
5:GET #1,K:IF K<>255 THEN 10100
010081 GOTO 10080
010100 PRINT CHR$(125)
010110 POSITION 2,3:PRINT "If you want
 to see a chart of the     correct not
e values for the different"
010120 PRINT "time signatures, type H
for HELP."
010130 POSITION 2,8:PRINT "You may typ
e H at anytime but your    score will
be lowered."
```

```
010170 POSITION 2,22
010180 PRINT [INVERSE]"<<<      HIT ANY
 KEY TO CONTINUE  >>>":K=0:POKE 764,25
5:GET #1,K:IF K<>255 THEN 10200
010181 GOTO 10180
010200 GRAPHICS 7:POKE 752,1
010210 A=USR(E,7,15,I1)
010213 POKE 656,0:PRINT BL$
010214 POKE 656,0:SOUND 0,121,10,10
010215 PRINT "THIS IS A HALF NOTE"
010220 FOR Z=1 TO 200:NEXT Z
010230 A=USR(E,7,15,I2)
010233 POKE 656,0:PRINT BL$
010234 POKE 656,0:SOUND 0,96,10,10
010235 PRINT "THIS IS A DOTTED QUARTER
 NOTE"
010240 FOR Z=1 TO 200:NEXT Z
010250 A=USR(E,7,15,I3)
010253 POKE 656,0:PRINT BL$
010254 POKE 656,0:SOUND 0,81,10,10
010255 PRINT "THIS IS A QUARTER NOTE
  "
010260 FOR Z=1 TO 200:NEXT Z
010270 A=USR(E,7,15,I4)
010273 POKE 656,0:PRINT BL$
010274 POKE 656,0:SOUND 0,91,10,10
010275 PRINT "THIS IS AN EIGHTH NOTE
  "
010280 FOR Z=1 TO 200:NEXT Z
010290 A=USR(E,7,15,I5)
010293 POKE 656,0:PRINT BL$
010294 POKE 656,0:SOUND 0,64,10,10
010295 PRINT "THIS IS A QUARTER REST
  "
010300 FOR Z=1 TO 200:NEXT Z
010310 A=USR(E,7,15,I6)
010313 POKE 656,0:PRINT BL$
010314 POKE 656,0:SOUND 0,60,10,10
010315 PRINT "THIS IS AN EIGHTH REST
  "
010320 FOR Z=1 TO 200:NEXT Z:SOUND 0,0
,0,0
010400 REM *** CHART FOR HELP ***
010410 GRAPHICS 0:POKE 752,1:SETCOLOR
2,4,0
010420 POSITION 2,2:PRINT "4/4 TIME"
010425 POSITION 4,4:PRINT "HALF NOTE
     = 2 BEATS"
010430 POSITION 4,5:PRINT "QUARTER NOT
E   = 1 BEAT"
010435 POSITION 4,6:PRINT "EIGHTH NOTE
     = 1/2 BEATS"
010440 POSITION 4,7:PRINT "DOTTED QUAR
TER = 1 & 1/2 BEATS"
010445 POSITION 4,8:PRINT "QUARTER RES
T   = 1 BEAT"
010450 POSITION 4,9:PRINT "EIGHTH  RES
T   = 1/2 BEAT"
010455 POSITION 2,22
010460 PRINT [INVERSE]"<<<      HIT ANY
 KEY TO CONTINUE  >>>":K=0:POKE 764,25
5:GET #1,K:IF K<>255 THEN 10500
010461 GOTO 10460
010500 PRINT CHR$(125)
010520 POSITION 2,2:PRINT "2/2  OR CUT
 TIME"
010525 POSITION 4,4:PRINT "HALF NOTE
     = 1 BEAT"
010530 POSITION 4,5:PRINT "QUARTER NOT
E   = 1/2 BEAT"
010535 POSITION 4,6:PRINT "EIGHTH NOTE
     = 1/4 BEAT"
010540 POSITION 4,7:PRINT "DOTTED QUAR
TER = 3/4 BEAT"
010545 POSITION 4,8:PRINT "QUARTER RES
T   = 1/2 BEAT"
010550 POSITION 4,9:PRINT "EIGHTH  RES
T   = 1/4 BEAT"
010555 POSITION 2,22
010560 PRINT [INVERSE]"<<<      HIT ANY
 KEY TO CONTINUE  >>>":K=0:POKE 764,25
5:GET #1,K:IF K<>255 THEN 10600
010561 GOTO 10560
010600 PRINT CHR$(125)
010620 POSITION 2,2:PRINT "SLOW 6/8 TI
ME"
010625 POSITION 4,4:PRINT "HALF NOTE
     = 4 BEATS"
010630 POSITION 4,5:PRINT "QUARTER NOT
E   = 2 BEATS"
010635 POSITION 4,6:PRINT "EIGHTH NOTE
     = 1 BEAT"
010640 POSITION 4,7:PRINT "DOTTED QUAR
TER = 3 BEATS"
010645 POSITION 4,8:PRINT "QUARTER RES
T   = 2 BEATS"
010650 POSITION 4,9:PRINT "EIGHTH  RES
T   = 1 BEAT"
010655 POSITION 2,22
010660 PRINT [INVERSE]"<<<      HIT ANY
 KEY TO CONTINUE  >>>":K=0:POKE 764,25
5:GET #1,K:IF K<>255 THEN 10670
010661 GOTO 10660
010670 IF FM=72 THEN GRAPHICS 7:POKE 7
52,1:GOTO 608
010680 RETURN
011000 H=H+1:GOSUB 10400
011010 GOTO 607
020000 GRAPHICS 17:POKE 710,210
020005 POSITION 1,3:PRINT #6;[INVERSE]
 music theory drills "
020010 POSITION 7,8:PRINT #6;"PART 3"
020020 POSITION 3,13:PRINT #6;"musical
 slots"
020025 GOSUB 25000
020030 GRAPHICS 17:POSITION 9,3:PRINT
#6;"by"
020040 POSITION 4,10:PRINT #6;[INVERSE
]"DUANE  TUTAJ"
020050 POSITION 2,21:PRINT #6;"COPYRIG
HT  1983"
020060 GOSUB 25000
020070 FOR Z=1 TO 100:NEXT Z:RETURN
025000 REM *** PLAY ARCADE MUSIC ****
025010 RESTORE 65
025300 FOR Z=0 TO 3:READ Y:IF Y=9999 T
HEN GOTO 25320
025305 YY(Z)=Y:NEXT Z
025310 SOUND 0,YY(0),10,10:SOUND 1,YY(
1),10,4:SOUND 2,YY(2),10,4:SOUND 3,YY(
3),10,10
025315 SETCOLOR 2,INT(16*RND(1)),6
025317 GOTO 25300
025320 FOR Z=0 TO 3:SOUND Z,0,0,0:NEXT
 Z:RETURN
```

Commodore
FONT EDITOR

by Stephen Reed

Have you ever wondered how the computer knows to display the letter "A" when it's supposed to — especially since all it really understands is 1s and Øs? The answer to this comes from something called the Video Character Generator or Character Matrix.

Simply put, this is a Read Only Memory (ROM) which is filled with data that tells the computer what each character in the alphabet is supposed to look like.

Whenever the computer needs to display a "S," for example, it looks through ROM until it finds the correct data for displaying a "S" on the screen. And since this matrix of data is stored in ROM, it's ready as soon as you turn on your Commodore 64.

Suppose you could change the data in the character matrix so the computer would display your own character set instead of the original. You easily could make your screen characters appear in Old English, or even other languages such as Russian or Greek complete with their special letters. You could define special characters such as smiling faces, *Pac-man,* stars, etc. Imagine the impact of a game with a special character font to match the game theme. Well, all this can be done.

The ROM which stores this character matrix cannot be altered, but there is a vector in the Commodore 64 which tells the computer where ROM is, and this vector is in RAM, which means it can be changed.

Therefore, if we change the vector so it looks into RAM to get the character matrix, we can have our own character information right there waiting for it.

The purpose of the Commodore Font Editor is to provide an easy interface to create character fonts. The program itself is straightforward with few tricks. Type the program in slowly and try to understand each module as it is entered.

Before trying to design a character font, first develop it on paper. This allows you to verify consistancy between the letters before you go into the editor. First, make up a character matrix such as fig. 1.
Then, define your character within the box. Remember to leave one side column and either the top or bottom row empty so your characters won't touch each other on the screen. If, of course, you are defining a cursive font, you will want the characters to connect with some sort of standard so they look consistent.

Also, you should not leave blank columns if you are designing special symbols that require more than one character, like a pointing hand.

Using the Font Editor

The first thing the program does after you type "RUN" is to load the character area with the data stored in ROM. This takes about 3Ø seconds and is only done the first time you type RUN. If you stop the program and type RUN a second time, the program will not get the ROM characters again. This is designed to prevent the program from wiping out a font you've been designing. The program checks for this by looking at the top row of the @ or character code Ø, so don't alter the top row of this character.

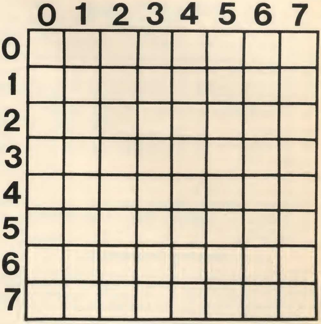


Fig. 1: Sample Character Matrix

After initializing the system, the program places the main menu on line. From this point, you can go anywhere in the editor.

Note: The character codes displayed and the inputs requested are not ASCII, but rather the Commodore Screen Display Codes, given on page 132 of *C64 Users Guide* and page 376 of the *C64 Programmers Reference Guide*.

Also, the character buffer for the font editor is from 12288 to 14335, (3ØØØH to 37FFH). The program and variable areas are from 2Ø48 to 12287, (8ØØH to 3ØØØH). The program protects the character buffer from BASIC so your characters will not get wiped out by accident.

Saving a Character Font

After selecting Item 1 on the main menu, you will be asked for a filespec. After putting a disk into the drive, type the name you want your font called. The name must be a valid Commodore filespec and can be no longer than eight characters. After you depress RETURN, the program will save your font to disk. The file format is a valid "PGM" file that loads into the character data area defined. Once a file has been developed and SAVEd, all you need to do to use it in your own programs is:

1) Protect the matrix area.
2) Set the matrix vector with: POKE 53272, (PEEK 53272) AND 24Ø) + 12.
3) Load the font file.

Loading a Character Font

After choosing main menu Item 2, enter the name of the file you want to load. This file must exist on the disk in drive 8. The program will then load the file, displaying the current character as it loads. When finished, you will return to the main menu.

Editing a Character

After selecting this option, you are asked which character you wish to edit—simply type your choice. You then will be given a sub-menu of commands to use for editing a given character.

- The E key will erase the character so you can start from scratch.
- The * and the . keys will set the pixel underneath the cursor.
- The space bar will reset or erase the pixel underneath the cursor.
- To move the cursor, use the cursor control keys in the normal manner.
- When finished with a character, depress RETURN and the new character data will be placed into the matrix.

Moving Characters

This function (menu Item 4) is used to move characters around in the character set. Be careful to avoid overlapping character areas. To use this option, you enter three values:

1) The first character to move.
2) The last character to move.
3) The first character of the area to move, too.

The system will then update the character matrix.

Note: This routine does not have wrap-around when moving characters, so keep this in mind.

Swapping Characters

Swapping characters (menu Item 5) is exactly like Item 4 (Move Characters) except the two areas affected are swapped, not overwritten as in Item 4.

Inverse Characters

This option allows you to create inverse characters from normal characters and vice versa. The parameters required are a starting and an ending character number. Once you develop your own font, simply move it into the inverse area, then reverse it. This will give you your font in the normal Commodore fashion.

Variable Table

R(Ø-7)	=	Powers of 2, 128...1
CC	=	Current character being displayed
OC	=	Previous character displayed
A$	=	Keyboard scan, etc.
O$	=	Print offset to place text on screen
B	=	Byte value of current location
ZP	=	Flag to indicate pixel set/reset
W	=	Result of numeric input routine
N$	=	File name of disk file
U	=	Value of data in editing window
X$	=	Horizontal printing offset
Y$	=	Vertical printing offset
YT	=	Flag for common routine usage

Downloading Characters

This routine allows you to selectively replace areas of your font with the character data which is stored in ROM. A starting and ending character input is required.

Increment/Decrement Display

These options (menu Items 8 and 9) allow you to scan through the characters so you can examine them close-up in the editing window.

Font Editor

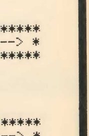
```
 1Ø REM COMMODORE FONT EDITOR
 2Ø REM PROGRAMMED BY STEPHEN REED
13Ø REM
14Ø REM ****************************
15Ø REM * INITIALIZE PROGRAM DATA *
16Ø REM ****************************
17Ø REM
18Ø POKE53281,1:POKE5328Ø,7
```

```
19Ø POKE52,48:POKE56,48:CLR:DIMR$(7)
2ØØ FORA=ØTO7:READR(A):NEXTA
21Ø DATA 128,64,32,16,8,4,2,1
22Ø PRINT"<CLR>"CHR$(142)CHR$(8)"<CTL-7>
<CTL-9>";
23Ø CC=1:OC=1
24Ø PRINT"              COMMODORE FONT EDITOR
          ";
25Ø PRINT"<CTL-1><CTL-9><COM-T><COM-T><C
OM-T><COM-T><COM-T><COM-T><COM-T><COM-T>
<COM-T><COM-T><COM-T><COM-T><COM-T><COM-
```

```
T><COM-T><COM-T><COM-T><COM-T><COM-T><CO
M-T><COM-T><COM-T><COM-T><COM-T><COM-T><
COM-T><COM-T><COM-T><COM-T><COM-T><COM-T
><COM-T><COM-T><COM-T><COM-T><COM-T><COM
-T><COM-T><COM-T><COM-T>"
260 GOSUB510:REM GET E WINDOW
270 GOSUB740:REM MENU ON SCREEN
280 GOSUB920:REM DISPLAY CHARACTERS
290 GOSUB1070:REM UPDATE SCRN FOR CC
300 IFPEEK(12288)=60THEN430
310 REM
320 REM ***************************
330 REM * DOWNLOAD ROM CHARACTERS *
340 REM ***************************
350 REM
360 PRINT"<HOME><CURDN><CURDN><CURDN><CU
RDN><CURDN><CURDN><CURDN><CURDN><CURDN><
CURDN><CURDN><CURDN><CURDN><CURDN><CURDN
><CURDN>";O$;"<CTL-7>PLEASE WAIT 30 SECO
NDS"
370 POKE56334,PEEK(56334)AND254
380 POKE1,PEEK(1)AND251
390 FORA=0TO2047
400 POKEA+12288,PEEK(A+53248):NEXT
410 POKE1,PEEK(1)OR4
420 POKE56334,PEEK(56334)OR1:GOSUB1070
430 GOSUB740
440 POKE53272,(PEEK(53272)AND240)+12
450 GOSUB1390:IFA$<"0"ORA$>"9"THEN450
460 A=VAL(A$):IFA=0THENA=10
470 ONAGOSUB1700,1890,2070,2430,2620,275
0,2920,3010,3170,3240
480 GOTO 450
490 REM
500 REM ******************************
510 REM * PUT EDIT WINDOW ON SCREEN  *
520 REM ******************************
530 REM
540 PRINT"<HOME><CURDN><CURDN><CURDN> <C
TL-1><CTL-9>";
550 PRINT"   12345678   "
560 PRINT" <CTL-9> <CTL-5><SHF-M>
   <SHF-N><CTL-1> "
570 PRINT" <CTL-9> <CTL-5> <SHF-M>
  <SHF-N> <CTL-1> "
580 PRINT" <CTL-9>1<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>1"
590 PRINT" <CTL-9>2<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>2"
600 PRINT" <CTL-9>3<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>3"
610 PRINT" <CTL-9>4<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>4"
620 PRINT" <CTL-9>5<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>5"
630 PRINT" <CTL-9>6<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>6"
640 PRINT" <CTL-9>7<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>7"
650 PRINT" <CTL-9>8<CTL-5>  <CTL-0>
   <CTL-9>  <CTL-1>8"
660 PRINT" <CTL-9> <CTL-5> <SHF-N>
  <SHF-M> <CTL-1> "
670 PRINT" <CTL-9> <CTL-5><SHF-N>
   <SHF-M><CTL-1> "
680 PRINT" <CTL-9>   12345678   <CTL-0>"
:RETURN
690 REM
700 REM ***************************
710 REM * PUT MAIN MENU ON SCREEN *
720 REM ***************************
730 REM
740 PRINT"<HOME><CURDN><CURDN><CURDN><CU
RRT><CURRT><CURRT><CURRT><CURRT><CURRT><
CURRT><CURRT><CURRT><CURRT><CURRT><CURRT
><CURRT><CURRT><CURRT><CURRT><CURRT><CUR
RT><CURRT><CURRT><CURRT><CURRT><CTL-3><C
TL-9>*** MENU ***<CTL-1>"
750 O$="<CURRT><CURRT><CURRT><CURRT><CUR
RT><CURRT><CURRT><CURRT><CURRT><CURRT><C
URRT><CURRT><CURRT><CURRT><CURRT><CURRT>
<CURRT>"
760 PRINT:PRINTO$;"1] SAVE FONT"
770 PRINTO$;"2] LOAD FONT"
780 PRINTO$;"3] EDIT CHARACTER"
790 PRINTO$;"4] MOVE CHARACTERS"
800 PRINTO$;"5] SWAP CHARACTERS"
810 PRINTO$;"6] INVERSE CHARACTERS"
820 PRINTO$;"7] GET ROM CHARACTERS"
830 PRINTO$;"8] DISPLAY DATA VALUES"
840 PRINTO$;"9] DECREMENT DISPLAY"
850 PRINTO$;"0] INCREMENT DISPLAY"
860 PRINT:PRINTO$;"<CTL-6>   SELECT OPTI
ON?      ":RETURN
870 REM
880 REM ******************************
890 REM * PUT CHARACTER DATA ON LINE *
900 REM ******************************
910 REM
920 PRINT"<HOME><CURDN><CURDN><CURDN><CU
RDN><CURDN><CURDN><CURDN><CURDN><CURDN><
CURDN><CURDN><CURDN><CURDN><CURDN><CURDN
><CURDN><CURDN><CURDN>";
930 PRINT"                           ";
940 PRINT"<CTL-1>@ABCDEFGHIJKLMNO";
950 PRINT"PQRSTUVWXYZ[£]↑← !"+CHR$(34)+"
#$%&'()*+,-./01234567";
960 PRINT"89:;<=>?<SHF-*><SHF-A><SHF-B><
SHF-C><SHF-D><SHF-E><SHF-F><SHF-G><SHF-H
><SHF-I><SHF-J><SHF-K><SHF-L><SHF-M><SHF
-N><SHF-O><SHF-P><SHF-Q><SHF-R><SHF-S><S
HF-T><SHF-U><SHF-V><SHF-W><SHF-X><SHF-Y>
<SHF-Z><SHF-+><COM--><SHF-->π<COM-*>"
970 FORA=94TO255
980 POKE1024+18*40+24+A,A
990 POKE55296+18*40+24+A,0
1000 NEXT
1010 PRINTCHR$(146):RETURN
1020 REM
1030 REM ******************************
1040 REM * PUT CURRENT DATA ON SCREEN *
1050 REM ******************************
1060 REM
1070 POKE55296+18*40+OC+24,0:OC=CC
1080 POKE55296+18*40+CC+24,2
1090 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN><CURDN><CURDN><CURDN><CURDN><CURDN>
<CURDN><CURDN><CURDN><CURDN><CURDN><CURD
N><CURRT><CURRT><CURRT><CURRT><CURRT><CT
L-5><CTL-9>CODE ";RIGHT$(STR$(CC),LEN(ST
R$(CC))-1);
1100 IFCC<10THENPRINT" "
1110 IFCC<100THENPRINT" "
1120 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN><CURDN><CURDN><CTL-0>";:FORA=0TO7:P
$=""
1130 B=PEEK(12288+A+(CC*8))
1140 IFB=>128THENZP=1:B=B-128
1150 GOSUB1320
1160 IFB=>64THENZP=1:B=B-64
1170 GOSUB1320
1180 IFB=>32THENZP=1:B=B-32
```

```
1190 GOSUB1320
1200 IFB=>16THENZP=1:B=B-16
1210 GOSUB1320
1220 IFB=>8THENZP=1:B=B-8
1230 GOSUB1320
1240 IFB=>4THENZP=1:B=B-4
1250 GOSUB1320
1260 IFB=>2THENZP=1:B=B-2
1270 GOSUB1320
1280 IFB=1THENZP=1
1290 GOSUB1320
1300 PRINT"<CTL-1><CURRT><CURRT><CURRT><
CURRT>";P$:NEXTA
1310 RETURN
1320 IFZP=1THENP$=P$+"<SHF-Q>":ZP=0:RETU
RN
1330 P$=P$+" ":RETURN
1340 REM
1350 REM ***************************
1360 REM * SCAN KEYBOARD FOR INPUT *
1370 REM ***************************
1380 REM
1390 GETA$:IFA$=""THEN1390
1400 RETURN
1410 REM
1420 REM ***************************
1430 REM *     ERASE MENU AREA     *
1440 REM ***************************
1450 REM
1460 PRINT"<HOME><CURDN><CURDN><CURDN>";
1470 FORA=0TO13
1480 PRINTO$;"                           "
1490 NEXTA:RETURN
1500 REM
1510 REM ***************************
1520 REM * FORMATTED NUMERIC INPUT *
1530 REM ***************************
1540 REM
1550 PRINT"<COM-D><COM-D><COM-D><CURLF><
CURLF><CURLF>";:Z=0:B$=""
1560 FORA=1TO3:GOSUB1600:IFZ=1THEN1580
1570 NEXT
1580 W=VAL(B$):IFW>255THENPRINT"<CURLF><
CURLF><CURLF>";:GOTO1550
1590 RETURN
1600 GOSUB1390
1610 IFA$=CHR$(13)THENZ=1:RETURN
1620 IFA$<"0"ORA$>"9"THEN1600
1630 B$=B$+A$
1640 PRINTA$;:RETURN
1650 REM
1660 REM **************************
1670 REM * SAVE FONT DATA TO DISK *
1680 REM **************************
1690 REM
1700 GOSUB1430
1710 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN>"O$"  <CTL-6>...SAVE DATA...<CTL-1>
"
1720 PRINT:PRINT"<CURDN><CURDN><CURDN><C
URDN><CURDN><CURDN><CURDN><CURDN><CURDN>
<CURDN>"O$"FILESPEC";
1730 INPUTN$:IFLEN(N$)>8THEN1700
1740 OPEN 1,8,12,N$+",P,W"
1750 PRINT#1,CHR$(0)+CHR$(48);
1760 FORI=0TO255
1770 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN><CURDN><CURDN><CURDN><CURDN>"O$"<CU
RRT><CURRT><CURRT><CURRT><CURRT><CURRT><
CURRT><CURRT>";I
1780 B$=""
1790 FORJ=0TO7
1800 B$=B$+CHR$(PEEK(12288+J+I*8)):NEXT
1810 PRINT#1,B$;
1820 NEXTI:CLOSE1:GOSUB1430
1830 GOSUB740:RETURN
1840 REM
1850 REM ****************************
1860 REM * LOAD FONT DATA FROM DISK *
1870 REM ****************************
1880 REM
1890 GOSUB1430
1900 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN>"O$"  <CTL-6>...LOAD DATA...<CTL-1>
"
1910 PRINT:PRINT"<CURDN><CURDN><CURDN><C
URDN><CURDN><CURDN><CURDN><CURDN><CURDN>
<CURDN>"O$"FILESPEC";
1920 INPUTN$:IFLEN(N$)>8THEN1700
1930 OPEN 1,8,12,N$+",P,R"
1940 GET#1,A$,A$
1950 FORI=0TO255
1960 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN><CURDN><CURDN><CURDN><CURDN>"O$"<CU
RRT><CURRT><CURRT><CURRT><CURRT><CURRT><
CURRT><CURRT>";I
1970 FORJ=0TO7
1980 GET#1,A$:A=0:IFA$<>""THENA=ASC(A$)
1990 POKE12288+J+I*8,A:NEXTJ
2000 NEXTI:CLOSE1:GOSUB1430
2010 GOSUB740:RETURN
2020 REM
2030 REM **************************
2040 REM * EDIT CHARACTER ROUTINE *
2050 REM **************************
2060 REM
2070 GOSUB1430
2080 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN>"O$"<CTL-5>  EDIT CHARACTER<CTL-1>"
2090 PRINT:PRINTO$"CHARACTER # ";:GOSUB1
550
2100 CC=W:GOSUB1070
2110 GOSUB1430:PRINT"<HOME><CURDN><CURDN
><CURDN><CURDN>"O$;"<CTL-5>EDIT CHARACTE
R<CTL-1>"
2120 PRINT:PRINTO$"[E]        ERASE BUFFER
"
2130 PRINTO$"[*]/[.]  SET.PIXEL"
2140 PRINTO$"[SPACE]  RESET PIXEL":PRINT
O$"[RETURN] FINISHED"
2150 PRINT:PRINT:PRINTO$" CURSOR KEYS MO
VE"
2160 PRINTO$" EDITING CURSOR."
2170 X=4:Y=4:G=1:GOSUB2350
2180 GOSUB1390:IFA$="E"THENGOSUB2410:GOT
O2180
2190 IFA$=" "THENG=3:GOSUB2350:GOTO2180
2200 IFA$="*"ORA$="."THENG=2:GOSUB2350:G
OTO2180
2210 IFA$=CHR$(13)THEN2280
2220 IFA$<>"<CURUP>"ANDA$<>"<CURDN>"ANDA
$<>"<CURLF>"ANDA$<>"<CURRT>"THEN2180
2230 IFA$="<CURUP>"ANDY>1THENG=0:GOSUB24
00:Y=Y-1:GOSUB2350:GOTO2180
2240 IFA$="<CURDN>"ANDY<8THENG=0:GOSUB24
00:Y=Y+1:GOSUB2350:GOTO2180
2250 IFA$="<CURRT>"ANDX<8THENG=0:GOSUB24
00:X=X+1:GOSUB2350:GOTO2180
2260 IFA$="<CURLF>"ANDX>1THENG=0:GOSUB24
00:X=X-1:GOSUB2350:GOTO2180
2270 GOTO 2180
2280 FORA=0TO7
```

```
2290 GOSUB2320:REM GET BYTE IN B
2300 POKE12288+(CC*8)+A,B:NEXTA
2310 GOSUB1070:GOSUB1430:GOSUB740:RETURN
2320 B=0:FORR=0TO7
2330 U=PEEK(1268+(A*40)+R):IFU=87ORU=81T
HENB=B+R(R)
2340 NEXTR:RETURN
2350 X$="<CURRT><CURRT><CURRT><CURRT><CU
RRT><CURRT><CURRT><CURRT>":Y$="<CURDN><C
URDN><CURDN><CURDN><CURDN><CURDN><CURDN>
<CURDN>":PRINT"<HOME><CURDN><CURDN><CURD
N><CURDN><CURDN><CURRT><CURRT><CURRT>";L
EFT$(X$,X);LEFT$(Y$,Y);
2360 IFG=2THENPRINT"<CTL-1><SHF-Q>":G=1:
GOSUB2350:RETURN
2370 IFG=3THENPRINT"<CTL-1> ":G=1:GOSUB2
350:RETURN
2380 K=PEEK(1227+X+Y*40):IFK=81THENPRINT
"<CTL-3><SHF-W>":RETURN
2390 PRINT"<CTL-3>+":RETURN
2400 POKE1227+X+Y*40,K:POKE55499+X+Y*40,
0:RETURN
2410 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN><CURDN><CURDN>";:FORA=1TO8:PRINT"<C
URRT><CURRT><CURRT><CURRT>          "
2420 NEXTA:G=1:K=32:GOSUB2350:RETURN
2430 GOSUB1430
2440 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN>"O$"STARTING CHAR ";
2450 GOSUB1550:M1=W
2460 PRINT:PRINTO$"ENDING CHAR ";:GOSUB1
550
2470 M2=W
2480 PRINT:PRINTO$"MOVE TO CHAR ";:GOSUB
1550
2490 IFM1>M2THEN2430
2500 IFW<=M2 ANDW=>M1 THEN 2430
2510 PRINT:PRINTO$;"<CTL-7>  ...MOVING..
.<CTL-1>"
2520 FORA=0TO(M2-M1)
2530 FORB=0TO7
2540 POKE12288+B+(A+W)*8,PEEK(12288+B+(A
+M1)*8)
2550 NEXTB,A
2560 GOSUB1430:GOSUB740:RETURN
2570 REM
2580 REM ****************************
2590 REM * SWAP CHARACTER DATA SUBR *
2600 REM ****************************
2610 REM
2620 GOSUB1430:PRINT"<CTL-1><HOME><CURDN
><CURDN><CURDN><CURDN>"O$"STARTING CHAR
";:GOSUB1550:SC=W
2630 PRINT:PRINTO$"ENDING CHAR ";:GOSUB1
550:EC=W
2640 PRINT:PRINTO$"SWAP WITH ";:GOSUB155
0
2650 FORA=0TO(EC-SC):FORB=0TO7
2660 T=PEEK(12288+B+(A+W)*8)
2670 POKE12288+B+(A+W)*8,PEEK(12288+B+(A
+SC)*8)
2680 POKE12288+B+(A+SC)*8,T:NEXTB,A
2690 GOSUB1430:GOSUB740:GOSUB1070:RETURN
2700 REM
2710 REM **************************
2720 REM * REVERSE CHARACTER DATA *
2730 REM **************************
2740 REM
2750 GOSUB1430
2760 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN>";:IFYT=0THENPRINTO$"<CTL-1>REVERSE
 DATA"
2770 PRINT:PRINTO$"STARTING CHAR ";:GOSU
B1550
2780 SC=W
2790 PRINT:PRINT:PRINTO$"ENDING CHAR ";:
GOSUB1550
2800 IFW<SCTHEN2750
2810 IFYT=1THENRETURN
2820 PRINT:PRINT:PRINTO$"  <CTL-7>...REV
ERSING...<CTL-1>"
2830 FORA=0TO(W-SC):FORB=0TO7
2840 POKE12288+B+(A+SC)*8,255-PEEK(12288
+B+(A+SC)*8):NEXTB,A
2850 GOSUB1430:GOSUB740
2860 GOSUB1070:RETURN
2870 REM
2880 REM **************************
2890 REM * GET ROM CHARACTER DATA *
2900 REM **************************
2910 REM
2920 YT=1:PRINT"<CTL-1>":GOSUB2750:YT=0
2930 PRINT:PRINTO$"<CTL-7>...LOADING FRO
M ROM...<CTL-1>"
2940 POKE56334,PEEK(56334)AND254
2950 POKE1,PEEK(1)AND251
2960 FORA=SC*8TOSC*8+(W-SC)*8+7
2970 POKEA+12288,PEEK(A+53248):NEXT
2980 POKE1,PEEK(1)OR4
2990 POKE56334,PEEK(56334)OR1
3000 GOSUB1430:GOSUB740:GOSUB1070:RETURN
3010 GOSUB1430
3020 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN>"O$;"<CTL-1>CHARACTER NUMBER ";
3030 GOSUB1550
3040 PRINT:PRINT:TP=CC:CC=W:GOSUB1070
3050 PRINT"<HOME><CURDN><CURDN><CURDN><C
URDN><CURDN>":FORA=0TO7
3060 PRINTO$;PEEK(12288+A+(W*8))
3070 NEXTA
3080 PRINT:PRINTO$;"     SPACE TO RETURN"
3090 GOSUB1390:CC=TP:GOSUB1070
3100 GOSUB1430:GOSUB740
3110 GOSUB1070:RETURN
3120 REM
3130 REM ***************************
3140 REM * DECREMENT CHARACTER --> *
3150 REM ***************************
3160 REM
3170 CC=CC-1:IFCC<0THENCC=255
3180 GOSUB1070:RETURN
3190 REM
3200 REM ***************************
3210 REM * INCREMENT CHARACTER --> *
3220 REM ***************************
3230 REM
3240 CC=CC+1:IFCC>255THENCC=1
3250 GOTO 3180
```



Olympia®

COMBINATION "DAISY WHEEL" PRINTER / TYPEWRITER

"WORLD'S FINEST"
LIST $799.00
SALE $489.00

- SUPERB COMPUTER PRINTER COMBINED WITH WORLD'S FINEST ELECTRONIC TYPEWRITER!
- BETTER THAN IBM SELECTRIC — USED BY WORLD'S LARGEST CORPORATIONS!
- TWO MACHINES IN ONE — JUST A FLICK OF THE SWITCH!
- SUPERB EXECUTIVE CORRESPONDENCE — HOME, OFFICE, WORD PROCESSING!
- EXTRA LARGE CARRIAGE — ALLOWS 14-1/8" PAPER USAGE!
- DROP IN CASSETTE RIBBON — EXPRESS LIFT OFF CORRECTION OR ERASER UP TO 46 CHARACTERS!
- PRECISION DAISY WHEEL PRINTING — MANY TYPE STYLES!
- PITCH SELECTOR — 10, 12, 15 CPS, AUTOMATIC RELOCATE KEY!
- AUTOMATIC MARGIN CONTROL AND SETTING! KEY IN BUFFER!.
- ELECTRONIC RELIABILITY, BUILT IN DIAGNOSTIC TEST!
- CENTRONICS PARALLEL OR SERIAL INTERFACE BUILT IN!
- 15 DAY FREE TRIAL — 90 DAY FREE REPLACEMENT WARRANTY!

Add $17.50 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

FLOPPY DISKS SALE *$1.19 ea.

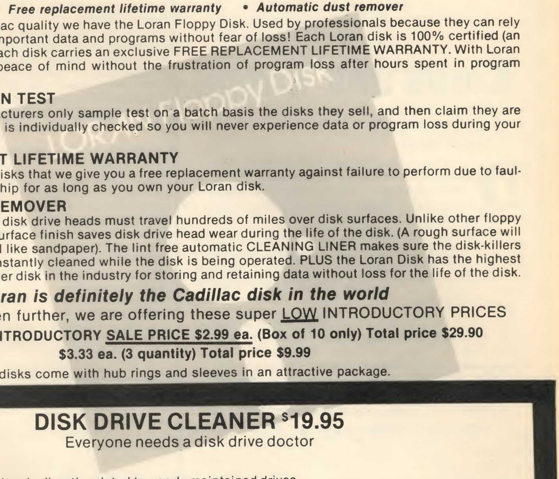
Economy Model or Cadillac Quality

LORAN CERTIFIED PERSONAL COMPUTER DISK ***We have the lowest prices!*** LORAN CERTIFIED PERSONAL COMPUTER DISK

***ECONOMY DISKS**

Good quality 5¼" single sided single density with hub rings.

Bulk Pac	100 Qty.	$1.19 ea.	Total Price	$119.00
	10 Qty.	1.39 ea.	Total Price	13.90



CADILLAC QUALITY

• Each disk certified • Free replacement lifetime warranty • Automatic dust remover

For those who want cadillac quality we have the Loran Floppy Disk. Used by professionals because they can rely on Loran Disks to store important data and programs without fear of loss! Each Loran disk is 100% certified (an exclusive process) plus each disk carries an exclusive FREE REPLACEMENT LIFETIME WARRANTY. With Loran disks you can have the peace of mind without the frustration of program loss after hours spent in program development.

100% CERTIFICATION TEST

Some floppy disk manufacturers only sample test on a batch basis the disks they sell, and then claim they are certified. Each Loran disk is individually checked so you will never experience data or program loss during your lifetime!

FREE REPLACEMENT LIFETIME WARRANTY

We are so sure of Loran Disks that we give you a free replacement warranty against failure to perform due to faulty materials or workmanship for as long as you own your Loran disk.

AUTOMATIC DUST REMOVER

Just like a record needle, disk drive heads must travel hundreds of miles over disk surfaces. Unlike other floppy disks the Loran smooth surface finish saves disk drive head wear during the life of the disk. (A rough surface will grind your disk drive head like sandpaper). The lint free automatic CLEANING LINER makes sure the disk-killers (dust & dirt) are being constantly cleaned while the disk is being operated. PLUS the Loran Disk has the highest probability rate of any other disk in the industry for storing and retaining data without loss for the life of the disk.

Loran is definitely the Cadillac disk in the world

Just to prove it even further, we are offering these super LOW INTRODUCTORY PRICES

List $4.99 ea. INTRODUCTORY SALE PRICE $2.99 ea. (Box of 10 only) Total price $29.90

$3.33 ea. (3 quantity) Total price $9.99

All disks come with hub rings and sleeves in an attractive package.

DISK DRIVE CLEANER $19.95

Everyone needs a disk drive doctor

FACTS

- 60% of all drive downtime is directly related to poorly maintained drives.
- Drives should be cleaned each week regardless of use.
- Drives are sensitive to smoke, dust and all micro particles.
- Systematic operator performed maintenance is the best way of ensuring error free use of your computer system.

The Cheetah disk drive cleaner can be used with single or double sided 5¼" disk drives. The Cheetah is an easy to use fast method of maintaining efficient floppy diskette drive operation.

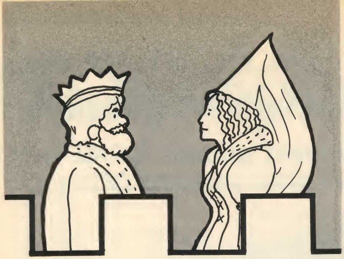
The Cheetah cleaner comes with 2 disks and is packed in a protective plastic folder to prevent contamination.

List $29.95 / Sale $19.95

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order



Lords and Ladys

by Stephen Reed

During the Middle Ages, life was short and simple. Seldom did anyone live past the age of 40, therefore earning a government post represented a very substantial accomplishment.

Of course, governments then were not what they are today. If you were not fighting your neighbors, you were probably busy fighting starvation or disease. Economies were based on little more than how much wheat was harvested each year.

Sounds simple???

Just remember, in those days, rulers were overthrown so fast that most of the subjects never knew who the local leader was!

Welcome to *Lords & Ladys,* where you have been given rule of a small territory. If you rule well and your state prospers, you will be promoted within the government.

Your goal is to become king of your assigned country.

This game supports from one to eight players and the first to become king of course wins. But life is short — in this game — and it's possible no one will live long enough to become king.

The game is straightforward and doesn't need instructions. I've purposely not told you the best ways to play because learning how to rule is the whole game in a nutshell.

Games of this nature have been around longer than home computers. The first Hammurabi or monarchy program was probably developed on a PDP system back in the 60s. But credit must be given here to two previous efforts from which this program is based. *Ruler* has been around since at least 1976 and its presentation inspired most economic programs to follow. Secondly, *Santa Paravia* was a significant contribution to economic simulation software. Originally developed on a Radio Shack Model 1, *Santa Paravia* was created about 1978.

Well, that's enough for you to go on. Take your time typing in this program — and enjoy.

Lords and Ladys

```
 10 REM LORDS & LADYS
 20 REM (C) 1984 STEPHEN REED
120 REM
130 GOTO 220
140 PRINT"<CTL-7><COM-T><COM-T><COM-T><C
OM-T><COM-T><COM-T><COM-T><COM-T><COM-T>
<COM-T><COM-T><COM-T><COM-T><COM-T><COM-
T><COM-T><COM-T><COM-T><COM-T><COM-T><CO
M-T><COM-T><COM-T><COM-T><COM-T><COM-T><
COM-T><COM-T><COM-T><COM-T><COM-T><COM-T
><COM-T><COM-T><COM-T><COM-T><COM-T><COM
-T><COM-T><COM-T>";
150 RETURN
160 PRINTTAB(7)"<CTL-1><CTL-9><PRESS ANY
 KEY TO CONTINUE><CTL-0>"
170 GETA$:IFA$=""THEN170
180 RETURN
190 PRINT"<CLR>"CHR$(8)CHR$(142)"<CTL-8>
<COM-@><COM-@><COM-@><COM-@><COM-@><COM-
@><COM-@><COM-@><COM-@><COM-@><COM-@><CO
M-@><COM-@><COM-@><COM-@><COM-@><COM-@><
COM-@><COM-@><COM-@><COM-@><COM-@><COM-@
><COM-@><COM-@><COM-@><COM-@><COM-@><COM
-@><COM-@><COM-@><COM-@><COM-@><COM-@><C
OM-@><COM-@><COM-@><COM-@><COM-@><COM-@>
";
200 PRINT"<CTL-9><CTL-6>                LOR
DS & LADYS               <CTL-0>";
210 PRINT"<CTL-8><COM-T><COM-T><COM-T><C
OM-T><COM-T><COM-T><COM-T><COM-T><COM-T>
<COM-T><COM-T><COM-T><COM-T><COM-T><COM-
T><COM-T><COM-T><COM-T><COM-T><COM-T><CO
M-T><COM-T><COM-T><COM-T><COM-T><COM-T><
COM-T><COM-T><COM-T><COM-T><COM-T><COM-T
><COM-T><COM-T><COM-T><COM-T><COM-T><COM
-T><COM-T><COM-T>":RETURN
220 Y(0)=1200
230 DATA PEASANT,WORKMAN,SIR,BARON,COUNT
,DUKE,PRINCE,KING
240 DATA SAMARIA,PERSIA,ROMAGNA,ARABIA,T
ORTICIA,ALBANIA,MARAVIA
250 DATA PALODIA
260 RESTORE:POKE 53281,0
270 POKE53280,0:GOSUB190
280 FORA=1TO8
290 READA$
300 NEXT
310 INPUT"<CTL-2>HOW MANY PLAYERS <1-8>"
;F
320 IFF<1ORF>8THEN260
330 PRINT:FORA=1TOF:READT$(A)
340 PRINT"<CTL-7>WHO SHALL RULE ";T$(A);
350 INPUTN$:PRINT
360 IFLEN(N$)>10THEN340
370 N$(A)=N$+" OF "+T$(A)
380 G(A)=10
390 H(A)=10
400 I(A)=10
410 J(A)=2
420 O(A)=1220+INT(RND(1)*30)
430 K(A)=1500
440 L(A)=12000
450 R(A)=4000
460 T(A)=1
470 U(A)=1
480 N(A)=5
```

```
490 P(A)=25
500 Q(A)=5
510 M(A)=20
520 S(A)=1800
530 NEXT
540 FORA=1TOF
550 RESTORE
560 B=T(A)
570 FORC=1TOB
580 READ T$(A)
590 NEXT
600 NEXT
610 GOSUB190
620 INPUT"<CTL-8>DO YOU WANT INSTRUCTION
S";A$
630 IFLEFT$(A$,1)="Y"THENGOSUB5170
640 POKE 53281,0:POKE53280,0
650 GOSUB190:PRINTTAB(13)"<CTL-7>1. <CTL
-2>NOVICE"
660 PRINTTAB(13)"<CTL-7>2. <COM-8>AMATUR
E"
670 PRINTTAB(13)"<CTL-7>3. <COM-5>PROFES
SIONAL"
680 PRINTTAB(13)"<CTL-7>4. <COM-4>BEST T
HERE IS"
690 PRINT
700 INPUT"<CTL-3>          HOW GOOD A PLAY
ER ARE YOU";U
710 U(0)=INT(U)
720 IFU<1THEN U(0)=1
730 IFU>4THEN U(0)=4
740 U(0)=U(0)+5
750 E=E+1
760 IFT(E)=-1THENE=E+1
770 IFT(1)<1ANDT(2)<1ANDT(3)<1ANDT(4)<1A
NDT(5)<1ANDT(6)<1ANDT(7)<1ANDT(8)<1THEN5
410
780 IFE>FTHENE=0:Y(0)=Y(0)+1:GOTO750
790 IFY(0)>O(E)THEN750
800 IFY(0)=O(E)THEN970
810 GOSUB1250
820 GOSUB1960
830 GOSUB2630
840 GOSUB2320
850 GOSUB3260
860 GOSUB4060
870 GOSUB4470
880 GOTO 750
890 POKE 53281,6:POKE53280,4
900 PRINT"<CTL-1><CTL-9><CLR>LAND  NOBLE
S SERFS CLERGY MERCHANTS ARMY";
910 PRINT:PRINT:FORA=1TOF:PRINT"<COM-5>"
T$(A)" "N$(A)
920 L$=STR$(L(A))
930 L$="<CTL-2>"+RIGHT$(L$,LEN(L$)-1)
940 PRINTL$TAB(7)N(A)TAB(12)S(A)TAB(21)Q
(A)TAB(28)M(A)TAB(35)P(A)
950 NEXT
960 PRINT:GOSUB160:RETURN
970 POKE 53281,2:POKE53280,2
980 PRINT"<CLR><CTL-9><CTL-1>
                                ";
990 PRINT"         <CTL-2>TERRIBLE NEWS HA
S ARRIVED<CTL-1>          ";
1000 PRINT"<CTL-1>
                         ";
1010 PRINT:PRINT"<CTL-8>"T$(E)" "N$(E)
1020 PRINT"HAS JUST DIED"
1030 T(E)=-1:Y=INT(RND(8)*8)+1
1040 IFY(0)>1450THENPRINT"OF OLD AGE AFT
ER A LONG REIGN.":GOTO1100
1050 IFY<4THENPRINT"OF FROSTBITE AFTER A
 COLD SNOW."
1060 IFY=5THENPRINT"IN A LOCAL PLAGUE EP
IDEMIC."
1070 IFY=4THENPRINT"OF INJURYS AFTER FAL
LING OFF A HOURSE."
1080 IFY=6THENPRINT"AFTER BEING ATTACKED
 BY ASSASSINS."
1090 IFY>6THENPRINT"OF AN UNDIAGNOSED IL
LNESS."
1100 PRINT
1110 EG=1
1120 GOSUB160
1130 GOSUB3260
1140 GOSUB890
1150 EG=0
1160 GOTO750
1170 I1=INT(I1):RETURN
1180 C1=INT(C1):RETURN
1190 S1=INT(S1):RETURN
1200 K(E)=INT(K(E)):RETURN
1210 Z=(INT(RND(A)*A)+1)*S(E)/100:Z2=Z:P
RINT"<CTL-6>"INT(Z2)"SERFS BORN THIS YEA
R."
1220 S(E)=S(E)+Z2:RETURN
1230 Z=(INT(RND(A)*A)+1)*S(E)/100:Z2=Z:P
RINT"<CTL-3>"INT(Z2)"SERFS DIED THIS YEA
R."
1240 S(E)=S(E)-Z2:RETURN
1250 W=(INT(RND(5)*5)+2+INT(RND(6)*6))/2
1260 ONWGOTO1270,1280,1290,1300,1310
1270 W$="DROUGHT":GOTO1320
1280 W$="BAD WEATHER":GOTO1320
1290 W$="AVERAGE WEATHER":GOTO1320
1300 W$="FINE WEATHER":GOTO1320
1310 W$="GREAT WEATHER":GOTO1320
1320 R=INT(RND(50)*50)+1
1330 R(E)=(R(E)*100-R(E)*R)/100
1340 X=L(E):Y=(S(E)-D(E)*100)*5
1350 IFY<0THENY=0
1360 IFY<XTHENX=Y
1370 Y=R(E)*2:IFY<XTHENX=Y
1380 Y=W-.5:H1=X*Y:R(E)=R(E)+H1
1390 D1=N(E)*100+C(E)*40+M(E)*30+P(E)*10
+S(E)*5
1400 L=(3*W+INT(RND(6)*6)+2+INT(RND(6)*6
)+10)/10
1410 IFH1<1THENY=2:GOTO1430
1420 Y=D1/H1:IFY>2THENY=2
1430 IFY<.8THENY=.8
1440 L=L*Y:L=INT(L*10):L=L/10
1450 Z=6-W
1460 G=(Z*5+INT(RND(5)*5)+INT(RND(5)*5))
/5*Y*20
1470 RETURN
1480 PRINT
1490 PRINT"<COM-2>RATS ATE"R"<CURLF>% OF
 YOUR STORED WHEAT."
1500 PRINTTAB(5)"<CTL-7>"W$
1510 PRINT"<CTL-1><SHF-Z> YEARS HARVEST:
"INT(H1)"BUSHELS <SHF-Z>"
1520 PRINT:IFK(E)<32766THENGOSUB1200
1530 R$=STR$(INT(R(E))):R$=RIGHT$(R$,LEN
(R$)-1):IFR(E)<0THENR$="   0"
1540 D$=STR$(INT(D1)):D$=RIGHT$(D$,LEN(D
$)-1)
1550 G$=STR$(INT(G)):G$=" "+G$:L$=STR$(L
):L$="   "+L$
1560 K$=STR$(INT(K(E)))
1570 PRINT"<CTL-1><COM-A><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><COM-R
><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF
-C><COM-R><SHF-C><SHF-C><SHF-C><SHF-C><S
HF-C><SHF-C><SHF-C><COM-R><SHF-C><SHF-C>
<SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-
C><COM-R><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><SHF-C><COM-S>";
1580 PRINT"<SHF-B> WHEAT <SHF-B>WHEAT <S
HF-B>COST OF<SHF-B>PRICE OF<SHF-B>$$$$$$
<SHF-B>";
1590 PRINT "<SHF-B>RESERVE<SHF-B>DEMAND<
SHF-B> WHEAT <SHF-B>  LAND  <SHF-B>
 <SHF-B>";
1600 PRINT"<CTL-1><COM-Q><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-+
><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF
-C><SHF-+><SHF-C><SHF-C><SHF-C><SHF-C><S
HF-C><SHF-C><SHF-C><SHF-+><SHF-C><SHF-C>
<SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-
C><SHF-+><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><SHF-C><COM-W>";
1610 PRINT"<SHF-B>"R$TAB(8)"<SHF-B>"D$TA
B(15)"<SHF-B>"G$TAB(23)"<SHF-B>"L$TAB(32
)"<SHF-B>"K$TAB(39)"<SHF-B>";
1620 PRINT"<CTL-1><COM-Q><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-+
><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF
-C><SHF-+><SHF-C><SHF-C><SHF-C><SHF-C><S
HF-C><SHF-C><SHF-C><SHF-+><SHF-C><SHF-C>
<SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-
C><SHF-+><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><SHF-C><COM-W>";
1630 PRINT"<SHF-B>BUSHELS<SHF-B>BUSH. <S
HF-B>1000 BL<SHF-B>  ACRE  <SHF-B> GOLD
<SHF-B>";
1640 PRINT"<CTL-1><COM-Z><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><COM-E
><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF
-C><COM-E><SHF-C><SHF-C><SHF-C><SHF-C><S
HF-C><SHF-C><SHF-C><COM-E><SHF-C><SHF-C>
<SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-
C><COM-E><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><SHF-C><COM-X>";
1650 RETURN
1660 J=(J(E)*300-500)*T(E)
1670 ONJ(E)GOTO1680,1700,1720,1740
1680 J$=" GRACIOUS"
1690 GOTO 1750
1700 J$="  FAIR"
1710 GOTO 1750
1720 J$=" UNJUST"
1730 GOTO 1750
1740 J$="TERRIBLE"
1750 Y=150-G(E)-H(E)-I(E)
1760 IFY<1THENY=1
1770 C1=(N(E)*180+Q(E)*75+M(E)*20)*(Y/10
0)+U(E)*100
1780 S1=(N(E)*50+M(E)*25+U(E)*19)*(Y/100
)*(5-J(E))/2
1790 I1=N(E)*250+U(E)*20+(10*J(E)*N(E))*
(Y/100)
1800 C1=C1*G(E)/100
1810 IFC1<32760THENGOSUB1180
1820 S1=S1*H(E)/100
1830 IFS1<32760THENGOSUB1190
1840 I1=I1*I(E)/100
1850 IFI1<32760THENGOSUB1170
```

```
1860 PRINT"<CTL-1><COM-A><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C
><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF
-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><S
HF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C>
<SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-
C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><COM-S>"
1870 PRINT"<SHF-B> TAX REVENUES"J+C1+S1+
I1"GOLD PIECES."TAB(38)"<SHF-B>"
1880 PRINT"<CTL-1><COM-Q><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C
><SHF-C><COM-R><SHF-C><SHF-C><SHF-C><SHF
-C><SHF-C><SHF-C><SHF-C><COM-R><SHF-C><S
HF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C>
<SHF-C><SHF-C><COM-R><SHF-C><SHF-C><SHF-
C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><COM-W>"
1890 PRINT"<SHF-B> CUSTOMS <SHF-B> SALES
 <SHF-B>  INCOME <SHF-B> JUSTICE <SHF-B>
"
1900 PRINT"<SHF-B>  DUTY   <SHF-B>  TAX
 <SHF-B>   TAX   <SHF-B>  LEVEL  <SHF-B>
"
1910 PRINT"<CTL-1><COM-Q><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C
><SHF-C><SHF-+><SHF-C><SHF-C><SHF-C><SHF
-C><SHF-C><SHF-C><SHF-C><SHF-+><SHF-C><S
HF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C>
<SHF-C><SHF-C><SHF-+><SHF-C><SHF-C><SHF-
C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><COM-W>"
1920 PRINT"<SHF-B>  "G(E)"<CURLF>%"TAB(1
0)"<SHF-B> "H(E)"<CURLF>%"TAB(18)"<SHF-B
>  "I(E)"<CURLF>%"TAB(28)"<SHF-B>"J$TAB(
38)"<SHF-B>"
1930 PRINT"<SHF-B>  "C1TAB(10)"<SHF-B> "
S1TAB(18)"<SHF-B>  "I1TAB(28)"<SHF-B>  "
JTAB(38)"<SHF-B>"
1940 PRINT"<CTL-1><COM-Z><SHF-C><SHF-C><
SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C
><SHF-C><COM-E><SHF-C><SHF-C><SHF-C><SHF
-C><SHF-C><SHF-C><SHF-C><COM-E><SHF-C><S
HF-C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C>
<SHF-C><SHF-C><COM-E><SHF-C><SHF-C><SHF-
C><SHF-C><SHF-C><SHF-C><SHF-C><SHF-C><SH
F-C><COM-X>"
1950 RETURN
1960 POKE 53280,7:POKE53281,7
1970 PRINT"<CLR><CTL-1>"T$(E)" "N$(E)
1980 GOSUB140
1990 GOSUB1480
2000 PRINT:PRINT" [1] BUY WHEAT
  [3] BUY LAND"
2010 PRINT" [2] SELL WHEAT            [4]
SELL LAND"
2020 PRINT"<CTL-7>    SELECT OPTION (0=C
ONTINUE)";:INPUTI
2030 IFI>4THEN1960
2040 IFI<1THENRETURN
2050 ONIGOTO2060,2150,2210,2260
2060 INPUT"<CTL-5>HOW MUCH WHEAT TO BUY"
;I
2070 IFI<1THEN2100
2080 K(E)=K(E)-(I*G/1000)
2090 R(E)=R(E)+I
2100 PRINT"<CLR><CTL-1>"T$(E)" "N$(E)
2110 GOSUB140
2120 PRINT:PRINT:PRINT
2130 PRINT:GOSUB1520
```

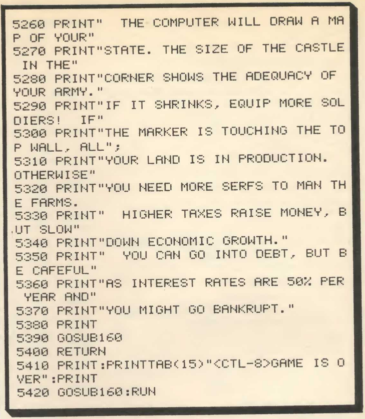
```
2140 GOTO 2000
2150 INPUT"<CTL-5>HOW MUCH WHEAT TO SELL
";I
2160 IFI<1THEN2100
2170 IFI>R(E)THENPRINT"<CTL-3>YOU DON'T
HAVE THAT MUCH":GOTO2150
2180 K(E)=K(E)+(I*G/1000)
2190 R(E)=R(E)-I
2200 GOTO 2100
2210 INPUT"<CTL-5>HOW MANY ACRES TO BUY"
;I
2220 IFI<0THEN2100
2230 L(E)=L(E)+I
2240 K(E)=K(E)-(I*L)
2250 GOTO 2100
2260 INPUT"<CTL-5>HOW MANY ACRES TO SELL
";I
2270 IFI<0THEN2100
2280 IFI>(L(E)-5000)THENPRINT"<CTL-3>YOU
 CAN'T SELL THAT MUCH.":GOTO2260
2290 L(E)=L(E)-I
2300 K(E)=K(E)+(I*L)
2310 GOTO 2100
2320 PRINT"<CLR>";:POKE 53281,5:POKE5328
0,6
2330 PRINT"<CTL-8>"T$(E)" "N$(E)
2340 GOSUB140
2350 PRINT:GOSUB1660
2360 PRINT:PRINT"  [1] CUSTOMS DUTY     [
3] INCOME TAX"
2370 PRINT"  [2] SALES TAX          [4] JUS
TICE"
2380 PRINT
2390 INPUT"<CTL-7><ENTER TAX # FOR CHANG
E, 0=CONTINUE>";I
2400 IFI>4THEN2380
2410 IFI<1THEN2590
2420 ONIGOTO2430,2470,2500,2530
2430 INPUT"<CTL-2>NEW CUSTOMS DUTY (0-10
0)";I
2440 IFI>100THENI=100
2450 IFI<0THENI=0
2460 G(E)=I:GOTO2320
2470 INPUT"<CTL-2>NEW SALES TAX (0-50)";
I
2480 IF(I>50)OR(I<0)THENI=5
2490 H(E)=I:GOTO2320
2500 INPUT"<CTL-2>NEW INCOME TAX (0-25)"
;I
2510 IF(I<0)OR(I>25)THENI=0
2520 I(E)=I:GOTO2320
2530 PRINTTAB(16)"<CTL-3>JUSTICE"
2540 PRINT"<CTL-2>  [1] GRACIOUS
  [3] UNJUST"
2550 PRINT"  [2] FAIR               [4] T
ERRIBLE"
2560 INPUT"<CTL-7>WHICH LEVEL";I
2570 IF(I>4)OR(I<1)THENI=1
2580 J(E)=I:GOTO2320
2590 K(E)=K(E)+C1+S1+I1+J
2600 IFK(E)<0THENK(E)=K(E)*1.5
2610 IFK(E)<(-10000*T(E))THEN4810
2620 RETURN
2630 PRINT"<CLR><CTL-1>"T$(E)" "N$(E)
2640 GOSUB140
2650 PRINT:PRINT:PRINT:PRINT:GOSUB1520
2660 PRINT:PRINT"<CTL-7>HOW MUCH WHEAT T
O FEED"
2670 INPUT"THE PEOPLE";G
2680 IFG<(R(E)/4)THENPRINT"<CTL-3>YOU MU
ST RELEASE AT LEAST 25%":GOTO2660
2690 IFG>(R(E)*.75)THENPRINT"<CTL-3>YOU
MUST KEEP AT LEAST 25%":GOTO2660
2700 R(E)=R(E)-G
2710 PRINT"<CLR><CTL-1>"T$(E)" "N$(E):PR
INT"<CTL-3><COM-T><COM-T><COM-T><COM-T><
COM-T><COM-T><COM-T><COM-T><COM-T><COM-T
><COM-T><COM-T><COM-T><COM-T><COM-T><COM
-T><COM-T><COM-T><COM-T><COM-T><COM-T><C
OM-T><COM-T><COM-T><COM-T><COM-T><COM-T>
<COM-T><COM-T><COM-T><COM-T><COM-T><COM-
T><COM-T><COM-T><COM-T><COM-T><COM-T><CO
M-T><COM-T>"
2720 POKE53281,1:POKE53280,1
2730 Z=G/D1-1
2740 IFZ>0THENZ=Z/2
2750 IFZ>.25THENZ=Z/10+.25
2760 Z2=50-G(E)-H(E)-I(E)
2770 IFZ2<0THENZ2=Z2*J(E)
2780 Z2=Z2/10
2790 IFZ2>0THENZ2=Z2+3-J(E)
2800 Z=Z+(Z2/10)
2810 IFZ>.5THENZ=.5
2820 IFG<(D1-1)THEN3070
2830 A=7:GOSUB1210
2840 A=3:GOSUB1230
2850 IF(G(E)+H(E))<35THENM(E)=N(E)+INT(R
ND(2)*2):Q(E)=Q(E)+INT(RND(3)*3)
2860 IFI(E)<INT(RND(20)*20)+1THENN(E)=N(
E)+INT(RND(2)*2):Q(E)=Q(E)+INT(RND(3)*3)
2870 IFG<(D1+D1*.3)THEN3010
2880 Z2=S(E)/1000
2890 Z=(G-D1)/D1*10
2900 Z=Z*Z2*INT(RND(25)*25+1)+INT(RND(40
)*40+1)
2910 IFZ>32000THENZ=32000
2920 Z2=Z:Z=INT(RND(4)*Z2)+1
2930 PRINT"<CTL-1>"Z"SERFS IMMIGRATE INT
O THE COUNTRY."
2940 S(E)=S(E)+Z
2950 Z2=Z/5
2960 Z=INT(RND(5)*Z2)+1
2970 IFZ>50THENZ=50
2980 M(E)=M(E)+Z
2990 N(E)=N(E)+1
3000 Q(E)=Q(E)+2
3010 IFJ(E)<3THEN3060
3020 J1=S(E)/100*(J(E)-2)*(J(E)-2)
3030 J1=INT(RND(8)*J1)+1
3040 S(E)=S(E)-J1
3050 PRINT"<CTL-4>"J1"SERFS LEAVE UNJUST
 RULE."
3060 GOTO 3130
3070 X=(D1-G)/D1*100-9
3080 IFX>65THENX=65
3090 IFX<0THENX2=0:X=0
3100 A=3:GOSUB1210
3110 A=X2+8:GOSUB1230
3120 GOTO 3010
3130 Z=A(E)*75
3140 K(E)=K(E)+Z
3150 IFZ>0THENPRINT:PRINT"<CTL-7> BAZZAR
 EARNED"Z"GOLD PIECES."
3160 IFS(E)<32766THENS1=S(E):GOSUB1190:S
(E)=S1
3170 Z=D(E)*(55+INT(RND(250)*250+1))
3180 IFZ>0THENK(E)=K(E)+Z:PRINT"<CTL-5>
YOUR MILL EARNED"Z"GOLD PIECES."
3190 Z=P(E)*3:PRINT"<COM-5> YOUR ARMY CO
ST"Z"GOLD PIECES.":K(E)=K(E)-Z
```

```
3200 IF(L(E)/1000)>P(E)THEN4930
3210 IF(L(E)/500)<P(E)THEN3250
3220 FORA=1TOF:IFA=E THEN3240
3230 IFP(A)>(P(E)*2.4)THEN4930
3240 NEXT
3250 PRINT:PRINT:GOSUB160:RETURN
3260 POKE 53281,5:POKE53280,4
3270 PRINT"<CLR><CTL-1>"
3280 L2=(L(E)/1000)
3290 IFL2<10THENX=30:Y=20:GOTO3360
3300 IFL2<30THENX=25:Y=17:GOTO3360
3310 IFL2<50THENX=20:Y=14:GOTO3360
3320 IFL2<70THENX=15:Y=11:GOTO3360
3330 IFL2<90THENX=10:Y=08:GOTO3360
3340 IFL2<110THENX=5:Y=5:GOTO3360
3350 X=1:Y=2:GOTO3360
3360 A$="<CTL-1>":FORA=1TOY:A$=A$+"<CURD
N>":NEXTA
3370 FORA=1TOX:A$=A$+"<CURRT>":NEXTA
3380 FORA=XTO39:A$=A$+"<SHF-V>":NEXT
3390 PRINT"<HOME>"A$
3400 A$="<CTL-1>":FORA=1TOY-1:A$=A$+"<CU
RDN>":NEXT
3410 FORA=1TOX:A$=A$+"<CURRT>":NEXT
3420 B$=A$+"<CURLF>":C$=A$:D$=A$:E$=A$:F
$=A$:G$=A$
3430 FORA=Y-1TO23:A$=A$+"<CURDN><CURLF><
SHF-V>":NEXT
3440 PRINT"<HOME>"A$;
3450 C1$="<COM-2><CURUP><COM-B><COM-B><C
OM-V><CURDN><CURLF><CURLF><CURLF><CTL-9>
<COM-K> <CURDN><CURLF><CURLF><COM-K> <CT
L-0>"
3460 C2$="<COM-2><CURUP><COM-B><COM-B><C
OM-V><COM-C><CTL-9><COM-B><COM-B><CURDN>
<CURLF><CURLF><CURLF><CURLF><CURLF><CURL
F><COM-K> <CTL-0><COM-£><COM-£><CTL-9> <
CTL-0><COM-K><CURDN><CURLF><CURLF><CURLF
><CURLF><CURLF><CURLF><CTL-9><COM-V>
<COM-C><CTL-0>"
3470 IF(P(E)-5)<(L(E)/1000)THEN3510
3480 IF(P(E)/2)<(L(E)/1000)THEN3500
3490 PRINT"<HOME>"B$C2$:GOTO3510
3500 PRINT"<HOME>"B$C1$
3510 FORA=Y-2TOY+INT((23-Y)*.8)
3520 C$=C$+"<CURDN>":NEXT
3530 FORA=X+1TOX+INT((39-X)*.1)
3540 C$=C$+"<CURRT>":NEXT:C$=C$+"<CURLF>
"
3550 Z=C(E)+1:IFZ>7THENZ=7
3560 IFZ=0THEN3640
3570 C$(2)="<CTL-7><COM-C><CTL-9><COM-I>
<CTL-0><COM-V><CURUP><CURLF><CURLF><CURL
F><CTL-9><CTL-5><SHF-£> <COM-*>"
3580 C$(3)=C$(2)+"<CURUP><CURLF><CURLF><
CTL-0><CTL-2><SHF-+>":C$(1)="  "
3590 C$(4)=C$(3)+"<CURDN><CURDN><CTL-7><
CTL-9><COM-I><COM-I><COM-I><COM-I><CTL-0
><COM-V>"
3600 C$(5)=C$(4)+"<CURUP><CURLF><CURLF><
CURLF><CTL-9><CTL-5><SHF-£> <COM-*>"
3610 C$(6)=C$(5)+"<CURUP><CURLF><CURLF><
CTL-0><CTL-2><SHF-+>"
3620 C$(7)=C$(6)+"<CURLF><CURLF><CURLF><
CTL-1><CTL-0><SHF-+><CURDN><CURLF><CTL-9
><CTL-8> "
3630 C1$=C$(Z):PRINT"<HOME>"C$C1$;
3640 FORA=Y-2TOY+INT((23-Y)*.8)
3650 D$=D$+"<CURDN>":NEXT
3660 FORA=XTOX+INT((39-X)*.5)
```

```
3670 D$=D$+"<CURRT>":NEXT
3680 IFB(E)=0THEN3750
3690 IFL(E)<10000THEND$=D$+"<CURLF><CURL
F>"
3700 C$(1)="<CTL-9><COM-2>::<CURUP><CURL
F><CURLF><CTL-5><SHF-£><COM-*>"
3710 C$(2)=C$(1)+"<CTL-0><CTL-8><SHF-U><
SHF-I><CURDN><CURLF><CURLF><COM-2><CTL-9
><COM-A><COM-S>"
3720 C$(3)=C$(2)+"<COM-2><CTL-9>::<CURUP
><CURLF><CURLF><CTL-5><SHF-£><COM-*>"
3730 A=B(E):IFA>3THENA=3
3740 PRINT"<HOME>"D$C$(A)
3750 FORA=YTOY+INT((25-Y)*.3)
3760 E$=E$+"<CURDN>":NEXT
3770 FORA=XTOX+INT((39-X)*.1)
3780 E$=E$+"<CURRT>":NEXT
3790 FG=A(E):IFFG>40-ATHENFG=40-A
3800 IFA(E)=0THEN3820
3810 FORA=1TOFG:E$=E$+"<CTL-0><CTL-7><SH
F-X>":NEXT
3820 PRINT"<HOME>"E$;
3830 FORA=YTOY+INT((25-Y)*.49)
3840 F$=F$+"<CURDN>":NEXT
3850 FORA=XTOX+INT((39-X)*.14)
3860 F$=F$+"<CURRT>":NEXT
3870 FG=D(E):IFFG>40-ATHENFG=40-A
3880 IFFG=0THEN3900
3890 FORA=1TOFG:F$=F$+"<CTL-8><CTL-0><SH
F-A>":NEXT
3900 PRINT"<HOME>"F$;
3910 Z=S(E)-D(E)*100
3920 G$=G$+"<CURDN>"
3930 IFZ<0THENZ=.1
3940 Z=Z*5/L(E)*10
3950 IFZ>10THENZ=10
3960 Z=10-Z
3970 Z=INT((24-Y)*(Z/10))+1
3980 FORA=1TOZ
3990 G$=G$+"<CURDN>":NEXT
4000 FORA=1TO(39-X)
4010 G$=G$+"<CURRT>":NEXT
4020 PRINT"<HOME>"G$"<CTL-7><CTL-9><SHF-
Z>";
4030 PRINT"<HOME><CURDN><CURRT><CURRT><C
URRT><CURRT><CURRT><CURRT><CURRT><CURRT>
<CURRT><CURRT><CURRT><CURRT><CURRT><CURR
T><CURRT><CTL-2>YEAR:"Y(0)
4040 IFEG=1THENPRINT"<HOME><CURDN><CURDN
><CURDN><CURDN><CURRT><CURRT><CURRT><CUR
RT><CURRT><CTL-8>"T$(E)" "N$(E)
4050 PRINT"<HOME>";:GOSUB160:RETURN
4060 PRINT"<CLR>";:POKE53280,9:POKE53281
,8
4070 PRINT"<CTL-1>"T$(E)" "N$(E)
4080 GOSUB140
4090 PRINT"<CTL-7>STATE INVESTMENTS:"
4100 PRINT"<CTL-1>[1] BAZZAR
 1000 GOLD PIECES"
4110 PRINT"[2] MILL               2000 G
OLD PIECES"
4120 PRINT"[3] PALACE (PARTIAL)   3000 G
OLD PIECES"
4130 PRINT"[4] CHURCH (PARTIAL)   5000 G
OLD PIECES"
4140 PRINT"[5] EQUIP ARMY PLATOON  500 G
OLD PIECES"
4150 PRINT:PRINT"<CTL-8>YOU HAVE"INT(K(E
))"GOLD PIECES.":PRINT
```

```
4160 PRINT"<CTL-2>0=CONTINUE, 6=COMPARE
STANDINGS"
4170 INPUT"YOUR CHOICE";I
4180 IFI<1THENRETURN
4190 IFI>5THENGOSUB890:GOTO4060
4200 ONIGOTO4250,4210,4300,4350,4400
4210 D(E)=D(E)+1
4220 K(E)=K(E)-2000
4230 U(E)=U(E)+.25
4240 GOTO4060
4250 A(E)=A(E)+1
4260 M(E)=M(E)+5
4270 K(E)=K(E)-1000
4280 U(E)=U(E)+.1
4290 GOTO4060
4300 B(E)=B(E)+1
4310 N(E)=N(E)+INT(RND(2)*2)+1
4320 K(E)=K(E)-3000
4330 U(E)=U(E)+.5
4340 GOTO4060
4350 C(E)=C(E)+1
4360 Q(E)=Q(E)+INT(RND(6)*6)+1
4370 K(E)=K(E)-5000
4380 U(E)=U(E)+1
4390 GOTO4060
4400 P(E)=P(E)+20
4410 S(E)=S(E)-20
4420 K(E)=K(E)-500
4430 GOTO 4060
4440 PRINT
4450 PRINT
4460 PRINT
4470 Z=0
4480 A=A(E):GOSUB4720
4490 A=B(E):GOSUB4720
4500 A=C(E):GOSUB4720
4510 A=D(E):GOSUB4720
4520 A=K(E)/5000:GOSUB4720
4530 A=L(E)/6000:GOSUB4720
4540 A=M(E)/50:GOSUB4720
4550 A=N(E)/5:GOSUB4720
4560 A=P(E)/50:GOSUB4720
4570 A=Q(E)/10:GOSUB4720
4580 A=S(E)/2000:GOSUB4720
4590 A=U(E)/5:GOSUB4720
4600 A=Z/U(0)-J(E)
4610 A=INT(A)
4620 IFA>8THENA=8
4630 IF(Y(0)+2)=O(E)THENT(E)=T(E)+1
4640 IFT(E)>=ATHEN4710
4650 T(E)=A
4660 RESTORE
4670 FORB=1TO(T(E)+V(E))
4680 READT$(E)
4690 NEXT
4700 IFT(E)=8THEN4760
4710 RETURN
4720 IFA>10THENA=10
4730 A=INT(A)
4740 Z=Z+A
4750 RETURN
4760 PRINT"<CLR>";:POKE53280,6:POKE53281
,14
4770 EG=1
4780 GOSUB3280
4790 GOSUB890
4800 GOTO5410
4810 POKE53280,2:POKE53281,2:PRINT"<CLR>
<CTL-4>";
4820 PRINTT$(E)" "N$(E)" IS BANKRUPT"
```

```
4830 PRINT
4840 PRINT"<CTL-2>CREDITORS HAVE SEIZED
MOST"
4850 PRINT"OF YOUR ASSETS.":PRINT
4860 PRINT"MANY SERFS FLEE COUNTRY.":PRI
NT
4870 GOSUB160
4880 A(E)=0:B(E)=0:C(E)=0:D(E)=0
4890 L(E)=6000:U(E)=1:K(E)=100
4900 S(E)=INT(S(E)*RND(1))
4910 IFR(E)>5000 THENR(E)=5000
4920 RETURN
4930 Z=0:FORA=1TOF
4940 IFA=ETHEN4980
4950 IFP(A)<P(E)THEN4980
4960 IFP(A)<(1.2*(L(A)/1000))THEN4980
4970 IFP(A)>P(Z)THENZ=A
4980 NEXT
4990 IFZ<>0THEN5030
5000 T$(0)="BARON":N$(0)="FREDRICK OF MO
NALUA"
5010 A1=1001+INT(RND(8)*9000)
5020 GOTO5040
5030 A1=P(Z)*1000-L(Z)/3
5040 IFA1>(L(E)-5000)THENA1=(L(E)-5000)/
2
5050 PRINT"<CTL-3><SHF-R><SHF-R><SHF-R><
SHF-R><SHF-R><SHF-R><SHF-R><SHF-R><SHF-R
><SHF-R><SHF-R><SHF-R><SHF-R><SHF-R><SHF
-R><SHF-R><SHF-R><SHF-R><SHF-R><SHF-R><S
HF-R><SHF-R><SHF-R><SHF-R><SHF-R><SHF-R>
<SHF-R><SHF-R><SHF-R><SHF-R><SHF-R><SHF-
R><SHF-R><SHF-R><SHF-R><SHF-R><SHF-R><SH
F-R><SHF-R><SHF-R>";
5060 PRINT"<CTL-1>"T$(Z)" "N$(Z)" ATTACK
S AND"
5070 PRINT"TAKES"A1"ACRES OF LAND!"
5080 L(Z)=L(Z)+A1:L(E)=L(E)-A1
5090 Z=INT(RND(4)*40)+1:IFZ>(P(E)-15)THE
NZ=P(E)-15
5100 PRINTT$(E)" "N$(E)
5110 PRINT"LOSES"Z"SOLDIERS."
5120 PRINT"<CTL-1><COM-Y><COM-Y><COM-Y><
COM-Y><COM-Y><COM-Y><COM-Y><COM-Y><COM-Y
><COM-Y><COM-Y><COM-Y><COM-Y><COM-Y><COM
-Y><COM-Y><COM-Y><COM-Y><COM-Y><COM-Y><C
OM-Y><COM-Y><COM-Y><COM-Y><COM-Y><COM-Y>
<COM-Y><COM-Y><COM-Y><COM-Y><COM-Y><COM-
Y><COM-Y><COM-Y><COM-Y><COM-Y><COM-Y><CO
M-Y><COM-Y><COM-Y>"
5130 P(E)=P(E)-Z
5140 PRINT
5150 GOSUB160
5160 RETURN
5170 PRINT"<CLR><CTL-1><CTL-9>
   LORDS & LADYS           "
5180 POKE53280,11:POKE53281,15
5190 PRINT"  YOU ARE THE RULER OF A 13TH
 CENTURY"
5200 PRINT"DYNASTY.  IF YOU RULE WELL, Y
OU WILL"
5210 PRINT"RECIEVE HIGHER TITLES.  THE F
IRST PLAYER";
5220 PRINT"TO BECOME KING WINS THE GAME.
  BUT"
5230 PRINT"ALAS, LIFE WAS SHORT THEN, AN
D IT'S"
5240 PRINT"POSSIBLE THAT YOU MAY NOT LIV
E LONG"
5250 PRINT"ENOUGH TO WIN."
```



```
5260 PRINT"  THE COMPUTER WILL DRAW A MA
P OF YOUR"
5270 PRINT"STATE. THE SIZE OF THE CASTLE
 IN THE"
5280 PRINT"CORNER SHOWS THE ADEQUACY OF
YOUR ARMY."
5290 PRINT"IF IT SHRINKS, EQUIP MORE SOL
DIERS!  IF"
5300 PRINT"THE MARKER IS TOUCHING THE TO
P WALL, ALL";
5310 PRINT"YOUR LAND IS IN PRODUCTION.
OTHERWISE"
5320 PRINT"YOU NEED MORE SERFS TO MAN TH
E FARMS.
5330 PRINT"  HIGHER TAXES RAISE MONEY, B
UT SLOW"
5340 PRINT"DOWN ECONOMIC GROWTH."
5350 PRINT"  YOU CAN GO INTO DEBT, BUT B
E CAFEFUL"
5360 PRINT"AS INTEREST RATES ARE 50% PER
 YEAR AND"
5370 PRINT"YOU MIGHT GO BANKRUPT."
5380 PRINT
5390 GOSUB160
5400 RETURN
5410 PRINT:PRINTTAB(15)"<CTL-8>GAME IS O
VER":PRINT
5420 GOSUB160:RUN
```

Painless Computing

by T. Lee Kidwell

After purchasing a home computer and beginning to learn its intricacies, the onslaught of new information can be perplexing. New words: RAM, ROM, CPU, operating system. New ways of thinking about numbers: binary, hexadecimal, octal.

This series will show how the computer "thinks" and how to control its "thoughts." By understanding the computer's inner workings, we humans can put this electronic brain to work to serve our purposes.

HI-RES will examine the uses for the myriad of user-friendly, flexible software such as data bases and spread sheets, and explore a program that writes programs.

At the forefront of this series is a look at Beginner's All-purpose Symbolic Instruction Code — BASIC.

Then there will be side trips to explore such subjects as how to use a disk drive (a matter of considerable interest to Commodore 1541 owners), how to interface a printer to your computer, using a modem, and any other excursions readers of HI-RES may request.

A Computer Is a Tool

An understanding of the uses for a computer and how to control its actions must be taken in the proper perspective — it is a tool!

A computer is not a magic box that, with one night's work, will organize one's life into an Utopian existence. Nor will a short course at a local college be sufficient to provide the knowledge to write complicated programs.

A computer is simply a tool that will aid in reaching a predetermined end.

Just a Bunch of Switches

To better understand how a computer thinks, one must gain knowledge of the computer's brain and its internal workings.

A computer is merely a series of switches in groups of eight called "bytes." These bytes are arranged in a very logical sequence, each serving some specific purpose. The off/on sequence controls the computer's performance.

The most important group of switches is a series of bytes (a chip) called the central processing unit (CPU). This is the control center of the computer. Any action which takes place in the computer is coordinated by the CPU.

To illustrate, think of the CPU as the control tower of a busy airport. No planes (data) can take off or land (be processed by the computer and put into the bytes) without clearance from CPU tower.

When a pilot requests landing instructions (the computer user presses a key), CPU tower communicates with the pilot and designates a specific runway and landing pattern (a unique pattern of switches is turned on and placed in the proper area of the computer.) CPU tower follows up on these runway assignments to ensure all air traffic is sent to the proper hangar. In the computer, the CPU interprets the pattern of the switches set by the depressed keys and effects the appropriate action (changing of switches in the proper area).

Two Types of Memory

Many of the switches in the computer are commonly known as "memory," of which there are two types — Random Access Memory(RAM) and Read Only Memory(ROM).

RAM is comprised of switches that are off when the computer is first turned on (initialized). Switches in RAM are turned on by the CPU as necessary when information, such as a BASIC program, is input. As long as power remains on, the unique pattern of off/on switches in RAM will be maintained or remembered. When the power is turned off, all switches in RAM are turned off and any information stored in RAM is sent to never-never land.

For this reason, RAM is known as "volatile" memory. Thus, to store a program or other data, a storage device such as a disk drive or tape drive is needed. When a program is SAVEd, the information in RAM is converted to digital signals and stored on a device much the same as music is recorded on a record or tape. When the information is later LOADed, these signals convert RAM back to the same on/off state, recreating your program or data.

ROM is different from RAM in that it maintains its pattern of off/on switches.

Continued on Page 50

SANYO MONITOR SALE!!

9" Data Monitor

- 80 Columns × 24 lines
- Green text display
- Easyto read - no eye strain
- Up front brightness control
- High resolution graphics
- Quick start - no preheating
- Regulated power supply
- Attractive metal cabinet
- UL and FCC approved

- *15 Day Free Trial - 90 Day Immediate Replacement Warranty*

9" Screen - Green Text Display	*$ 69.00
12" Screen - Green Text Display (anti-reflective screen)	*$ 99.00
12" Screen - Amber Text Display (anti-reflective screen)	*$ 99.00
14" Screen - Color Monitor (national brand)	*$239.00

*PLUS $9.95 for Connecting Cable.

Display Monitors From Sanyo

With the need for computing power growing every day, Sanyo has stepped in to meet the demand with a whole new line of low cost, high quality data monitors. Designed for commercial and personal computer use. All models come with an array of features, including upfront brightness and contrast controls. The capacity 5 × 7 dot characters as the input is 24 lines of characters with up to 80 characters per line.

Equally important, all are built with Sanyo's commitment to technological excellence. In the world of Audio/Video, Sanyo is synonymous with reliability and performance. And Sanyo quality is reflected in our reputation. Unlike some suppliers, Sanyo designs, manufactures and tests virtually all the parts that go into our products, from cameras to stereos. That's an assurance not everybody can give you!

SANYO

Official Video Products of the Los Angeles 1984 Olympics

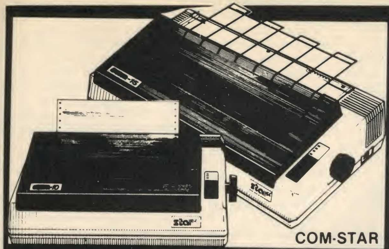
• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor
Friction
Printer

only **$199****

•15 Day Free Trial -180 Day Immediate Replacement Warranty

- Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.
- Fast 80-120-160 Characters Per Second
- 40, 46, 66, 80, 96, 132 Characters Per Line Spacing
- Word Processing
- Print Labels, Letters, Graphs and Tables
- List Your Programs
- Print Out Data from Modem Services
- "The Most Important Accessory for Your Computer"

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$199.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$489.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$579.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64—$49.00
For All Apple Computers—$79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

From Page 47

ROM switches are preset to come on or remain off when power is applied. ROM cannot be changed. Therefore, the data stored in this type of memory always will remain unchanged. ROM generally is used to permanently store some very important programs in the computer.

Operating System

Switches do not a working computer make! The computer is just a complicated piece of electrical hardware. To perform a meaningful function, this hardware must be programmed by a mere human. In a special part of ROM is just such a program — the "operating system."

This system is a series of bytes, set to logical patterns, which form a program that controls the operation of the computer. This interprets user input and works in cooperation with the CPU to prompt the computer to perform the proper tasks, such as PRINTing a word on the screen.

A program of this type, which is permanently stored in ROM, is called "firmware," as opposed to "software." Both are programs, but software is normally SAVEd on magnetic storage media (tape or disk) and is INPUT into the computer's RAM as needed.

To understand how RAM, ROM, CPU and the operating system interact, let's investigate how the computer reacts when the user turns it on and types: PRINT "Hello" *RETURN*.

When power is turned on, the CPU immediately goes to work. The CPU starts by sending control of the action to the operating system. The operating system then performs the initialization procedures, which include such steps as setting the switches in the correct bytes in RAM to determine screen, border, and character colors.

Once all the necessary housekeeping procedures are complete, the operating system routine finishes by printing a prompt on the screen — usually "READY." This is an indication to the user that the computer is awaiting instructions.

Next, the user types in the command: PRINT "Hello." As this statement is typed, each key is evaluated by the CPU. If the depressed key is a special command key, such as the key to clear the screen, the CPU will send control of the computer to the part of the opeerating system which contains the "subroutine" (a program within a program) to clear the screen. If the key is a character, such as those in the line being evaluated, the computer will store the keys in RAM as a series of bytes. When the RETURN key is depressed, the CPU evaluates the group of bytes in RAM that have been changed since the last RETURN to determine the keys and take the appropriate action — in this case, PRINTing the word "Hello" on the video screen.

Yes or No

Before going to the first excursion into BASIC programming, there is one theoretical area still to explore: computer logic. This could be an exceptionally long trip if sidetracked along all of the side streets such as Binary Boulevard and Boolean Logic Drive. However, for sake of sanity and brevity, take a stroll down Easy to Understand Street.

BASICally speaking, this "bundle of switches" can do one thing: compare one series of switches to another. The programmer must tell it what to do based on whether all switches are set to the same or different patterns, in other words, whether the comparison is true or false.

That's it, it is simply yes or no. It is amazing how simple the logic is in programming the computer. The excursions into BASIC will reveal statements such as IF THEN. The logic of this statement is simply IF a situation is true, THEN perform a defined function.

There's also the ON GOTO statement. This command allows testing of multiple true/false conditions, simultaneously: ON the results of the test of several conditions, GOTO a designated subroutine and perform the programmed task.

Even statements such as PRINT are evaluated using this logic. The computer is constantly comparing the input from the keyboard to a known factor. In this case the "P" is tested first. Since "P" could be the beginning of the word PRINT, a command the computer has been programmed to respond to, it will test to see if the remainder of the word is "RINT." If yes, the PRINT routine in the operating system will be called by the CPU. If false (such as the word PAINT) another action would be taken.

When designing a program, this true/false logic must be followed.

Organizing Logic

Since the computer is a tool of logic, it is very unforgiving of disorganized logic. Before writing a program, a plan of action should be worked out. These planning steps are sometimes tedious, but will save a great deal of frustration and prove to be well worth the up-front time, taking some of the pain out of computing.

There are five steps which should be taken in the development of a program:

- Define the task to be accomplished.
- Outline the solution to the problem (algorithm).
- Diagram the solution.
- Enter the program into the computer.
- Test and debug the program.

Following these steps will force the user to determine an objective before sitting down at the computer to type a disorganized program. The logic will be organized and easy to follow — an essential element for the test and debug steps.

Workable Task

In defining the task to be accomplished, the user must consider the true/fale logic of the computer. A task requiring decisions about numerous gray areas will not be workable. The computer cannot think!

The electronic brain is very good at solving problems that involve manipulating data. For example, organizing a mailing list, tracking and recording financial transactions, and solving complicated mathematical problems would all be feasible tasks. But a task such as deciding where to spend a vacation would be difficult for the computer.

With this in mind, consider some of the tasks you would like to accomplish and the feasibility of performing the tasks on a computer.

Next issue will further explore BASIC and work toward writing a useful program.

Editor's Note: Send correspondence to Painless Computing, HI-RES Magazine, 280 W. Canton Ave., Suite 310, Winter Park, Fla. 32789.

A Lifetime of Discovery Begins With ATARI® Computers

ATARI HOME COMPUTERS

ATARI 600XL™ Home Computer (16K RAM)	CALL
ATARI 800XL™ Home Computer (64K RAM)	
ATARI 1200XL™ Home Computer (64K RAM)	
ATARI 1400XL™ Home Computer (64K RAM)	CALL
ATARI 1450XLD™ Home Computer (64K RAM)	

ATARI ADDITIONAL EQUIPMENT

ATARI 410™ Program Recorder	CALL
ATARI 830™ Acoustic Modem (ATARI 850™ required)	
ATARI 835™ Direct Connect Modem	
ATARI 850™ Interface Module	CALL
ATARI 1010™ Program Recorder	
ATARI 1020™ 40-Column Color Printer/Plotter	
ATARI 1025™ 80-Column Printer	CALL
ATARI 1027™ Letter Quality Printer	
ATARI 1030™ Direct Connect Modem	
ATARI 1050™ Disk Drive	CALL
CX85 Numerical Keypad (Includes Handler Diskette, for use with VisiCalc)	
ATARI CP/M Module	
CX77 ATARI Touch Tablet™	CALL
CX30-04 Paddle Controller (Pair)	
CX40 Joystick (Single)	
CX40-04 Joystick (Pair)	CALL
CX60 ATARI Ultimate Joystick	
CX42 ATARI Remote Control Joystick	
CX80 ATARI Trak-Ball™	CALL

ALL-IN-ONE-PAKS

KX7110 The Writing System	CALL
KX7111 The Entertainment System	
KX7114 The Programming System	CALL

ADD-A-PAKS

KX7098 The ATARI Accountant	
CX418 The Home Manager	CALL
CX488 The Communicator	
KX7102 The Arcade Champ	
KX7099 The Basic Tutor I	CALL

PRINTERS

Gemini STX-80 (80 Column Thermal)	$169.00
Gemini 10X Printer (80 Column)	299.00
Gemini 15 Printer (136 Column)	399.00
Delta 10 Printer (80 Column)	479.00
Epson RX-80 (80 Column) Dot Matrix	379.00
Epson RX-80FT (80 Column)	469.00
Epson FX-80 (80 Column) Dot Matrix	579.00
Epson FX-100 (136 Column) Dot Matrix	749.00
Prowriter 8510 AP	375.00
Gorilla/Banana Printer (80 Column)	199.00
Axiom AT-100 (80 Column) Inc. Int. Cable	229.00
Silver-Reed EXP 500 Daisy Wheel Printer (80 Column)	429.00
Silver-Reed EXP 550 Daisy Wheel Printer (136 Column)	679.00

PRINTER INTERFACE CABLES

ATARI Universal Cable	$ 59.00
MPP-1100 Parallel Printer Interface	79.00
Parallel Printer Cable	30.00

PRINTER RIBBONS

Gemini Printers (Black/Blue/Red/Purple)	$ 3.00
Epson Printers	6.00

MONITORS

Gorilla™ Hi Res 12" Non-Glare Screen	$ 89.00
Gorilla™ Hi Res 12" Non-Glare Amber Screen	99.00
Amdek Video 300 Green Screen	130.00
Amdek Video 300A Amber Screen	144.00
Amdek Color-I Screen	280.00
Sakata SC100 Color Screen	239.00
BMC Color Screen	229.00
Monitor Cable	12.00

MODEMS

MPP-1000 Modem	$129.00
Signalman Mark II Modem	79.00
Hayes Stack Smartmodem (300 BAUD)	239.00
Hayes Stack Smartmodem (1200 BAUD)	549.00
Novation J CAT	119.00
Novation 103 Smart CAT	189.00
Novation Auto CAT	199.00

DISK DRIVES

RANA 1000	CALL
TRAK AT1 (Single/Double Density)	CALL
TRAK ATD1 (Single Density/Parallel Interface)	CALL
TRAKATD2 (Single/Double Density/Parallel Interface)	CALL
ASTRA 1620	CALL

GENERIC DISKS

Generic 100% Defect-Free/Guaranteed MINI-FLOPPY DISKS

Diskettes (1 Box Min.) — 10 per box

	SS/SD	SS/DD	DD/DD
1 or 2 Boxes	14.99/box	17 49/box	2(99/box
3 - 9 Boxes	13.99/box	:5.99/box	19.99/box
10+ Boxes	13.49/box	14.99/box	18.99/box

Bulk Diskettes with Sleeves — Price per Disk

	SS/SD	SS/DD	DD/DD
10 - 29	1.39	1.59	1.99
30 - 99	1.29	1.49	1.89
100+	1.25	1.45	1.79

DISKETTE/CARTRIDGE/CASSETTE FILES

Flip 'N' File 10	$ 3.95
Flip 'N' File 15	6.95
Flip 'N' File 25	19.95
Flip 'N' File 50	26.95
Mini Flip 'N' File	19.95
Maxi Flip 'N' File	24.95
Cassette 'N' Game File	14.95
Disk Bank	4.50
Disk Case (Plastic)	1.99
Blank 30 Min. Cassettes	1.29

RAM (MEMORY) BOARDS

ATARI 16K	$ 29.00
Intec 64K Board (400)	109.00
Intec 48K Board (400)	79.00
Intec 32K Board (400)	59.00
Mosaic 32K Board	99.00
Mosaic 64K Board	149.00

80 COLUMN BOARDS

BIT 3	$249.00
Austin Franklin	239.00

KEYBOARDS

Inhome Keyboard for (400)	$ 59.00
Joytyper Keyboard for (400)	79.00
Tara Keyboard for (400)	69.00

To Order Call TOLL FREE 1-800 824-7506

MC

VISA

C.O.D.
(Add $2.50)

COMPUTER CREATIONS, Inc.
P.O. Box 292467
Dayton, Ohio 45429
For Information Call:
(513) 294-2002
(Or to order in Ohio)

All Orders Add $3.00 Shipping & Handling.
Ohio Residents Add 6% for Sales Tax.

ASK FOR OUR FREE CATALOG

HARDWARE COVERS

ATARI 400	$ 5.95
ATARI 800	5.95
ATARI 1010	5.95
ATARI 410	5.95
ATARI 810	5.95
ATARI 1200	5.95
Gemini 10 Printer	5.95

ATARI COMPUTER BOOKS

20862Q	Advd. Programming Tech. Atari	$11.95
07076Q	Atari Assembler	12.95
49449Q	Atari Basic Learning By Using	4.95
51617Q	Atari Basic Source Bk.	11.95
76689Q	Atari For Beginning Beginner	7.95
33839Q	Atari Games & Recreations	12.95
19183Q	Atari Programming W/55 Programs	12.95
44613Q	Atari Sounds & Graphics	9.95
42053Q	Best of Softside Atari	16.95
39236Q	Bk. of Atari Software-83	12.95
56640Q	Computers For Kids Atari	4.95
49158Q	Computes First Bk. Atari	10.95
51589Q	Computes First Bk. Atari Games	10.95
50773Q	Computes First Bk. Atari Graphics	10.95
56473Q	Computes Second Bk. Atari	10.95
50446Q	Computes Third Bk. Atari	10.95
44650Q	Creative Atari	12.95
49522Q	Games for the Atari	6.95
49555Q	Ht. Program Your Atari in 6502	8.95
69573Q	Ht. Win At Atari Computer Games	7.95
49134Q	Inside Atari DOS	16.95
52577Q	Kids and The Atari	16.95
66064Q	Learning W/Atari Logo	12.95
51518Q	Mapping the Atari	12.95
44426Q	Picture This	12.95
43944Q	Some Common Basic Prog. Atari	14.95
24333Q	Understanding Atari Graphics	2.95
47044Q	Visicalc Bk. Atari Ed.	12.95
43581Q	Your Atari Computer	14.95
97149Q	31 New Atari Computer Prog. Home	7.95

SOFTWARE — ALL MFGRS. — DISCOUNT PRICES

Atari
Adventure Int.
Avalon Hill
Broderbund
Datasoft

Don't Ask
Edu-Fun
Educational
Epyx
Infocom

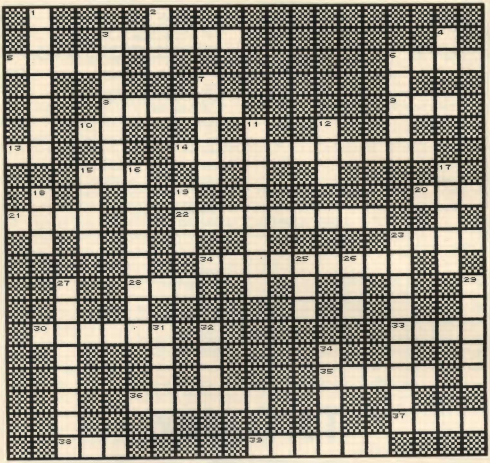
LJK
Program Design
Sierra On-Line
Spinnaker
Synapse

CALL FOR PRICES

★★★

Cryptics

by Patrick R. Serafine



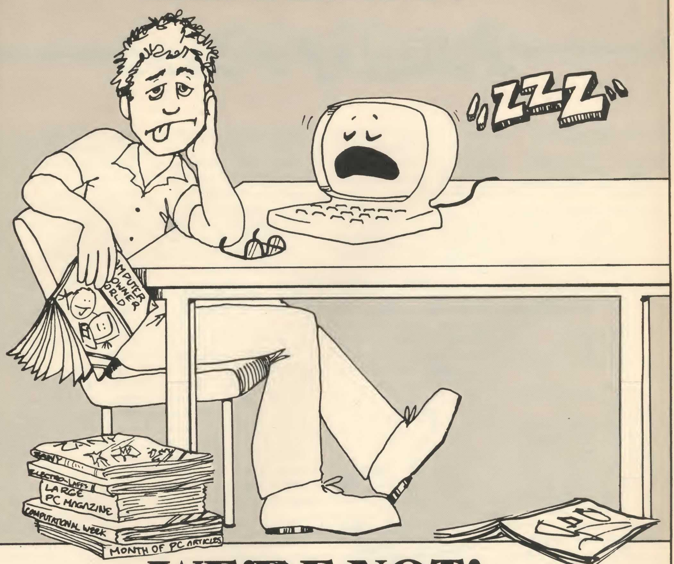
Across

3. Computer which represents all numbers as electrical signals of corresponding magnitudes
5. Programmed instruction, learning or teaching
6. Programmable read only memory
8. Eighteen per track
9. Generalized input/output command in Atari BASIC
10. Integrated circuit
13. Logical operator to test if either of two expressions is true
14. Add-on IO devices
15. Binary digit
20. Read only memory
21. Input or output channel
22. Program which turns source code into object code
23. Information used in the execution of a program
24. Module used for connection between peripherals
28. Assembler mnemonic for no operation
30. Half of a byte
33. LSI chip which gives graphics mode 9, 10, and 11
35. LSI chip which processes the video display for Atari
36. Middle console button
37. Unconditional branch in BASIC
38. Numbered or lettered button
39. Programming language named for a French scientist

Down

1. Device for making hardcopy
2. Random access memory
3. Character codes on Atari
4. Input output acronym
6. Contraction for the words "picture element"
7. LSI chip which processes the sounds and monitors the Atari keyboard
11. Mass storage medium which is random access
12. Base sixteen numbering
15. Eight bits
16. IO device with a keyboard
17. Programming language invented by Charles H. Moore
18. Disk operating system
19. Error 139 - device —
23. Exterminating gremlins
25. Used for putting comments in a program in BASIC
26. Logical operator for testing whether two expressions are both true
27. A type of controller
29. A circular construct on the surface of a diskette
31. Used to retrieve listed or untokenized BASIC source
32. Used to write tokenized BASIC source to storage medium
34. Beginner's all-purpose symbolic instruction code

LISTLESS?



WE'RE NOT!

Hi-Res can put an end to those home computer blues and keep you out of the red with reader-written programs that are educational and entertaining.

Subscribe now and save up to 43% off the newsstand price!

Each issue, Hi-Res also brings you:

- Articles written by the best technical writers in the industry.
- Informative news on hardware, programming and peripherals relating to the Atari and Commodore.
- Software reviews.
- Family-oriented features and much more!

Hi-Res Magazine 280 W Canton Ave. Suite 310 Winter Park, Fla. 32789

Family-Place

by Dorothy Heller

A new contender in the educational software field whose motto is “software designs for developing young minds” is offering a different slant on children's computer products for home and school.

“We decided from the beginning that curriculum-oriented software and “shoot-em-ups” were out,” says Kari Beims of Maximus.

“We wanted to do something new.”

Maximus' approach is to combine the concept of “software movies” for children with games that teach “lessons in daily living.” The products for the Atari and Commodore 64 computers, which include *Safetyline, Storyline, Travelline, Scienceline* and *Societyline*, deal in morals and guidelines for behavior instead of arcade action or computer-assisted instruction.

“Although children can informally learn reading, spelling and other skills with our products, we wanted to go beyond school learning,” said Beims. “Our goal is to reinforce the lessons that parents try to teach in a fun way.”

For this month's “Family Place”, we interviewed Beims on the company's philosophy and product development. We also reviewed two of their products — *Safetyline* and *Storyline* — with the assistance of an expert in the field of early childhood education who has been working with children for more than 5Ø years!

Unlike Sesame Street Software, Scholastic, Inc. and Spinnaker Software, Maximus doesn't have major corporations or an easily recognizable brandname as backing. The MacLean, Va., company and its products are the creation of David W. Mastran, Maximus president, and a small but versatile staff who wear many hats.

Typical of the Maximus staff is Beims, who designs computer graphics, works with marketing, goes to trade shows, does package design and performs several other functions. When Mastran told her to create cartoons and design computer graphics for educational games, she obligingly put on another hat and learned by doing.

“I was petrified at first,” Beims recalled. “My background is in studio art and art history and I really hadn't worked much with computers or computer graphics before. I was originally hired to do business graphics.”

Undaunted, Beims went to the library, taught herself how to program in BASIC, took a class in computer graphics, and worked closely with the programming team to translate ideas into visual images on the screen.

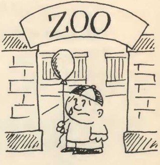
Beims was inspired by her mother's example, who returned to college after years of raising a family, earned a scholarship, and became a legislative assistant to the Maryland Commission on Women and Appointments officer to the governor of Maryland. “Seeing my mother graduate was one of the greatest moments of my life,” Beims said. “I learned from her that you can do anything and learn anything if you want to.”

Beims began her computer graphics career by drawing a cartoon of “Max the Cat.” Maximus' “software movies” developed from this character, who grew from a concept to a cartoon to a character on the computer screen. “We decided to have his lips move while he talked, then added more and more animation. By the time we were finished, we had a fully animated movie with a lip-synched narration on the cassette tape recorder. We then developed interactive games to reinforce the movie.”

Program on Safety

“Parents are universally concerned with teaching their children lessons about safety, like what to do when lost, whether to speak to strangers, how to cross the street and deal with traffic,” stated Beims. “Although the parents know how important these lessons are, often it goes in one ear and out the other when they try to teach their children. The purpose of *Safetyline* is to reinforce guidelines that will help kids to get safely through typical situations.”

Safetyline is designed to be fused either with a cassette-only version, or both disk drive and cassette. Side A features a movie called “Sam Goes to School; Side B features “Sam Gets Lost at the Zoo.” The two software movies, narrated by “Max the Cat”, include specific guidelines for crossing the street safely and what to do when lost. Each software movie is accompanied by two games that reinforce the lessons that are dramatized by the movie.



In “Streetcross”, the player must get Sam to school safely, using either a joystick or keyboard control. To win the game and earn a high score, the child has to remember the safety rules. If the player moves Sam across the street in the middle of the block, when the light is red or when a car is still moving through the intersection, he hears warning music and the program moves Sam back to a safe place. When Sam reaches the school doors safely, the program plays winning music and displays the child's score.

In "Hidden Tips", one of Max's Safety Tips appears at the bottom of the screen and the keyword in the safety tip appears at the top of the screen. The keyword is then hidden in the game field by surrounding it with random letters. The player can choose the easy or hard version.

In the easy version, the word is arranged only left-to-right or top-to-bottom. In the difficult version, the keyword may appear in any direction.

When the player finds the keyword and moves the cursor over the letters, using keyboard or joystick, the screen changes and another keyword appears hidden in the game field. The challenge is to find all of the keywords from the safety tip as quickly as possible before the clock counts down to zero.

"Sam Gets Lost at the Zoo" is accompanied by "Tipmatch" and "Zoomaze." In "Tipmatch", eight squares appear on the game field. Behind each square is one of the five safety tips from the movie. Children can play the game individually or together to match the tips. Kids can choose from an easy or difficult version.

In "Zoomaze", Sam is lost in the zoo. The goal of the game is to help Sam find his way safely back to his teacher.

Sam's cap appears in the middle of the gamefield. There are four boxes in the corners of the gamefield, hiding a popcorn stand, a policeman, Sam's teacher and a stranger.

When the child moves Sam toward one of the boxes, he or she encounters an invisible maze. The challenge is to guide Sam through the maze to the right boxes. If Sam reaches his teacher, the child hears winning music. If Sam ends up with the stranger, the game is lost. Bouncing Sam against the walls of the maze also decreases the child's score. In the easy version, the maze appears on the screen; in the more difficult version, the maze disappears.

Bedtime Stories

The *Storyline* software movies use traditional folk tales, narrated by Clover the clown, to teach lessons about behavior and attitude. In "Rumpelstiltskin", the movie reinforces morals about boasting and promising "more than you can deliver."

"The Ugly Duckling" reinforces morals about respecting others who are "different" and about "following your heart."

Two games accompany "Rumpelstiltskin: "Guess My Name" and "Promises, Promises." In "Guess My Name", the player builds a stack of gold by guessing the name of the troll, letter by letter. The troll can have one of almost 2ØØ boy's and girl's names. The child can determine the number of guesses he or she wants, from one to 12. The number of letters in the name appear as blank spaces at the top of the screen. As the child types in a correct letter, it appears in the name. Letters that don't belong in the name appear at the bottom of the screen to remind the child not to repeat wrong guesses.

"Promises, Promises" teaches not to "promise more than you can deliver." The goal of the game is to promise less than Rumplestiltskin wants. The player uses the joystick or cursor control keys to choose a number between 1Ø and 9Ø. Rumpelstiltskin then moves his arrow and stops at the number he wants. If the player has promised less than Rumplestiltskin asks, he or she keeps the points. If the player wins, Rumpelstiltskin explodes and becomes a pile of straw.

"The Ugly Duckling" is accompanied by "Pick the Twins" and "Duck Puzzle." In "Twins", the player must match images of different ducks or flowers in as few tries as possible.

In "Duck Puzzle", the goal is to unscramble the puzzle on the screen using either keyboard controls or the joystick.

Good Value

We reviewed *Safetyline* and *Storyline* with the assistance of our educational expert and several children. We concluded the packages are exceptionally good values for home and school, with several reservations that we hope Maximus will correct in future products.

We liked the software movies, the originality of Maximus' approach to educational software, and the concept of teaching "lessons in daily living." The products and the direction Maximus is moving show lots of potential for a new kind of educational home software.

The Academic A

by Lloyd R. Prentice

The Titans of science have taught us the "things" in the universe are much less interesting than the forms in which they are combined. Hold a "thing" under close scrutiny and the sense of "thingness" gives way to form or process — patterned structure in time and space of yet more elementary things. Mountain yields to rock formation; rock yields to crystalline structure; crystal yields to atomic matrix; atom yields to system of particles.

Facts are the "things" of education. Out of context, a singular fact is a bit of an embarrassment, a sort of mental fifth wheel or odd man out with little power to stir human imagination, passion or action. "The Declaration of Independence was signed in 1776." So what. Disembodied facts pile up in the litter bins of the mind, of little use to anyone except the odd trivia buff or game show contestant.

But examine the structure implicit in the fact. Trace the processes that generate it. Make connections with what's previously known. Draw the inferences. Question, probe, relate. Suddenly the fact becomes a springboard into a larger universe, a ticket to a grand tour, a tasty morsel from a gourmet meal. Understand "Independence" in the context of human oppression. Connect "Declaration" with martyrdom and courage. See "1776" against the widescreen of history. Suddenly you've discovered the stuff that swells the heart and brings tears to the eyes. This is what education is about. Or should be, anyway.

Good teachers know how to make these connections.

But suppose you focus on "structure" or "process" at the expense of fact? Suppose you melt down the concrete impressions and experiences of life to extract "pure form" or "pure process" — the logical, mathematical or aesthetic essenses of the connections between things? Suppose you divorce relationships and processes from stuff that you can see, hear, feel, taste or smell?

Now you enter a world of abstraction, theory, logic, philosophy, mathematics, music, experimental poetry and art. Take a wrong turn and you enter the world of delusion, hysteria and poppycock. If you have a mind for it, the abstract landscape can be exotic, and striking. But stray too far and reason becomes disembodied — at best, brilliantly focused like a laser beam, cutting through appearances, laying bare ever more exotic landscapes; at worst, deranged and incoherent; or, more often, sane, but hopelessly out of touch with the gritty, life-affirming surfaces of consensual reality.

Most people have little taste for it, preferring the raw experience of a sunset or the companionship of a friend.

The computer is the product of formidable abstract thinking. But as an educational medium it can take us in either direction — toward palpable fact or ineffable abstraction. At one extreme the computer is a channel of communication from the instructional designer to the learner, conveying a highly structured message of fact piled upon fact — an interactive audio-visual medium. In this mode, often called the tutorial approach, the learner is largely receptive, passive. Learner involvement is limited to signaling comprehension in terms defined by the program's creator.

At the other extreme the computer is a medium of expression or discovery, putting the initiative in the hands of the learner — a tabula rasa, pregnant with potential for self-directed insight. A programming language provides the crudest example of such a learning system. With sufficient fluency the learner can explore content in many domains. The logic of the language itself becomes a tool with which to unveil mysteries. The youngster provides the facts and the computer stands in for the teacher in helping to make connections. But look carefully, the logic of the language and the inadequacies of the computer impose their own constraints on thought. The syntax of the command set structures expression in much the way the interstate highway system structures travel across the continent.

These two ways of using the computer, then, are something like the difference between a museum and a paint-by-number set — the difference between being exposed to art and doing it.

The notion of using the computer as a medium of discovery, or as a "tool to think with," is wonderfully seductive, but difficult in practice. The best demonstration, Seymour Papert's *Logo*, was more than a decade in the making. Many designers are exploring possible formats. A small hint of some of the difficulties can be seen in Spinnaker Software's *KinderComp*, a "collection of learning exercises" for kids three to eight.

KinderComp includes six exercises — Draw, Scribble, Names, Sequence, Letters, and Match. Three of the exercises are biased in the direction of discovery learning. The other three are conventional drill-and-practice activities dealing with number sequences, capital and lowercase letters, and pattern recognition. The educational objective of the discovery learning exercises, it seems, is to encourage youngsters to use the computer and to become comfortable with the keyboard. But it's far from rigorous. Says author Doug Davis, *KinderComp* was written for our daughter Amy. I wrote it because I wanted her to have fun with the computer."

The conventional drill-and-practice exercises are Sequence, Letters and Match.

In Sequence, five numbers are presented in numerical order. The learner has to fill in the next number in the sequence.

In Letters, a lowercase letter appears on the screen. The learner is rewarded for pressing the uppercase letter on the keyboard that matches the letter on the screen.

Match involves matching one of three patterns with a fourth pattern.

Each of these three exercises is simple and clear. Responses for right and wrong are appropriate. Graphics are simple, adequate, but not outstanding.

The three discovery learning games are Draw, Scribble and Names.

In Names, the youngster types in a name of up to 15 characters. The computer then plays with the name on the screen, creating a colorful pattern and movement. Unfortunately, the pattern is always the same, so Names quickly becomes tedious.

In Scribble, the user types a character. The computer then repeats the character for a full line. Typing a new character at the beginning of each successive line enables the user to create patterns on the screen. If the youngster is using an Atari computer, he can change the color of the characters by typing the Atari logo key or the CAPS/LOWER key.

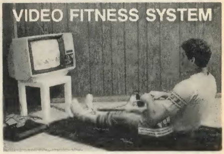
Draw is a simple Etch-a-Sketch program with a fill function. It enables the user to draw pictures on the screen using a joystick. The user can change the width and color of the brush and the color of the background. Pictures cannot be saved. Says Davis, "Draw . . . is our pride and joy."

The best test of a program like *KinderComp* is to put it into the hands of the intended user. Since I did not have a convenient tot at hand, I have to pass on the question of whether or not the program is indeed fun. I do not doubt the first time through it meets the goal, but I have some question about repeated play value. Draw is the most open-eneded exercise, but I feel it has two drawbacks: 1) the joystick offers the user very poor control of the drawing process. Even with adult motor control, I could not always place the lines where I wanted them. Moreover, since I could not find an easy way to turn off the trace, or brush, I could not make drawings that involved discontinuous lines, for example, the eyes in a face. 2) Even if I were able to create a successful drawing, I could not save it to show later to grandma or my friend next door.

The big question with Draw, Scribble and Names, however, is what do they teach? What connections do they inspire? Some things are obvious. Names and Scribble encourage keyboard exploration and letter recognition skills. Scribble encourages left to right scanning and Draw has the potential for developing motor control and right brain functions.

The value of *KinderComp* may well rest in the fact that it contains six different exercises and in this way gives the youngster the feeling of "a lot to do." The balance between structure and free exploration also is a plus. Moreover, it may indeed help encourage a youngster's innate interest in the computer and help them develop keyboard skills. But the big problem is the exercises greatly limit the scope of the child's explorations. They quickly put the mind in a playpen rather than encouraging free exploration of new horizons. They promise much, but do not unleash the potential of the medium. For connections to matter, they must add up to something.

KinderComp is available for Atari computers.



Commodore On-Line

by Stephen Reed

Many Commodore 64 owners want more from their home computers than the manufacturer can provide. Innovations, adaptations, simplifying — the list is endless.

The purpose of this new department is to serve as a source to users who want technical information about their C64. Hardware and software will be covered in a wide variety of applications and interests.

However, hardware projects presented in this department on occasion will involve opening the C64 cabinet. Please note: this will VOID Commodore's warranty.

Power LED cable Mark Losh

Anyone who works often with assembly language programming understands the frustrations of putting the machine into some endless loop where the only way to escape is by turning the computer completely off.

Of course, this action forces the user to reload all of the object files before giving another try.

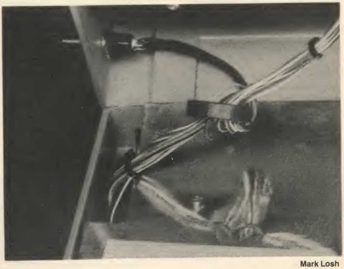
This issue describes how to install a hardware reset switch to the C64 that will reset the system NO MATTER WHAT THE SYSTEM IS DOING. The benefit of such a switch becomes clear for debugging as using the reset will not destroy memory.

Don't Forget the Screwdriver

Necessary for this procedure are the following tools: a low wattage soldering iron, some fine gauge resin core solder, a Phillips screwdriver and wire stripper/cutters. Also needed is a momentary contact pushbutton switch and a two-conductor molex-type connector pair. These can be purchased at most electronics stores for less than $5.

Opening the C64

First, disconnect the C64 from the rest of the system and turn it upside-down on a soft surface to prevent scratching the keycaps. Remove the three Phillips screws from the bottom of the case, put the screws aside and turn the C64 upright.



Mark Losh

Keyboard cable connection (entwined wiring)

Now, open the cabinet from the bottom — there are hinge-like formations on the front of the cabinet to make opening exceptionally easy.

Locate and unplug the power LED cable — this connection will be in the lower-righthand corner of the case. Remember how the connector is positioned for later reassembly.

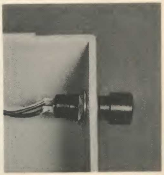
At this stage, unfasten the keyboard cable connection in the upper-lefthand corner. Remove the upperhalf (keyboard) of the cabinet.

Switch Installation

Drill a mounting hole in the keyboard-half of the cabinet for installing the reset switch. The hole size should be determined by the size of switch.

Locate the switch on the lefthand side of the cabinet about an inch down from the top and an inch from the back. Drill a hole and insert the switch for a test fit. Be sure everything is correct before going any further.

Cut a piece of two-conductor wire about five



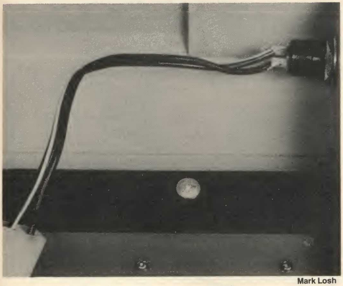
Mark Losh

Location of reset switch

inches long. Strip the ends of both wires, solder them at one end to the molex connector, crimp and insert the molex pins into the plastic housing.

The next step is to pass the wire through the hole in the cabinet so the reset switch can be soldered. Don't worry about polarity as this modification is simply a short and it doesn't matter which wire goes where.

After soldering, install the switch in the cabinet.

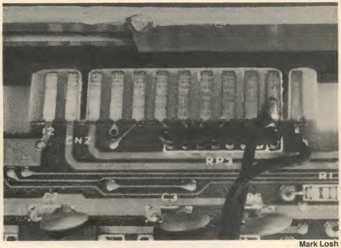


Mark Losh

Reset switch to molex connector assembly

Now, place the keyboard assembly aside and get the computer housing. Place it in front as if it were going to be used. On the lefthand side of the back is the user port (see page 355 of the Programmers Reference Guide). Counting from the right, we will be shorting pins (lines, connections, whatever) one and two together to force a system reset.

Cut another piece of two-conductor wire and connect one end to the mate of molex connection previously used with the switch. The other ends of the wire are soldered to the back of the edge connections on pins one and three.



Mark Losh

Shorting wire soldered to user port

Final Test

To test the work, connect the computer to the video and power supply without the keyboard and turn the system on. After the screen has come up, short the wires together. The screen will flicker and the system will reset in about a second. If this does not happen, recheck the work.

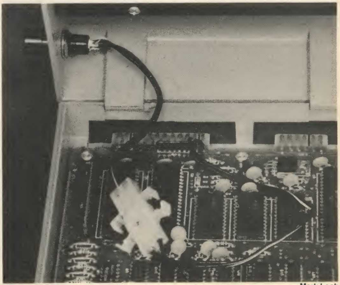
Finally, get the keyboard assembly, connect the reset connector, keyboard connector and power LED connector. Put the cabinet together and replace the screws in the back of the housing.

This procedure now offers a master override to force a system reset at any time.

Next issue: Hate blue letters on blue? On-Line will try black on white — just like reading a book.

★

Editor's note: HI-RES invites readers' contributions and questions. Please address correspondence to Commodore On-Line, HI-RES Magazine, 280 W. Canton Ave., Suite 310, Winter Park, Fla., 32789.



Reset switch via molex to user port connection

Mark Losh

Strolling Forth

by Stephen Maguire and Evan Rosen

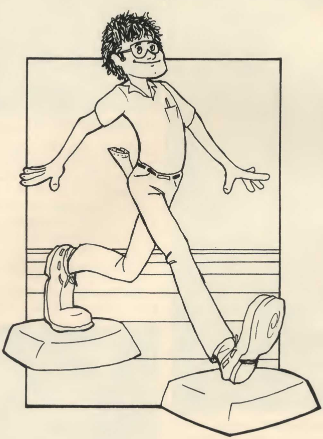
In the previous three articles, we have discussed what FORTH is and have given some examples of FORTH code. But the novice still has many questions stemming from FORTH's esoteric way of doing things. For example, why doesn't FORTH run under a Disk Operating System? This installment will answer this and many other questions still remaining about FORTH.

In fact, if the programmer understands why FORTH seems quirky, a better appreciation of the language is gained.

What is "Fig" FORTH?

All of the popular FORTHs for the Atari are implementations of the public domain software provided by the FORTH Interest Group (fig). About six years ago, this group implemented a usable FORTH system on many different computers. This was made available to anyone for reproduction costs and put no restrictions on its use. The effort to increase the popularity of FORTH resulted in literally hundreds of implementations of fig-FORTH. This dialect by far has become the most popular.

And just for the record, a great deal of credit should be given to Bill Ragsdale for making his 6502 fig-FORTH available to the public. Without his implementation, none of the FORTHs for the Atari would exist. He too was a main driving force in making fig-FORTH available for as many processors as possible.



Ironically, fig-FORTH (which conforms to the 77-standard) has become so widely accepted all attempts to bring it up to the 79-standard have failed. Incidentally, the 83-standard has just been accepted and a new public domain FORTH soon will be available that conforms to this standard.

To DOS or Not to DOS

This is probably the question most frequently asked by new FORTH programmers. There are several reasons for not using a DOS. The best answer is the vast majority of programs do not need the capabilities provided by DOS and would waste 10K-plus by having it in memory. And for those programs which do need to save and retrieve data from disk, very few require the file manager DOS affords. Many can simply perform basic read/writes to the disk.

A second reason for going DOSless is transportability of data. For example, FORTH source code can be written on a TRS-80 — which unlike the Atari uses a standard disk format. And an IBM PC running FORTH can read the disk as though it was developed on that machine.

However, disks written using TRS-80 DOS cannot be read by the IBM DOS because the file structure is vastly different. Thus, because of software incompatability in the DOS, file transfers between the two systems are practically impossible. This problem does not exist in FORTH.

Finally, it is very easy to implement FORTH on any new processor. Minimal disk routines need to be written to have FORTH up and going, whereas a good DOS could take more than a year to write.

Then Why Use Screens?

FORTH source code is not stored in files, but on screens. By using the minimal disk routines available in FORTH, source code is entered directly onto disk sectors. A screen is generally 512 or 1024 bytes (four or eight sectors) in length. A special editor is used to write source code to these screens. To compile a FORTH program, it is LOADed. LOAD takes a screen number on the stack and source code on that screen is compiled.

Because source code is stored directly on disk and not in files, a great deal of available disk space is wasted. For example, if only the first two lines of a screen contain code, the remainder of the screen is wasted disk spaces. Usage shows that nearly half the disk is effectively wasted. It is for this reason FORTH programmers traditionally pack as much code on a screen as possible. Basically, program readability is given up for increased disk storage. Hence, FORTH has become known as a write-only language.

With the cost of disks much lower today than ever before, it is far better to code for readability than to minimize disk usage. This means indenting DO loops — IF...ENDIF constructs and the like.

Using Reverse Polish Notation

Without going into detail, Reverse Polish Notation is a method of expressing arithmetic formulas without the need for parentheses. Hewlett Packard has used this form of notation in its calculators for more than a decade. FORTH does not use RPN for any particular reason except it fits nicely into the whole FORTH picture.

The FORTH language is composed entirely of subroutines which all behave in the same manner. These subroutines — words — take a number of arguments, operate on them and return a number of arguments.

To simplify, FORTH sets aside a special stack called the parameter stack which holds arguments until needed. FORTH words will remove arguments from this stack and possibly replace with others. The arithmetic words — *, /, etc. — do exactly this. Each removes two arguments and puts one back.

Thus, to write the expression (3 + 5)/2 in FORTH, we have:

```
3 5 + 2 /
```

This is evaluated as follows: The numbers 3 and 5 are pushed onto the parameter stack and the word + is called, which removes them. The sum is computed and the results, 8, is pushed onto the stack. Next, 2 is pushed and / is called which then divides 8 by 2 and pushes 4. Notice no parentheses were needed to ensure the addition was done before the division. If we had wanted to express 3 + (5/2), we would have written:

```
3 5 2 / +
```

Because FORTH uses the stack in this manner, it is called a stack-oriented language — which has a great deal to offer.

Because words can leave more than one argument, FORTH can perform many computations more efficiently (and therefore faster) than its non-stack counterparts such as BASIC. For example, when a division is performed internally both the quotient and the remainder come out naturally. But BASIC allows only one value to be returned from an operation. If both the quotient and the remainder are wanted in BASIC, the division routine must be called twice — once more than needed. FORTH does not suffer from this. The FORTH word /MOD takes two numbers and leaves two — the quotient and the remainder.

Neither BASIC nor FORTH support complex numbers, but they can be added to both languages. Since complex numbers have both a real and an imaginary part, they must be represented by two numbers. This presents problems in BASIC. How would 3 + 5i and 4 + 2i be multiplied? BASIC allows only one value to be returned from a function and we need to return two. In FORTH, simply write a word C* — which takes four numbers off the stack and leaves two. C* would have the stack notation

```
( rl il r2 i2 — - r3 i3 ).
```

To do the multiplication above, we need only write:

```
3 5 4 2 C*
```

Doing this, the result is left on the stack. Of course, a complete set of complex operators could be written and used just as easily.

Since BASIC does not allow more than one argument to be returned from a function, many simple programs become quite complicated. Any program which needs to read a joystick becomes cluttered because a single stick value must be broken into its horizontal and vertical components. A BASIC program which has to perform a simple function like returning the current (x,y) cursor position also becomes cluttered. Stack-oriented languages do not suffer the same problems.

Extensible Defined

FORTH is one of few languages which are extensible. This means the FORTH compiler can be extended and made more useful. Everything in FORTH can be changed to suit individual preferences. This cannot be done in BASIC.

If a Microsoft BASIC program must be translated to Atari BASIC, there will most probably be statements Atari BASIC does not have such as the WHILE ... WEND construct. If a WHILE...WEND is in the original program, it must be recoded into recognizable keywords. This may take a great deal of time.

In FORTH, however, if a WHILE...WEND is needed it can be added to the compiler and the program itself does not need to be recoded. Hence, the compiler is extended. For example, there is no FOR ... NEXT structure in FORTH. The DO...LOOP structure is very similar to a FOR...NEXT loop, but it is not as nice as it could be. Take the following program:

```
10 REM:   TEST
20 FOR I = 1 TO 10
30 PRINT I
40 NEXT
50 END
```

In traditional FORTH, this is written as:

```
: TEST      ( - - - )
  11 1 ( last + 1 1st )
  DO I . CR ( . means PRINT )
  LOOP ; ( CR means c/r )
```

If we were to extend the compiler, we could have:

```
: TEST      ( - - - )
  1 10
  FOR
  I PRINT
  NEXT ;
```

Screen 10 actually contains the source code for the FOR...NEXT structure. This extensibility makes FORTH so powerful Charles Moore, creator of FORTH, was able to write a BASIC compiler in only eight screens of code!

Dear Dr. Quatro,

I've looked at the definition of FILL in my FORTH and it is defined using the word CMOVE. I'm confused, what's up?

Yottsu Daiskii,
Yokohama, Japan

Do you ever wonder where the programmers get the name "hacker"? Well, I'll tell you. They get this name because they sit around all day thinking up the "hacks" — the programming tricks. In FILL, CMOVE is a hack. If you put a character in the first byte, move it to the second byte, then move the second byte to the third byte, you are filling memory.

Some Assembly Required

by Robert Peck

Last issue presented a program for bulletin board operators to help them screen the incoming messages. This installment will cover one of the techniques used in that article.

When programming directly in machine language, the system can perform many times faster than when programming in BASIC because it is doing ONLY what it has been asked. It does not have to do the tasks of translating or interpreting everything every time it gets to a set of program statements.

But because each item the machine is to do must be detailed, it may take more time to write the program in the first place. Since BASIC is often too slow, a programmer may be able to make best use of time by combining it with machine language. The parts that do not need to be especially fast, but easy to program in BASIC, can remain in BASIC. The parts that must be fast can be done in machine code.

USR Function

```
(ADR(E$): Where the machine code is located.
ADR(A$): First Parameter
ADR(B$): Second Parameter
LEN(B$)): Third Parameter
X = USR( ADR(E$) , ADR(A$), ADR(B$), LEN(B$))
```

Fig. 1

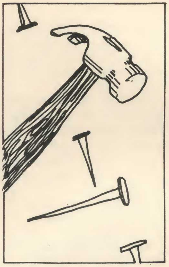
Atari BASIC provides the USR function for this purpose, which is highlighted by the fact it supplies as many machine code groups as desired. The function, however, must be instructed where the routine is located.

Figure 1 is a sample use of the USR function, taken from the last article.

This function allows the user to "pass parameters" to the assembly language program. A parameter is a number in the range Ø to 65535. If there is a fractional part to any of the numbers being sent to the machine, they will all be rounded down to the next lower integer value.

The postal clerk examples discussed in the second part of this series will help in comparing the parameter passing to a mailman's duties.

Atari BASIC will take apart the USR instruction line and change all the functions into numbers. When it has all the numbers in the correct form, it is ready to pass control to the machine language processor. Notice how the central processing unit (CPU), which is the machine language processor, has been doing the work while Atari BASIC is active.



This time, however, BASIC knows there is another language which is to be obeyed and it simply acts to interpret, separate and count the "stack" of numbers which are to be given to this other language. In this way, no matter how many numbers are given, BASIC can provide a warning to the other language.

In fact, since the stack area of the machine is somewhat limited, the other language had better get rid of the leftovers before it completes its job. Otherwise, the central processor can get lost and never return control to the BASIC language.

Let's go to the central processor BASIC parameter passing box (address) and wait for BASIC to start a machine code operation. BASIC now discovers the first parameter — in this case ADR(E$) — is the location where the control is to be passed. It converts all of the other expressions or functions into integer numbers, stacks them up in a special memory area and then passes control.

The "clerk" — CPU — gets the message, goes to the stack and pulls the first item. This item tells there are three more items on the stack which must be used by the machine language processor. Actually, all of them must be removed from the stack because underneath is the address of the next instruction to be performed after the machine language processor is finished with this special instruction group. This next instruction is the one which makes the BASIC language active again.

The first part of this article showed how the numbers on the stack could range from Ø to 65535. But because this is an 8-bit machine, each individual item on the stack can hold only a number from Ø to 255. Therefore, BASIC, although with three parameters on the stack, actually used two memory locations to hold each of them. There will be twice that many, or six bytes of data on the stack, passed from BASIC.

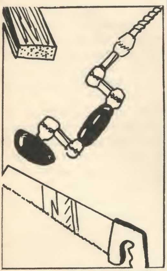
In the sample program from the March 1984 issue, the first instruction commented as: PLA; discard count of items on the stack. This was simply convenient and was NOT good programming practice. Make sure the stack is emptied of the data BASIC puts there — or truly TRUST the user of the routine not to mess it up by using too few or too many parameters in the USR call.

Back to the topic.

In order to store these numbers on the stack, Atari BASIC has to split them into the two parts, as the assembly language processor. The high-byte (most significant part) is formed by the value:

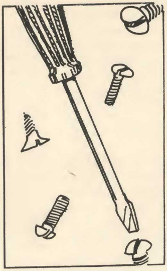
```
HIPART = INT ( NUM /
256 )
```

Then the low-byte (least significant) is formed by the value:



```
LOPART = NUM - ( HI-
PART * 256 )
```

This is actually the remainder of the division of the original number by 256.



Atari BASIC places these numbers on the stack in the following sequence:

```
COUNTOFPARAME-
TERS
HIPARTOFFIRST
LOPARTOFFIRST
HIPARTOFSECOND
LOPARTOFSECOND
HIPARTOF3RD
LOPARTOF3RD
```

Stacked under all of these items is the LOW and then HIGH byte of the return address.

Therefore, the assembly language processor must pull off all of the other items so it can see the return address when the RETURN (RTS) instruction is given.

The instruction used for this data removal is: PLA, which means pull the next item off of the stack into the accumulator area. This is an area reserved for arithmetic instructions. But it also can be used for data handling, as it is done here. A PLA instruction not only moves the data, but also changes a pointer — the stack pointer.

There isn't really a physical "stack" of items with which the processor is dealing. There is, instead, an area of memory (such as a whole line of pigeonholes) into which data such as single letters may be placed. There is a number — an address associated with each of these memory locations in the stack area.

Imagine the clerk has a sliding pointer along the side of the series of pigeonholes. That pointer is aimed at one of these slots. This is the "stack pointer". If the clerk is told to get one of the items from the slot at which the pointer points, he also will be expected to move the pointer to the next slot in line. Therefore, the next time he is asked to get an item, he will know to retrieve the next one and not just the same one again.

This is what the PLA instruction means. GET (copy, not remove) the item just above the address to which the stack pointer points, and INCREMENT (add one) the stack pointer.

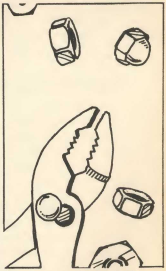
There is another instruction called PHA — PUSH accumulator value onto the stack — which does the opposite. It places a value into a memory area (replaces the old value there), then decrements (substracts one) the stack pointer.

In this way, the stack pointer always shows where the next available memory storage slot is located. In other words, if an item has been PULLED off the stack, the memory slot it used is now fair game to use again. If an item is PUSHED onto the stack, the item goes into the location pointed to "now", then the pointer is decremented so the next possible memory area will be used.

When reading a memory location, its value is not changed, but is only copied from the source to the destination. Therefore, all the data from the stack would not immediately have to be copied as was done with the sample program last time.

Instead, keep track of the pointer and simply pull (copy) the items from the stack as needed. This topic will be discussed in future articles. At this point, however, moving all of the data was the safest and quickest way to show how the data was being passed from BASIC.

Once the parameters have been removed, a Return From Subroutine (RTS) will revert back to BASIC. Of course, the instructions in between will be expected to do some useful work. To pass a value back to BASIC in a return, before the RTS, put that value into addresses $D4 (low byte) and $D5 (hi byte). These are addresses 212 and 213 in decimal. Atari BASIC will assign to X a value from Ø to 65535 by the formula:



```
X = 256 * PEEK(213)
+ PEEK(212)
```

From Page 21

support the machine with software and peripherals. Instead, TI chose to protect its secrets and gluttonously kept the software and peripheral business to itself. But the Peripheral Expansion System TI offered was not priced in line with its reduced-price TI99/4A. The expansion system was difficult to understand and confused many retailers and consumers.

In a hard-fought and grievous battle, TI suffered serious damage.

Meanwhile, Atari confidently defended itself, introducing a new game system (5200) and line of computers (the XL series) while avoiding the price-cutting for as long as possible. Cries from the masses for lower prices, however, eventually won out and the defending champion conceded to lower prices.

Rumors became prevalent IBM would return to enter the late rounds of jousting for the low-end market — a cause for concern by many of the weaker players.

Coleco continued to promise a system that would knock the competition off of their steeds, estimating shipment of 500,000 Adams by the end of 1983.

The Timex/Sinclair team faded into the background as the price cutting brought more elegant computers into the same price range as the ZX-80.

Champion Emerges

Then, an event occurred catching some competitors off guard — something few expected would have such a devastating impact on the jousting competition. Commodore followed through on its promise to put the C64 into the mass market. This computer upstart sported a 6502 CPU (the same used in the Apple, Atari and VIC-20, and manufactured by a Commodore subsidiary), 64K random access memory, 16 colors, a sound synthesizer, sprite graphics and a full-stroke keyboard. Riding this steed, Commodore began to become the favorite of the crowd to win the 1983 jousting championship.

Commodore had a definite advantage over the competition in the low-end market — it manufactured its own chips. "Commodore's manufacturing abilities and vertical integration allowed us to make money even at the low prices," said Neil Harris, a communications executive at Commodore. This allowed Commodore to slash away at the competition with price decreases. By the Christmas of 1983, the retail price of the C64 had been slashed to as low as $188.00 and its forerunner, the VIC-20, was selling for $84. At these prices the Commodore computers could not be ignored.

Texas Instruments was devastated and decided to drop out of the competition. In late November, Atari announced the price of its XL series machines would increase in early 1984 and production of the new machines would be limited until that time. Commodore took the reins of the 1983 Christmas selling season.

Atari suffered large losses for the year and the Great Blue One's rumored appearance into the low-end proved not to be so low-end in price. The PCjr actually was at the high-end of the home market, selling for approximately $1,300 including printer and disk drive. Coleco only was able to ship 93,000 of the promised 500,000 Adams and rumors of poor quality in the system were flowing freely.

Commodore clearly was the champion for 1983, and by early 1984 had sold 2 million VIC-20s and 1 million Commodore 64s.

Damage

The last six months of the '83 joust were extremely damaging for home computer manufacturers and retailers. Battered Texas Instruments showed a year-end loss of more than $660 million.

Atari, although a survivor due to its previous — and continuing — success with its game machines (selling 1.3 million in December 1983 alone), suffered losses of more than $500 million for the year.

J.C. Penney Co. announced it would stop selling home computers because it was unable to get systems and could not make an adequate profit margin on what it did receive. The company announced cancellation of remaining orders for the Adam because the system "repeatedly" failed to meet quality tests performed by its Merchandise Test Center.

Treaty Call

Amid the devastation of 1983 Commodore showed a year-end profit of $74.5 million.

Market analysts and manufacturers alike predict 1984 will be a more stable year in the low-end home computer market. Bill Heintzman, an account executive with Thomson McKinnon Securities, Inc., Orlando, Fla., contemplated the 1984 home computer stocks: "They will gain, but in moderation as compared to last year."

With the 1983 price war at an end, indications seem bent toward more conventional competition, concentrating on improving system user-friendliness and developing uses for the hardware already on the market. Industry leaders at the January 1984 Consumer Electronics Show said IBM's entry into the market with its higher-priced PCjr will help bring an end to the price wars that resulted in more than $1 billion in losses for home computer manufacturers in 1983.

Commodore's Harris agreed: "I think the price slashing is over. Atari and Coleco signaled that when they raised the prices on their products and Commodore signaled that when we did not announce any further price decreases at the January Consumer Electronics Show.

"I think the price-slashing was very destructive to this industry and as you can see by reading the quarterly reports from Atari and Texas Instruments, they show huge, huge losses ...We're happy it's over."

This may sound odd coming from an executive at the company that used price-slashing to capture the glory in 1983. However, recent changes at Commodore — i.e., the resignation of founder Jack Tramiel and four other top executives on Jan. 13, 1984 — could be an indication of a turnaround in Commodore's philosophy. One former Commodore executive is reported to have said: "There will definitely be a shift to a new school of thinking."

Tramiel's method of operation was a one-man, reckless style, many feel.

Irving Gould, Commodore chairman, reportably is installing his own more conventional strategy. But Jim Gracely, Techinical Editor of *Commodore Magazine*, doesn't express the same feelings. "I wouldn't say things are in turmoil. We haven't seen much of a change. He's (Tramiel's newly installed replacement, president Marshall Smith) not making big changes," Gracely said.

What's in Store

With IBM choosing to stay out of the low-end market, it would seem the only competitors remaining for the '83 champ are a quickly-recovering Atari, a questionable Coleco and a long-shot Timex/Sinclair. Some tension among the survivors and a cautious optimism still fills the air.

"Basically what's happening in the whole (low-end computer) market where we are is that we have done such a good job killing everybody that we are running out of competitors," Harris said. "We would like to see some other people in the market. We think it will help everybody be strong if there are some other strong companies in the low-end.

"Atari has been rumored to be paring down their product line, from the large number of separate computer products that they've had, to just concentrating on a couple, but we don't really know for sure. We're just waiting to see what they are going to do," Harris said.

"Where we see the competition on the low-end is really not too clear right now. Atari is the only one in there with us, and their sales are so much below our's, it's really pretty pitiful.

"We see IBM and Coleco as certainly being a force, although Coleco seems to be fading because of their own internal problems. Coleco seems to have really dropped the ball altogether with the Adam due to technical problems. The people who are targeted to buy that product are not the kind of people who can fix it if it breaks, which it tends to do fairly often.

"I think the IBM PCjr is going to do fairly well, but it is mainly a competitor to the Apple, which is at a higher end. The SX-64 (a portable version of the C64) is going to give even the Peanut a run for its money."

Scott Badler, an Atari media specialist, said of the alleged turmoil at Commodore and its effect on Atari: "We've had our share of turmoil. We can't spend too much time worrying about the other guy. We've slowed down for a while, but we will continue with our plans."

Atari also has a newly-installed president in former Phillip Morris executive James J. Morgan.

"Overall, he tells us we must be much more productive. The main thing he wants to establish is more of a corporate culture," said Badler.

Of Kings

Hardware may take a back seat this year to what many consider to be much more important at this point in the home computer's continuing growth — user friendliness.

"No, I don't believe there is a need for more sophisticated machines," said Bruce Entin, vice-president of Atari corporate communications. "Consumers need to understand what they have now. ... The need is for more sophisticated software and for the industry to explain what the machines now available can do for the consumer.

Entin continued: "We think the market is virtually untapped. Proper use of microprocessor technology is the key."

Entin emphasized Atari's commitment to the hardware market and the development of the 600XL, 800XL and 1450XLD. This is a point of optimism for Atari. Said Steven Ross, chairman of parent-company Warner Communications: Atari has solved most of its inventory problems and its video games and 600XL and 800XL computers are "selling well."

Entin also talked of Atari's thoughts for the future.

"We're committed to the home computer business, but nowhere is it written in stone that the computer has to be in a box with a keyboard," Entin said. "That is not to say we are not committed to the home computer business as it is today. We are. We're looking at future applications as far out as 1990."

As for Atari's predictions for '84, Entin added: "We don't make sales or production estimates. Anyone who does should look back at '83. The predictions in '83 were worthless by the Spring of '83. It was a very volatile market.

"We're looking for more stability in '84. There are fewer players, the price wars are over and we are looking forward to a very challenging year for the industry.

"We've got to show consumers how to use rather than why to buy. Morgan believes the consumer is king."

C64 a Winner

The defending champion of the 1983 joust is looking for another good year with the steady steed that brought it the victory.

"The Commodore 64 is by far the strongest selling home computer in the marketplace. It is selling like gangbusters," Harris said. "Even though the Commodore 64 is selling for $200 and the VIC-20 has been selling on the market for $80 or so, the 64 has been outselling the VIC by two or three to one. The Commodore 64 is going to be *the* product for this year regardless of what other products are around."

The other products that could be around include the Commodore 116, 264 and 364.

Year of User

All in all, the 1983 joust, though destructive to some, brought the price of home computing within the reach of many. With sales continuing at a steady pace in '84, the installed base of home computers is growing rapidly to very significant numbers.

Many first-time computer users are looking to companies like Atari and Commodore for help in understanding how to use these machines. Unfortunately, the slim profits and lack of educated sales people made it difficult for both retailers and manufacturers to provide the user support in '83.

The more stable '84 market should allow the companies to concentrate on these users. Look for '84 to be the year the home computer makes a giant leap toward being an everyday influence on our lives and a useful tool in our homes.

From Page 18

rect attempts you are allowed to make are determined by your rank in the British Army.

You may begin the mission in any of seven ranks, from a Lance Corporal to a Lieutenant. Just remember, the higher your rank, the more difficult your mission.

Summary

With three beautifully rendered screens, a realistic, if not terrifying, scenario and gameplay that requires quick reactions and a sound strategy, *London Blitz* is one of the most exciting and original games to be released for the VCS in the past year.

If you like games you can take home, pop into the cartridge slot and play well without reading any directions, then *London Blitz* may not be your cup of tea. But if you want a game that will challenge your mind as well as your reflexes, then *London Blitz* is a jolly-good choice.

— Lloyd Davies

Gateway to Apshai
EPYX
Sunnyvale, Calif.
Atari 400/800/1200
Commodore 64
Cartridge$30.00

Gateway to Apshai is a *Dungeons and Dragons* sort of game where you create a character and then explore a labyrinth of mazelike hallways and rooms.

The goal of the game, which is in machine language, is to score as many points as possible by obtaining treasures and killing monsters. As you explore any one of the 16 multi-level dungeons, you will obtain various armor, weapons and spells to assist you in dispatching the monsters to their doom.

As a general rule you will normally fight most monsters, however, there are certain monsters you will have to confront in a different manner. For example, the vampire monsters have to be defeated with a cross.

Gateway to Apshai, which is well designed, is controlled by the joystick and function keys. It is played on a scrolling window and includes several information screens that cán provide the player with data. The sound effects are good and they fit the game theme well, however, I found the graphics to be the best feature.

There are, however, two aspects of the game I found fault with. First, the manual was written for the Atari version of the program and references were not covered for the Commodore 64. Also, it did not give complete coverage on how to play the game.

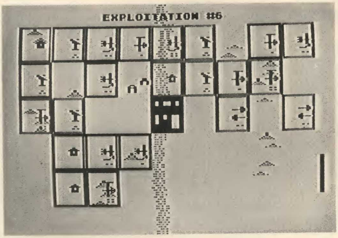
Second, this game can last for hours and the program does not support a save game feature. It is for this reason that I've yet to get past level 9. I just don't have the time to play an entire campaign in one sitting.

In summary, the game is well designed, fun to play and will definitely take a place on my "favorites" shelf, but there is still room for improvement.

— Stephen Reed

M.U.L.E.
Electronic Arts
San Mateo, Calif.
Atari 400/800/1200
Commodore 64
48K Disk
$40.00

I admit I may not be the best person to review video games. I love them too much to be unbiased and I've played so many of them that I'm no longer thrilled by the simple motion across my TV screen. But, somehow, I don't think I'm alone. Aren't we all looking for a little more nowadays?



M.U.L.E.

Unfortunately, even though our video tastes have matured, it seems the industry has hardly moved at all. No wonder profits are down. But even as pundits proclaim the passing of the video game, a bunch of guys living in Arkansas have been busy writing a program that may revitalize the industry. It ought to make them rich anyway.

I'm talking about Ozark Softscape and its new game *M.U.L.E.* (distributed by Electronic Arts). The acronym stands for Multiple Use Labor Element, and the game stands for family fun of the first order.

M.U.L.E., which begins with the most captivating musical come-on I've ever heard, is descended from the classic computer game *Hammurabi*. In this economic simulation, you and three other colonists must compete for resources on a barren planet with only your *M.U.L.E.s* for help.

The colonist who makes the most money wins. Lest you become too rapacious, however, the colony as a whole also must do well, or all the colonists will be sent home to work in a *M.U.L.E.* factory.

Sounds a little like life, doesn't it? Don't tell the kids, but playing *M.U.L.E.* is educational. I learned more about practical economics by playing this game than in all my college economic courses.

There are three different levels of play: Beginner, Standard and Tournament. The rules and speed of play vary slightly from level to level. I'll describe the Standard game here.

Up to four people may play, each with a joystick. No keyboard input is required. If less than four people play the computer will play for the remaining colonists.

The game begins with each player choosing one of eight characters. Each is animated and just watching their distinctive walks is immense fun. You may be a Mechtron, Bonzoid, or Flapper, a Gollumer (which looks suspiciously like ET), Packer, or Spheroid, or (for the more conservative players) a Humanoid.

After you've picked your characters, they'll march single file onto the screen for the first Status Summary. Each of the 12 rounds end with this counting up of assets, both of the individual and of the colony. These balance sheets determine who is winning, and, at game's end, who has won.

Each round consists of several phases. After the Status Summary, plots of land are distributed one to a colonist. There also may be a land auction — the perfect time to shop for the colonist who has everything and wants more. The planet Irata has a river valley, mountains and plains. The mountains are best for mining, the plains for energy production and the river valley for growing food.

Once you've selected your plot you must decide what commodity to produce on it. Each player takes a turn buying a *M.U.L.E.* and outfitting it appropriately. You then install the *M.U.L.E.* on your plot of land. If time allows, there are various leisure activities to be indulged in after the hard work of settling your land. You may gamble in the casino, or hunt the Wampus (a tribute to another classic computer game).

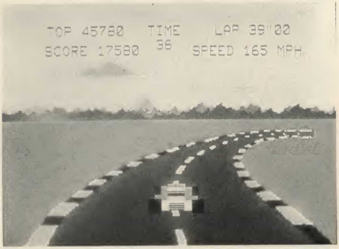
In between turns, chance may choose to smile (or frown) on a colonist. *M.U.L.E.* beauty contests, inheritances from long-lost uncles and off-planet investments may fatten the coffers of the lucky planeteer. On the other hand, pirates, pestilence, mechanical malfunction or runaway *M.U.L.E.s* may plague the unlucky player.

After each colonist has settled their plot, it's time to relax. Now you know why you brought the *M.U.L.E.s* with you. Given their share of food and energy, they'll do all the work of production on your plot and you reap the profits.

Irata, which is capitalistic, has a finely tuned economy. There is an abundance of natural resources — enough wealth for all the colonists — but human greed, and sometimes human incompetence, may cause an imbalance in the economic ecology. These economic tides are reflected in the auction phase of the game.

Each colonist brings all he has produced to the market. Here he may auction off his surpluses and purchase his necessities for the next round. It is in this phase the player will gain the keenest insights into economics. And, interestingly enough, there is a lesson to be learned — no player can succeed without the help of the others. You may get rich on Irata, but you will not win the game by sacrificing the rest. You must temper your avarice with a measure of community spirit.

After the auctions, the players march on the screen again for the Status Summary. Each player's wealth in money, land and goods is tallied and the next round begins. At the end of 12 rounds the player with the most personal wealth wins.



Pole Position

Of necessity, this brief description of *M.U.L.E.* leaves out many of the fine points that makes this such a wonderful game. For instance, players may collude to leave another player out in the cold. Collusion is the element that makes the board game *Risk* such a success. It has the same powerful impact here. And the computer makes a perfect referee.

Most importantly for the long-term enjoyment of the game, there is a great strategic depth to *M.U.L.E.*. The inter-relation between the players and the commodities they produce is incredibly subtle. Any number of strategies can be attempted and no doubt many of them would work to win the game.

But there's more to *M.U.L.E.* than great strategy. This game is exquisitely designed. The auction screens are done in seven colors and are beautifully animated. The music and sound effects are witty and appropriate. The joystick controls are logical and simple to use, and the pacing of the game is perfect.

Despite the complexity and depth of the simulation behind the scenes, *M.U.L.E.* is simple to play. You will learn the basic rules in one sitting. But it may take a lifetime to master the game. The manual is informative, complete and entertaining. It even includes a discussion of economic concepts and economies of scale.

If you ask me, *M.U.L.E.* is the perfect game. From the opening music theme to the final tally of assets, *M.U.L.E.* will captivate your eyes, ears and mind as no other computer game ever has. It satisfies every criterion for a good game.

— Léo G. Laporte

Pole Position
Atari
San Jose, Calif.
Atari 5200
Cartridge
$40.95

Pitstop
EPYX
Sunnyvale, Calif.
Atari 400/800/1200
Commodore 64
Cartridge
$39.95

The winter of 1983 will certainly be remembered for its harsh weather, but in the annals of computer software, it may be remembered as the winter of race car games.

Three significant software packages of this category were introduced last winter. They are *Pole Position* by Atari, *Pitstop* by EPYX and *Rally Speedway* by Adventure International, which was reviewed in the March issue of *Hi-Res.*

First and probably best known is the coin-op classic *Pole Position*. In this one player game you control the car by your joystick and speed around a three dimensional race course.

Pole Position is unique in that there are two phases to playing the game. First, you must run a qualifying lap on the raceway in order to determine your starting position in the game. An excellent run would result in you being placed first in the starting lineup, while a poor run might put you in the back with a pack of computer controlled formula I racecars.

Once your starting position is established, you race the course trying to get the best time possible. The computer controlled cars serve as obstacles so you can weave in and out.

After playing *Pole Position* for a relatively short time, I found the playability of the game to be limited and the graphics to be the strongest aspect of the game.

In *Pitstop*, there's a perspective like *Pole Position* where you're actually in the drivers seat looking ahead of the car. But the actual racing portion of the program does not have the extended playability or good graphics.

While racing around one of the several courses, you have to maneuver around the other cars and avoid collisions as this causes tire wear. When your tires wear down they change color indicating how much wear they've endured. If you ignore the wear on your tires, they will eventually wear out and you will crash.

The unique feature of this game is the ability to go into the pits and service your car. The object is to change the tires and refuel your car in the fastest time possible. After servicing the car a flagman will signal you to get back into the race.

While this is an interesting aspect to the program, its attractiveness is short lived because the pits soon become an obstacle to racing.

While *Pole Position* and *Pitstop* shine in their own way, *Rally Speedway* gets my vote as being the preferred game of the Winter of '83.

But then again, the Winter of '84 is coming soon.

— Stephen Reed

Rana 1000 Disk Drive
Rana Systems
Chatsworth, Calif.
Atari
$449.00

In today's marketplace it is a pleasant surprise to discover that inexpensive does not necessarily mean cheap. Rana Systems has demonstrated this with the introduction of the *Rana 1ØØØ Disk Drive* for all Atari personal computers. It is one of the lowest-priced disk drives available, yet it is fully capable and quite versatile.

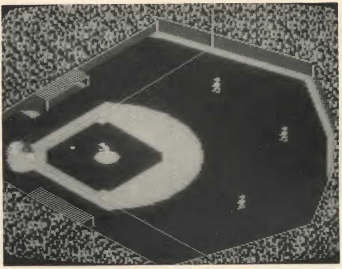
The *Rana 1ØØØ* is a compact unit, only 7x1Øx2 7/8 inches and matches the style and color of the Atari 4ØØ/8ØØ computers. The most obvious feature of the *Rana 1ØØØ* is the front panel which contains a two-digit LED display and a series of function switches. The switches are the membrane type and allow the user to display disk density, track number, drive identification and diagnostic error codes. The front panel also contains a write protection switch, as well as "power" and "busy" LED indicators.

Internally, the *Rana 1ØØØ* contains an impressive diagnostic capability. It is an intelligent drive and is able to automatically adapt to any of the three disk formats, making it compatible with the Atari 81Ø, Atari 1Ø5Ø and true double density drives. The *Rana* drive continuously monitors itself and displays an error code in the event of an internal failure. These codes are listed in the Installation Guide which comes with the unit. This well written and informative manual thoroughly describes each error code and suggests a possible course of action to resolve any given situation. For further diagnostics, the *Rana 1ØØØ* contains a row of switches on the rear of the unit which provides the capability to format a disk, select density and run both random and sequential access diagnostics.

The unit operates quietly and does not generate a lot of heat, even after extended periods of use. It is easy to install and comes with its own power supply and interface cable, which plugs directly into the serial port of any Atari computer. The other end of the cable plugs into one of two Atari-type serial ports on the rear of the drive. The remaining port is used to daisy-chain additional peripherals into the system.

Rana Systems has produced a high performance, low-cost disk drive and that should mean success for the *Rana 1ØØØ*.

— Jim Ponce



Baseball

Star League Baseball
Gamestar Software
Santa Barbara, Calif.
Atari 400/800/1200
Commodore 64
Disk and Cassette
$31.95

With the crack of the bat, the runners were moving. A shadow crossed the infield as the ball arced into deep center. The crowd grew silent and all eyes were drawn to the centerfielder, who was racing toward the wall. Then, in an instant, it was over. The ball plopped into the fielder's glove, the stadium cheered and the game ended. Another victory for the computer.

Yes, the computer. I'm sorry. I was so wrapped up. Didn't I tell you? This is Gamestar's *Star League Baseball*.

Oh, video baseball, you say. Yes, you've played that. Who hasn't? Almost every computer system has at least one rendition of America's favorite pastime. Nothing to get excited about, right?

Wrong, because Gamestar has dared to be different and that should be noted. Rather than adapt and duplicate the same old ballgame, they have made some bold changes. This not only gives its contest heightened playability, but also reassures the public there are still more programmers with originality.

The first change is in player perspective. Until now, most video ballgames were viewed from behind home plate, approximately 6Ø degrees above the imaginary play surface. The pitcher would throw from center screen to the lower portion — home plate.

In Gamestar's rendition, we're all the way up in the right field bleachers, and wait until you see the action from here. With the pitcher now throwing right to left in 3-D fashion, we can watch the ball (and its shadow) dip and dance at the batter. When contact is made, the ball moves realistically, either bouncing through the infield or sailing with convincing flight toward the warning track. The illusion is most apparent when a pitch is fouled back, up into the right field stands. You'll almost be tempted to lunge at your television for that elusive souvenir.

The second innovation will become obvious after the ball is hit. Unlike other games where you would have to position the joystick or push a keypad to designate which fielder to move, the Gamestar team knows who covers what. The players automatically spring into action. You control their subsequent moves and throws, but you don't have to give them an electronic wake-up call. Although this facet was probably brought on by the limitations of the joystick, its imaginative result forces the offense to rely on strategy and skill, rather than on an inexperienced opponent scrambling to activate the correct fielder.

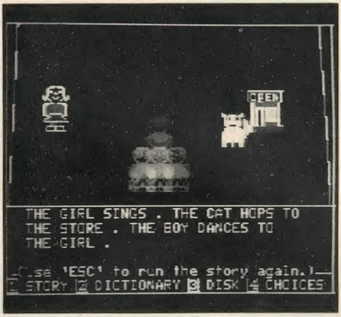
The game also incorporates other special features to insure major league quality. You can choose from different line-ups, planning your game around the single hitters or the big ball boomers. There are different pitchers to start, with a reliever patiently waiting in the bullpen. A lively organist keeps the screaming fans happy while you play against an opponent or computer in a full nine-inning game or simple batting practice.

The between innings, full screen score board adds polish to this appealing package, with enough messages to keep you amused and impressed.

— Mark Cotone

Story Machine
Spinnaker Software
Cambridge, Mass.
Atari 400/800/1200
Commodore 64
Disk and Cartridge
$29.95

Story Machine, an educational program for 5-to-9-year-olds, is a fascinating way for children to learn the basics of writing and to be entertained at the same time. In addition, the child will get experience in typing words and sentences and also become accustomed to the keyboard.



THE GIRL SINGS . THE CAT HOPS TO THE STORE . THE BOY DANCES TO THE GIRL .

(Use 'ESC' to run the story again.)

Story Machine

The part of the program that captures the young writer's interest is the graphics and sound capability. For example, if THE GIRL RUNS TO THE STORE. is typed in, the computer will show a girl on the screen followed by a few chords of music after the word GIRL is input. The same thing happens after the word STORE is entered. Following the typed-in period, the sentence is then acted out. The girl runs across the screen to the store while a variety of tunes such as "Dance of the Sugar Plum Fairy," "Hall of the Mountain King" and "A Long Way from Tipperary" are played. The ability to see the story acted out will really captivate the interest of the young writer.

The menu gives children a choice of four options: 1) watch the computer create the story; 2) alternate choosing words with the computer so you and the computer create the story; 3) input all the words yourself; 4) replay a previously saved story. On the Commodore 64, there is a fifth option available — input five names of your choice to use in place of some of the nouns on the vocabulary list in the story.

The program has a vocabulary of 47 words, 13 nouns, seven articles, four adverbs, 10 verbs, 11 pronouns and two prepositions. Plurals for nouns and verbs can be used and the rules of grammar are programmed so if you make a mistake the computer will tell you. For example, if you mix singular and plural in a subject-verb combination the computer will tell you on entry that you cannot mix them.

The screen can display a maximum of seven images at one time. If your story reaches an eighth, one of the previous images will disappear. A very handy feature is the REPLAY function. While writing you can hit REPLAY(#1) and your story will be recreated on the screen, word-by-word, image-by-image, but probably with different music. The dictionary can be called up on the screen by hitting the 3 key. It is also included in the instruction booklet.

Story Machine is a real winner for youngsters. It encourages them to construct proper sentences in grammatical form, while using a very good display of graphics and sound.

— Mal Bowden

Suspended
Infocom, Inc.
Cambridge, Mass.
Atari 400/800/1200
Commodore 64
48K Disk
$49.95

Suspended is an excellent adventure game and possibly one of the most advanced games to play. It is a completely text-oriented game with no pictures or sounds. Everything is done in the imagination of the player, with a large amount of detail supplied by the game.

Let me explain the story of the game. You are put into suspended animation in the world of Contra. If an emergency occurs and the automatic controls are not working, then you are to be awakened. Unfortunately, an earthquake has occurred, damaging the cables in the primary and secondary control systems. You, meanwhile, find yourself underground in a complex of equipment and machinery.

Since it is dangerous to remove you from your cyrogenic chamber, you will have six robots, each representing different skills, abilities and senses, to act for you. Their names are Auda, Iris, Poet, Sensa, Waldo and Whiz.

Auda represents the sense of hearing and can hear if problems develop. Iris is the sense of SIGHT, a most useful attribute. Poet knows about electronics and can diagnose electrical flows. Since he is a poet, his answers may not make much sense. Sensa perceives things magnetic and electronic. Waldo has the sense of touch and a great deal of dexterity, so he is the main worker robot. Finally, Whiz commands the informational computer and can do errands.

The object of the game is to fix all of the equipment. In order to do this, you have to assign the correct robot to the correct job. To learn all the details and to get it right is what makes this game a lot of fun.

Besides the normal difficulties of *Suspended,* there are other details used to make the game more realistic and enjoyable. First, the game is played in real time. There are people on the surface of Contra that are being killed because of the problems. The longer you take to solve them, the more people killed and the lower your score. Second, the game changes as you play. There are some events that occur during play. The first event is an earthquake at cycle 15 which causes an acid spill. The acid will kill any robot that passes through until the acid is shut off. At cycle 75, there is another earthquake which destroys the hydrophonics and transit equipment on the surface. These have to be repaired quickly because of starvation that occurs. Finally, at cycle 100, the surface humans break into the underground complex to find out why you have not repaired everything.

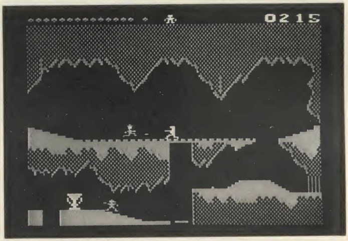
To command your robot, use complete sentences, such as Poet, go to the weather control, or Whiz, take container and grasper. Aside from the complexity of the commands possible, *Suspended* has the touch that makes it an excellent game.

You can SAVE the game at any time and RESTORE it after you have destroyed too many robots. You can use the LOOK AROUND command to find out information and if you have a printer, you can use the SCRIPT command to get a written transcript.

The instruction manual is complete and explains the background needed to start the game. There is a good quality board supplied for the Underground complex, and plastic stick-on-tokens representing the robots are also supplied.

This is an excellent game. It is imaginative and very well thought out. It is a puzzle worthy of the many hours which are needed to solve the game. Infocom has made a worthy successor to the *Zork* series.

— Sol Guber



Pharoah's Curse

Pharaoh's Curse
Synapse Software
Richmond, Calif.
Atari 400/800/1200
Disk or Cassette
$34.95

What do a mummy, a pharaoh, an explorer and a ton of treasures all have in common? They tie into the game *Pharaoh's Curse*.

This game gives you the feeling of how Indiana Jones felt when he was trying to find treasures.

In this fast-paced action game, you maneuver your character who is trying to find the treasures of the pharaoh. Sounds easy, right? Not at all. Because while trying to find the treasures you're being chased by pharaohs, mummies and hands that pop out of the ground.

This *Tutankham* takeoff is marvelous and the joystick capability is magnificent.

— Reid Nicholson,
Age 11

When I'm 64
The Alien Group
New York, NY
Commodore 64
Disk
$25.00

Because of the Commodore 64's SID (Sound Interface Device) chip, the computer is said to have greater musical capabilities than almost any other low-priced computer on the market. However, if you've ever tried "peeking" and "poking" around with the C64 to produce even the simplest tune, you realize that not only do you have to know a thing or two about music, but you also need a degree in computer science before you can get the thing to play "Mary had a Little Lamb."

This is because each parameter of a single note has to be specified at various locations in the computer's memory. For example, before you can produce a note, you have to POKE a value into 54296 to indicate the desired volume, another at 54276 to select a waveform, additional values for attack, decay, sustain and release to describe the notes envelope and two more values to assign a pitch. Finally, you will have to specify the duration of the note with a FOR statement.

When I'm 64 is a menu-drive program which takes a lot of the tedium out of creating music with your computer by allowing you to establish parameters for SID's three voices with a few keystrokes. In addition, you are able to record one voice at a time, either by playing in real time, or by selecting the notes one by one and adjusting the timing later.

When recording, the top two rows of keys are used like a piano keyboard with "I" as the middle-C. After each voice has been recorded, you can accent specific notes and add vibrato as well. When the entire tune meets your approval, it can be saved on a disk. It's almost like having your own multi-track recording studio. But that's not all.

The Alien Group also produces *The Voice Box*, a $129.00 speech synthesizer that plugs right into the C64's user port. Together, *The Voice Box* and *When I'm 64* allow you to include a vocal part in your compositions. That's right, now your computer can sing along!

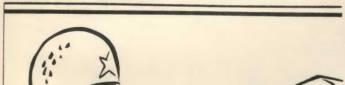
To write the lyrics of your compositions you must enter them into the computer by spelling each word phonetically. This can take a little getting used to at first, but with the aid of Appendix III in the instruction booklet (a list of acceptable phonemes), you should be able to easily input the lyrics to a short song in no time. When you're done, you can adjust vocal expression by adding vibrato and glissando (a gradual sliding between notes).

Also included is an animated face that sings the songs you've created. Although the process is time-consuming because you have to change one pixel at a time, you can redraw the face to suit your liking. After you see it, you'll know why they included this option.

When I'm 64 will let you tap into the Commodore SID chip to produce a three-part tune with voice. There is still a lot of work involved, but then who ever said the recording business was going to be easy? The only major flaw I can find is there is not any graphic representation of your music on a staff or anything else that resembles sheet music. Therefore, unless you're an improvisatorial genius, you'll have to do the actual song writing elsewhere.

Remember, to get the most out of any music program, you should have a basic understanding of composition. *When I'm 64* will not teach you how to write good music, but if you have the patience, it will help you unlock your creativity and save your masterpieces for prosperity.

— Lloyd Davies

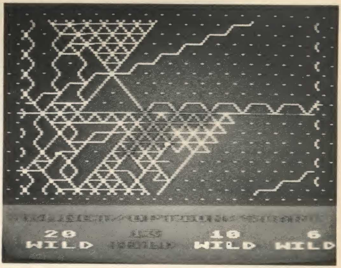


Worms

Electronic Arts
San Mateo, Calif.
Atari 400/800/1200
Commodore 64
48K Disk
$35.00

What's this *Worms*? They want me to review a game callėd *Worms*? For this I went to college?

Let's see. The usual stylish Electronic Arts package. Inside an interview with the author, David S. Maynard. He says he got the idea from Martin Gardner's Mathematical Games column in *Scientific American*. Good column — maybe a good game too? The package says not to read the instructions. Just to start. Great! I never liked reading manuals anyway. Let's boot this thing up.



Worms

(Fifteen minutes later.)

I think I'd better read the instructions.

(Two minutes and four pages later.)

Simple enough. There are four "worms," each capable of moving in six directions. Each worm exists in one of five states: new, same, auto, wild or asleep. New worms do not know how to move — I train them. Or I can use the same worm I trained last game. Auto worms are trained by the computer. Wild worms move randomly. Worms that are asleep skip the game entirely.

Did I say game? That may not be the correct word. It's more cooperative than competitive. The playfield is a matrix of dots. The worms move from dot to dot provided they are not obstructed by the path of another worm. Points are scored by the worm that blocks the last path from a point. The game continues until none of the worms can move.

Let's try it. I'll start with one worm on the board and put the rest of them asleep. The worms are trained using the keyboard or a paddle. It's really quite simple — just point the worm in a direction and press the trigger to send it that way. It will move one step and wait for further instructions. As soon as it recognizes a pattern, it will follow that pattern until it finds itself in an unrecognized situation. Then I'll have to tell it what to do and off it goes again. Each step it takes is accompanied by a flash of color and a musical tone. Very nice, very pretty, very boring.

Maybe if I add another worm.

To make a long story short, I've been playing with this program for two weeks. It hasn't gotten any more interesting. Somehow, I feel like I'm at fault. Isn't Electronic Arts supposed to be pushing back the frontiers of home computer software with innovative and brilliant new games? Isn't this one of those games?

No. This is a game that draws pretty pictures on the screen with musical tones. It's worth about an hour of your time. Considering the price of *Worms* that's an expensive hour. Thanks, but no thanks.

— Léo G. Laporte

Lights, Camera, *ACTION!*

Action!
Optimized Systems
San Jose, Calif.
Atari
Cartridge
$99.00

I popped the fluorescent orange cartridge into its slot and slammed the hood shut. Let's take this baby out for a spin.

Switch on the ignition, put it into low, and let's go. Pretty smooth, pretty responsive — so far. Let's really test this thing out. I floored it. Zero to 6Ø in 1Ø seconds. Woowee! Goodbye BASIC, farewell Forth, here comes *Action!*. This is the best thing to happen to Atari since Nolan Bushnell figured out people would play ping-pong on a TV screen.

Whoa, fella. Slow down. That's a pretty bold statement. Yeah I know, but *Action!* is such a revolutionary product.

Action! is a cartridge-based programming language that requires 16K bytes random access memory. A cassette or disk drive is recommended for saving your programs and everything you need to write *Action!* programs are included in the cartridge: a complete text editor, the *Action!* compiler and a monitor to debug your programs.

The editor is the best I've ever used. It's a full-screen editor that uses all the Atari's built-in features. Also, it supports:

- A split screen mode that allows the simultaneous editing of two files.
- A comprehensive search and replace function.
- Text lines as long as 24Ø characters.
- A "paste" buffer of almost unlimited size to move or copy blocks of text.
- "Tags" that mark your text, so you can jump from place to place with ease.
- Movement through the text a line or a page at a time.
- Type-over or insert modes, allowing you to choose between typing over what you've already written, or automatically inserting your additions between existing text.

In fact, except for the lack of word wrap and printer formatting commands, this is a complete word processing system that's very responsive. After using the *Action!* editor for a few hours, other software seem sluggish.

Action! was designed for people who program in Atari BASIC. Now, I'll admit I have a prejudice against BASIC. I use it because there has been nothing better. Sure, BASIC has its good points — it's universal and friendly, you can run a program, stop it if it doesn't work, make a few changes and continue exactly where you left off — but you pay a price for this convenience. It's s-l-o-w.

Action! on the other hand, is very fast. About 2ØØ times faster than BASIC. The only language that is faster is assembly language, and that is just by a little. (Look at the Benchmark Table to get an idea of how fast *Action!* is.) Our benchmark program calculates the first 1,899 prime numbers 1Ø times for a more accurate timing. BASIC takes nearly an hour to do it. Even using a very good C complier on a Z-8Ø computer running twice the speed of the Atari, I couldn't improve *Action's!* time by more than a few seconds. Also, it compiles programs just as quickly. A 13K byte source program will translate into machine code in less than 3Ø seconds.

There is one drawback, however. BASIC programmers may be thrown by the syntax of *Action!*. There are not any line numbers, no GOSUB or GOTO statements. But don't worry, you'll get used to it — you may even grow to like it!

Action! is based on structured, procedure-oriented languages like PASCAL, C and PL/I. The program consists of a number of independent modules called procedures or functions, each with its own task to perform and its own set of variables. The last procedure in the program puts all of them together in the right sequence. For example, if you were to write a program to describe your morning run, it might include the procedures: GET-DRESSED, GO-OUTSIDE, LEFT-FOOT-FORWARD, RIGHT-FOOT-FORWARD, TOO-TIRED, GO-HOME and so on. Your final procedure, GO-JOGGING, puts them together like this:

```
GO-JOGGING:

GET-DRESSED
GO-OUTSIDE
DO
LEFT-FOOT-FORWARD
RIGHT-FOOT-FOR-
WARD
UNTIL TOO-TIRED

GO-HOME
```

That's how an *Action!* program works. If necessary, any procedure may communicate with any other by passing along numeric values.

In the *Action!* version of the benchmark program, Prime seive() does all the work. Above it are some variable definitions. These are called declarations, and this is another area where *Action!* differs from BASIC. All variables must be declared. That is, you must tell the computer each variable's name and type before you use it. The fundamental types are CHAR for ATASCII letter, BYTE for numbers in the range Ø-255, CARD for Ø-65,535, and INT for −32,768 to +32,767. *Action!* also supports one-dimensional arrays, strings, pointer variables, and a limited record type.

You may give any variable a starting value, or tell the computer where to store the variable in memory. They may be declared globally, so they are available to the whole program, or locally, or so they are available only to the procedure in which they are used.

Action! allows you to store a procedure anywhere in memory. Or you may name a code block already in memory and then call it by that name in your program. Programmers can use this feature to access pre-assembled routines, or to use the routines built into the operating system. Machine language also may be typed directly into your source program.

Action!, which has simple syntax, looks a little like a stripped down C or PASCAL. It has a small set of built-in commands, but provides most of the control structures and arithmetic operators you would expect from a powerful high-level language.

Similar to C, *Action!* has no built-in commands for input or output. All I/O and most high-level functions are provided by a library of routines in the cartridge. There are nearly 7Ø prewritten procedures and functions that duplicate most BASIC commands. BASIC programmers will feel right at home with the library routines, since they're just like BASIC. *Action!* Setcolor (2,Ø,Ø) does exactly what BASIC SETCOLOR 2,Ø,Ø does.

OSS also is offering a Programmer's Aid Disk with supplemental routines, including player-missile graphics procedures, floating point functions, a powerful print formatting procedure and two games written in *Action!*. The disk is available for $3Ø and a complete *Action!* source code is included.

After you've written your program, you can use the *Action!* monitor to do simple debugging. You may examine or write to memory, examine or set a variable's value, or run any procedure or function by itself. There is a trace mode which will display the routines called by your program as it runs and the values passed to them. Also, the monitor allows you to change certain system parameters to suit your taste, or execute an *Action!* statement stand alone.

You compile *Action!* programs from the monitor and you can run the program there or save it to disk as a binary file. Programs saved this way may be run using the DOS "L" command. If, during compilation, *Action!* detects an error, it will stop and print an error message. As with most compilers, *Action!* is often in the dark about what went wrong, but it will leave the cursor in the text where it stopped. So it's fairly easy to find your mistake.

Action! compiles to true 65Ø2 machine code, but the cartridge must be in place to provide support routines while your program runs. OSS offers a run-time package that may be appended to *Action!* programs to allow them to run without the cartridge's support. The run-time package is small, less than 4K bytes, and will be inexpensive. OSS plans to charge $3Ø for public-domain and personal use, several hundred dollars for commercial software publishers.

I've been using *Action!* for a month now, and I've never seen a more powerful or useful programming environment on a microcomputer. This is all the more amazing, since the editor is only 4K bytes and the compiler only 6K.

Naturally, some compromises have been made to achieve this compactness:

- Floating point numbers are not directly supported by *Action!* They are available on OSS's Programmer's Aid Disk, but the implementation is awkward.
- There is no CASE-type control structure.
- *Action!* does not allow negative compiler constants. You can't initialize a variable to a negative number — inconvenient, but not fatal.
- You must compile your entire program at one time. Modules cannot be separately compiled. And although you can compile from disk, you cannot compile to disk, so the size of your compiled program may not exceed available memory. That's about 27K on my 48K machine.

I think the biggest problem most will encounter will be getting used to a structured language. However, in the long run this language will be easier. Learning it on your own can be tough, though, and unfortunately the manual is not much help. Many of the features of the language are ignored and in some parts the manual is wrong. What we really need is a tutorial. OSS is working on one, but until they release it, look for some late-night sessions with the computer.

Given the choice between *Action!* and any other Atari-oriented language, I would vote for *Action!*. This language is like a finely tuned racing car. There's a lot that can go wrong and it may take a little more skill to drive. If you'd prefer to stick with the BASIC station wagon, with its automatic transmission and CPU hogging engine, I won't blame you. As for myself, give me the feel of the wind in my hair. Give me *Action!*.

— Léo G. Laporte

Prime Number Benchmark

A benchmark is a program that tests the speed with which a language performs certain tasks. The following programs are based on a standard benchmark first published in BYTE Magazine. This program calculates all the prime numbers between 3 and 16,ØØØ using an algorithim called Eratosthanes Seive. Each version repeats its calculations 1Ø times for more accurate timing.

Programmers experienced in each of the following languages will no doubt find more efficient means of performing these calculations, but it is our intention that the methods used below reflect the approach a typical user of each language might take.

No single benchmark can accurately reflect the true efficiency of a language. These tests should be used

Continued on Page 74

only to gauge the relative performance of each language in the areas of data manipulation, controlled loops, and memory references. In other words, use these figures for comparison only. Your mileage may differ.

Action!

```
DEFINE  size = "8190",
        ON = "1",
        OFF = "0"

BYTE ARRAY flags(size+1)

CARD count, i, k, prime, jifs

BYTE  secs, hunds,
      iter,
      tick=20,
      tock=19

PROC Prime seive()
  tick = 0
  tock = 0
  FOR iter=1 to 10
    DO
    count = 0
    SetBlock (flags, size, ON)
    FOR i = 0 to size
      DO
      IF flags(i) THEN
        prime = i+i+3
        k = prime + 1
        WHILE k <= size
          DO
          flags(k) = OFF
          k = =+ prime
          OD
        count = =+ 1
      FI
      OD
    OD
  jifs=tick+256*tock
  secs=jifs/60
  hunds=((jifs MOD 60)*100)/60
  PrintF("%U.%U secs", secs, hunds)

RETURN
```

BASIC

```
10  SIZE=8191: DIM FLAG$(SIZE)
20  FOR ITER= 1 TO 10
30  FLAG$="1":  FLAG$(8191)=  FLAG$:
    FLAG$(2)=FLAG$
70  FOR I=1 TO SIZE
80  IF FLAG$(I,I)="0" THEN 160
90  PRIME=I+I+3
100 K=I+PRIME
110 IF K>SIZE THEN 150
120 FLAG$(K,K)="0"
130 K=K+ PRIME
140 GOTO 110
150 COUNT=COUNT+1
160 NEXT I
170 NEXT ITER
```

valFORTH 1.1

```
Scr # 1
 ( Prime Number Benchmark )

 8191 CONSTANT SIZE
 SIZE CARRAY FLAG

 : PRIME NO SEIVE
 10 0 DO 0 FLAG SIZE 1 FILL
   SIZE 0 DO
     1 FLAG C
     IF I DUP + 3 + DUP I +
       BEGIN DUP SIZE < WHILE
         0 OVER FLAG C!
         OVER +
       REPEAT DROP DROP
     ENDIF
 LOOP LOOP :
```

Language	Compile Time	Object Code	Run Time
BASIC	—	443	3140 secs
FORTH	3 sec	114	168 secs
Action!	<1 sec	423	18 secs
C (Z-80)	132 sec	290	14 secs

Advertiser Index

Advertiser	Page
(101) ATARI	2
(102) MAXIMUS	3
(103) ATARI	5
(104) INDUS	7
(105) ADDISON-WESLEY	8
(106) COMVOLOGIC	10
(107) FUNSOFT	11
(108) RADICAL	12
(109) MICRO PROSE	13
(110) O.S.S.	15
(111) COMPUTER PALACE	17
(112) RESTON	19
(113) COMPUTER BOOK CLUB	23
(114) COMP SOFT SERVICES	25
(115) PROTECTO	38 · 39
(116) MICRO PULSE	46
(117) COMPU-TECH	46
(118) PROTECTO	48 · 49
(119) COMPUTER CREATIONS	51
(120) GARDEN & GREEN	57
(121) FOREST HILLS	57
(122) SUPERWARE	93
(123) STARBOUND	93
(124) AVALON-HILL	95
(125) SIERRA ON-LINE	96

In Search Of...

by Stephen Reed

The 1541 Disk Drive from Commodore is a complete computer all by itself! This is why no Disk Operating System is required to use it as with most other microcomputers.

The 1541 Micro is complete with its own 6502 processor, an operating system in ROM, and internal RAM for data storage.

It is for this reason the use of the drive is so rigid. There is very little flexibility in the system for doing things not originally planned.

As you know by now, to load a BASIC program from disk the command is: LOAD"Filespec",8
And the command for loading a machine language program into memory is: LOAD"Filespec",8,1

There is, however, no way to ask the disk drive directly where in memory a program is going to load. This is an important thing to know so that separate programs will not conflict in RAM. This brings us to Addressit, a reasonably short utility which will interrogate a program file and then tell you where in memory it will load.

Variable List

F$	=	Filespec
A$	=	ISB Input of Load Address String
B$	=	MSB Input of Load Address String
A	=	LSB of Load Address
B	=	MSB of Load Address
D	=	Load Address in Decimal
T	=	Powers of 16 Counter
H$	=	Load Address in Hexdecimal

Addressit

```
10 REM
20 REM ADDRESSIT  VERSION 1.0
30 REM LAST MODOFIED`11/22/83
40 REM
50 REM PROPERTY OF STEPHEN REED
60 REM
70 REM
100 POKE 53281,1:PRINT"<CLR><CTL-1>"
120 PRINT"ADDRESSIT- BY STEPHEN REED"
130 PRINT:PRINT:INPUT"ENTER FILESPEC TO
CHECK";F$
150 OPEN 1,8,12,F$+",P,R"
160 GET#1,A$,B$:CLOSE 1
170 IFA$=""THENA=0:GOTO180
175 A=ASC(A$)
180 IFB$=""THENB=0:GOTO190
185 B=ASC(B$)
190 D=B*256+A+1
195 IFD=3528THENPRINT:PRINT"DISK ERROR":
GOTO300
200 PRINT:PRINT:PRINT"DECIMAL LOCATION =
";D-1
210 A$="0123456789ABCDEF":H$=""
220 FORT=3TO0STEP-1:P=0
250 D=D-16↑T:P=P+1:IFD>0THEN250
260 D=D+16↑T
270 H$=H$+MID$(A$,P,1):NEXTT
280 PRINT:PRINT"HEX LOCATION      = ";H$
300 PRINT:PRINT:PRINT"<SPACE TO RUN AGAI
N>"
310 GETA$:IFA$=""THEN310
320 RUN
```

Scrolling Joystick Banner

by Sol Guber

Several of the recent popular arcade games use a horizontal scrolling effect to simulate movement. This is a moderately difficult task since what appears on the screen has little relationship with what is in memory.

This program generates a display on the screen and uses the joystick to control the display. The display is a message the user generates. She can vary the size and shape of the letters in the message. The message can be so long that all the letters can not be seen at one time.

The joystick is used to scroll the banner. When the joystick is moved up, the banner is moved up. When the joystick is moved to the left, the message moves to the left until the end of the banner is reached. When the trigger is pressed, the banner color changes. There can be up to 30 letters in the banner.

Before explaining how to perform horizontal scrolling on a large scale, a little review of the Atari screen memory system seems appropriate.

Screen memory is a term that relates what appears on the monitor screen with a certain portion of the memory of the computer. Atari uses a floating screen memory — its exact location will vary from Graphics mode to Graphics mode. The only way the system knows the location of the screen memory is by use of a pointer in locations 88 and 89. The actual start of screen memory can be calculated by PEEK (88) + 256* PEEK (89).

No matter what the Graphics mode, the start of the screen will be this memory value. The screen is mapped linearly in memory. Say that we are in Graphics 0 and the start of screen

'Joystick Control program generates a display on the screen and uses the joystick to control the display.'

Color Your World

by Sol Guber

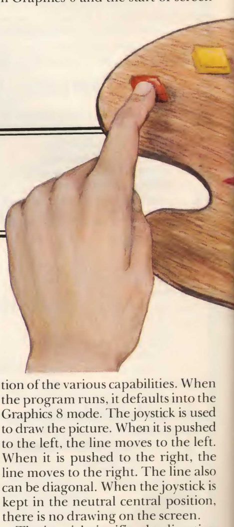
This program uses a joystick control to draw lines and figures on the television screen. There are three different types of drawing capabilities and several modifications of each one.

There is a fine drawing capability using Graphics 8 mode. Another capability uses different shades of the same color in Graphics 9. The last capability in Graphics 11 uses 16 colors simultaneously on the screen.

There also is the ability to change the thickness of the line drawn with the joystick and to fill in shapes drawn on the screen. There is the ability to save and load pictures from either disk or cassette. The different modes can be found by pressing the OPTION button on the consol. The OPTION button will remove the system from the present graphics mode and go into the MENU section. The background of MENU has a different color for each selection so little reading ability is needed.

The following is a detailed description of the various capabilities. When the program runs, it defaults into the Graphics 8 mode. The joystick is used to draw the picture. When it is pushed to the left, the line moves to the left. When it is pushed to the right, the line moves to the right. The line also can be diagonal. When the joystick is kept in the neutral central position, there is no drawing on the screen.

The joystick signifies the direction of the movement and the amount of movement is determined by the duration of the non-neutral positioning. When the line moves off the screen, a BEEP sounds, signaling you are off the screen. There are no other error messages. To clear the screen, the trigger is pressed. A new drawing



memory is 40000. Since there are 40 characters per line, locations 40000-40039 will be the first line, locations 40040-40079 will be the second line, locations 40080-40119 will be the third line and so on. From these numbers it can be seen that on the screen the value at location 40038 will be to the left of the value in memory location 40039. The value in memory location 40039, however, will not be to the left of the value in memory location 40040. It will be the end of line 1 and the value of the memory location 40040 is the start of line 2. Thus, the screen memory is linear with each value following the next one, but it is mapped onto the screen in a two-dimensional pattern.

To do large scale horizontal scrolling, it must be remembered that the screen memory is linear, but the screen itself is two-dimensional. To make the system scroll horizontally, there are two methods:

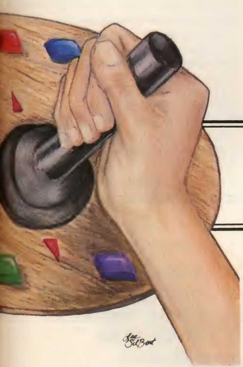
- Using the display list to vary its parameters.
- To make up a large string variable that has the information needed to be interpreted in a two-dimensional pattern.

I have used the latter method to make a pattern on the screen. But rather than use some sort of a mapping scheme or other pattern, I have used a message banner that can be moved around the screen with the joystick. The message only will show a portion of it at a time and the joystick is used to move the total message into view. Different size letters are used to vary the interest.

Figure 1 shows what must be done to have a proper picture on the screen. The periods represent the information in memory and the line represents the television screen. The end of the top line of periods is next to the beginning of the second line, etc. The variable is linear in memory. I have called this variable L$ and the length is 960 characters long, divided into eight, 120 characters-long lines. Thus, for the data to be vertical on the screen, it is 120 spaces away in memory. The reason there are eight long lines is there are eight bytes of information needed to define one character in the Atari character set. For the character T, the hex information is 00, 7E, 18, 18, 18, 18, 18, and 18. When the binary information is listed vertically (see Figure 2), the '1's will form a 'T'. When this sort of information is put into the variable L$, the first number 00 will be a memory location, the second number 7E will be 120 memory locations further on, the 18 will be 240 memory locations from the first, and so on. When the information is put on the screen, it starts

Continued on Page 81

'This mode is best for young children because there are no keys to push and the out-of-boundary beep signal can be learned very quickly.'



now can be made. This mode is best for young children because there are no keys to push and the out of boundary BEEP signal can be learned very quickly. The only controls are the joystick and the trigger button.

Another option, which also uses Graphics 8, is a slower drawing program. The line drawn by the joystick can have a different width. For example, the 'S' key can be pressed for single thickness, the 'D' for double thickness and the 'T' for triple thickness. At the end of the line will be a flashing point on the screen. The return key does not need to be pressed and has no effect when it is done. The keys can be pressed in any order and at any time to change the thickness of the line. The trigger is used to clear the screen. This option also has an eraser mode. The 'E' key can be pressed and the eraser mode can be used. This will make the line the same color as the background and thus erase a part of the screen.

The thickness of the eraser will be the same as the previous line, however, the eraser line width can be changed at any time.

To go back to the non-eraser mode, the 'E' key is pressed again. The BEEP is used to signal the out of boundary condition. This program is more suited for 5 to 8 year olds. A little keyboard recognition is needed. Using good hand-eye coordination, very complicated drawings can be made.

A final option, started by pressing the OPTION button is similar to the second option, but has a shape-filling capability with the various line widths. When the 'F' key for FILL is pressed, the program will fill in the figure. A short BEEP at a different tone will signal when the fill operation is completed. At the end of the FILL operation, the line will be back at the original position. The shape must be a closed one or the filling subroutine will escape from the figure and draw a line to the side of the screen.

Depending on the part of the program the fill subroutine uses, there may be several lines drawn before the subroutine stops. Again this will be signaled by a short beep. The trigger will clear the screen. This program is suited for 8 to adult, since the filling capability is more complicated to use. Care must be taken to move the end of the line into a closed figure. If the figure is not closed, then the line will escape. After the BEEP, to show the fill is completed, the end of the line will be in the center of the screen. The only normal keys that will work on the keyboard will be E, S, D, T, and F. The others will not be recognized. Do not use FILL in the second option since the escape from a non-

Continued on Page 78

From Page 77

enclosed figure will not return to the right part of the program.

Except for the first option, the drawing can be saved to either a disk or a cassette. When the Menu is shown, the SELECT key can be pressed rather than RETURN. When this is done, another set of commands are shown. The commands are CTRL-S (the CTRL key held down and then the S key pressed) to Save the drawing, CTRL-L (CTRL first and then the L) for Loading a picture, and the CTRL-U for Updating the picture file number.

TABLE I

OPTION	GRAPHICS MODE	LINE WIDTH	FILL CAPACITY	COLOR CHANGE	CLEAR SCREEN
1	8	SINGLE	NO	NO	TRIGGER
2	8	MULTIPLE	NO	NO	TRIGGER
3	8	MULTIPLE	YES	NO	TRIGGER
4	9	MULTIPLE	NO	TRIGGER	START
5	9	MULTIPLE	YES	TRIGGER	START
6	11	MULTIPLE	NO	TRIGGER	START
7	11	MULTIPLE	YES	TRIGGER	START

When the SELECT Menu is shown, the system asks if the picture is to be saved to a cassette or to a disk. It expects either a C or D answer. If neither of these answers are given, the question is asked again. If a 'D' answer is given, then the file name desired can be entered into the system. Be sure to use a colon after the D.

An example of a correct response is D:GAME or D:SNOOPY. Only slight checking is done to make sure the file is a proper one.

At the bottom of the Menu is the name of the file in which the next picture will be stored. There is a default name PICSAVE. Before any picture can be saved, the SELECT must be pressed at least once to give a file name where the drawing is to be saved. When you are in the Menu section, SELECT can be pressed to see the name of the next file to be saved.

During the drawing of a picture, CTRL-S can be pressed and the screen will be saved on the predetermined device. If the CTRL-L is pressed, the program will stop and ask for the name of the file to be Loaded. The picture will be in the appropriate graphics mode and will use the proper colors. If CTRL-U is pressed, then the number of the file will be Updated by one so a new picture can be saved without destroying the old file. Be sure to press CTRL-U before saving a picture so it will not write over a previous picture. This feature is not suitable for young children since it uses up large amounts of disk space quickly. Each picture saved will use 68 sectors on a disk. The save will take about 50 seconds on disk and about 4 minutes on cassette. The Load will take about the same time as the Save.

Options 4 and 5 can be found by pressing the OPTION button again. These are similar to options 2 and 3, in that they can have different widths of the lines under keyboard control. Option 5 uses the FILL capability. The Graphics mode is 9, and 16 different shades of the same color can be used. The trigger is the controller for the shades. As the trigger is held down, the different shades cycle through. To use a specific shade, the trigger is released. In both options 4 and 5, the background color can be controlled. In the Menu section, RETURN must be pressed to start the program. If a number is inputted rather than just the return button pressed, then the number will be used as the background color. The various colors are 0-15, corresponding to the 16 colors used in the SETCOLOR command.

If the number is not correct or a letter is pressed, then the system defaults into green. To clear the screen anytime during the drawing, the START button must be pressed. To go into any other options, the OPTION button must be pressed. To save the drawing, the CTRL-S key is pressed. The shades can be changed before the 'F' command is used. This option is suitable for age 8 to adult. The various shadings can be used to make quite sophisticated drawings.

Options 6 and 7 can be found by pressing the OPTION button again. They are similar to options 2 and 3 in that they have different widths of the lines under keyboard control. Option 7 has the fill capability. The Graphics mode is 11, and 16 different colors, are usable. The trigger button is the controller of the different colors. The colors cycle through the 16 possibilities as the trigger is held down. The START button is used to clear the screen. The OPTION button is used to go to the various options.

As in the previous modes, the widths of the lines can be used in any order, and at any time by pressing the appropriate button. Again the eraser option and the save option are usable. These options are suitable for age 8 to adult. However, because there are only five keys to operate under normal conditions, younger children can easily learn to use the system and make picture and designs with the various colors.

TABLE II
WORKING KEYS

S -	SINGLE THICKNESS
D -	DOUBLE THICKNESS
T -	TRIPLE THICKNESS
E -	ERASE MODE
F -	FILL MODE
CTRL-S	SAVE PICTURE
CTRL-L	LOAD PICTURE
CTRL-U	UPDATE PICTURE NUMBER

Pastel Pencil

```
1 GOTO 4
2 FOR INV=C1 TO LEN(INV$):? CHR$(ASC(I
NV$(INV))+128);:NEXT INV:RETURN
4 C255=255:C4=4:C3=3:C6=6:C15=15:C5=5:
C7=7:C8=8:C9=9:C10=10
5 GOSUB 2000
10 REM ETCH- A SKETCH
12 C0=0:C1=1:C2=2
15 FILL=11000:START=53279
20 OFLG=C1:RIGHTADR=170
30 DIM YSTEP(20),XSTEP(20),A$(12),FN$(
17),INV$(100)
40 SETCOLOR C0,C2,8
50 COLOR C1
80 XSTEP(C5)=C1:YSTEP(C5)=C1
90 XSTEP(C6)=C1:YSTEP(C6)=-C1
100 XSTEP(C7)=C1:YSTEP(C7)=C0
110 XSTEP(C9)=-C1:YSTEP(C9)=C1
120 XSTEP(C10)=-C1:YSTEP(C10)=-C1
130 XSTEP(11)=-C1:YSTEP(11)=C0
140 XSTEP(13)=C0:YSTEP(13)=C1
150 XSTEP(14)=C0:YSTEP(14)=-C1
160 XSTEP(15)=C0:YSTEP(15)=C0
165 GRAPHICS C8*C3:X=160:Y=90
168 C=C1:COLOR C
170 SOUND C0,C0,C0,C0
180 IF STRIG(C0)=C0 THEN 165
185 IF PEEK(START)=C3 THEN GOTO 700
190 S=STICK(C0)
195 TRAP 240
200 X=X+XSTEP(S):Y=Y+YSTEP(S):PLOT X,Y
230 GOTO 170
240 X=X-XSTEP(S):Y=Y-YSTEP(S)
250 FOR I=15 TO C0 STEP -C1
260 FOR J=C1 TO C6:NEXT J
270 SOUND C0,136,C10,I
280 NEXT I
285 TRAP 240
290 GOTO RIGHTADR
430 GRAPHICS GR
432 D3=C0:M=C0
433 C=C1:COLOR C:D1=C
435 SETCOLOR C0,C2,C8:RIGHTADR=440
437 IF GR=C9 THEN SETCOLOR C4,CM,C0
440 X=XX:Y=YY
470 SOUND C0,C0,C0,C0
471 IF PEEK(764)<>C255 THEN GOSUB 1472
:M=C
480 IF STRIG(C0)=C0 AND OFLG<C4 THEN G
OTO 430
482 IF STRIG(C0)=C0 THEN C=C+C1:COLOR
C:D3=C0:D1=C:IF C=16 THEN C=C1:COLOR C
485 IF PEEK(START)=C3 THEN GOTO 700
488 IF PEEK(START)=C6 THEN GOTO 430
490 S=STICK(C0)
493 IF S=C15 THEN M=D3*(M=C)+C*(M=D3):
COLOR M:PLOT X,Y:FOR KL=C1 TO 25:NEXT
KL:COLOR C:PLOT X,Y:GOTO 471
495 TRAP 240
500 X=X+XSTEP(S):Y=Y+YSTEP(S):PLOT X,Y
502 IF DT=C1 THEN PLOT (X+YSTEP(S)+XST
EP(S)*YSTEP(S)),Y+XSTEP(S)
504 IF FT=C1 THEN PLOT (X-YSTEP(S)-XST
EP(S)*YSTEP(S)),Y-XSTEP(S)
530 GOTO 470
600 REM
631 ? "PRESS 'S' FOR SINGLE THICKNESS"
633 ? "PRESS 'D' FOR DOUBLE THICKNESS"
635 ? "PRESS 'T' FOR TRIPLE THICKNESS"
636 ? "PRESS 'E' FOR ERASER"
637 IF OFLG=C3 OR OFLG=C5 OR OFLG=C7 T
HEN ? "PRESS 'F' FOR FILL"
640 ? :? "PRESS ";:INV$="SELECT":GOSUB
 C2:? " TO GET INFORMATION ABOUT SAVIN
G AND LOADING PICTURES"
699 RETURN
700 OFLG=OFLG+C1:IF OFLG=C8 THEN OFLG=
C1
705 CM=C10+C5
710 IF OFLG<C1 OR OFLG>C7 THEN GOTO 16
5
720 ON OFLG GOTO 722,750,780,810,840,8
70,900
722 GRAPHICS C0:POSITION C5,C5:INV$="G
RAPHICS 8":GOSUB C2:? :GOSUB 920
724 POSITION C2,C9:GOSUB 930
726 IF PEEK(53279)=C5 THEN GOSUB 6000
728 IF PEEK(764)<>255 THEN 165
730 GOTO 726
750 GRAPHICS C0:SETCOLOR C2,C4,C4:POSI
TION C5,C5:INV$="GRAPHICS 8":GOSUB C2:
? :GOSUB 920
752 ? :GOSUB 600:?
754 GOSUB 930
755 IF PEEK(53279)=C5 THEN GOSUB 6000
756 IF PEEK(764)<>255 THEN GR=C8:XX=19
0:YY=90:GOTO 430
758 GOTO 755
780 GRAPHICS C0:SETCOLOR C2,C1,C4:POSI
TION C5,C5:INV$="GRAPHICS 8":GOSUB C2:
? :GOSUB 920
782 ? :GOSUB 600:?
784 GOSUB 930
786 GOSUB 935
787 IF PEEK(53279)=C5 THEN GOSUB 6000
790 IF PEEK(764)<>255 THEN GR=C8:XX=19
0:YY=90:GOTO 430
791 GOTO 787
810 GRAPHICS C0:SETCOLOR C2,C5,C4:POSI
TION C5,C5:INV$="GRAPHICS 9":GOSUB C2:
GOSUB 925
812 ? :GOSUB 600:?
814 GOSUB 930
815 IF PEEK(53279)=C5 THEN GOSUB 6000
816 IF PEEK(764)=255 THEN 815
817 INPUT A$:IF LEN(A$)=C0 THEN 826
818 TRAP 824:CM=VAL(A$)
820 IF CM>16 THEN CM=CM-INT(CM/16)*16
824 TRAP 40000
826 GR=C9:XX=45:YY=90:GOTO 430
840 GRAPHICS C0:SETCOLOR C2,C0,C4:POSI
TION C5,C5:INV$="GRAPHICS 9":GOSUB C2:
GOSUB 925
842 ? :GOSUB 600:?
844 GOSUB 930
846 GOSUB 935
850 GOTO 815
870 GRAPHICS C0:SETCOLOR C2,14,C4:POSI
TION C5,C5:INV$="GRAPHICS 11":GOSUB C2
:GOSUB 925
872 ? :GOSUB 600:?
874 GOSUB 930
875 IF PEEK(53279)=C5 THEN GOSUB 6000
876 IF PEEK(764)<>255 THEN GR=11:XX=45
:YY=90:GOTO 430
```

```
878 GOTO 875
900 GRAPHICS C0:SETCOLOR C2,C10,C4:POS
ITION C5,C5:INV$="GRAPHICS 11":GOSUB C
2:GOSUB 925
902 ? :GOSUB 600:?
904 GOSUB 930
906 GOSUB 935
910 GOTO 875
920 POSITION C2,C7:INV$="PRESS FIRE TO
 CLEAR SCREEN":GOSUB C2:? :RETURN
925 POSITION C2,C7:INV$="PRESS START T
O CLEAR SCREEN":GOSUB C2:? :RETURN
930 ? "      ";:INV$="PRESS":GOSUB C2:?
 " RETURN ";:INV$="TO START":GOSUB C2:
? :RETURN
935 INV$="WHEN USING FILL, BE SURE THA
T YOU ARE":GOSUB C2:? " ";:INV$="WITHI
N THE FIGURE.":GOSUB C2:? :RETURN
1000 IF DT=C1 THEN X=X+C1:Y=Y+C1
1001 IF FT=C1 THEN X=X-C1:Y=Y-C1
1002 XX=X:YY=Y:GOSUB 3000
1005 RETURN
1472 IF PEEK(764)=62 THEN DT=C0:FT=C0
1473 IF PEEK(764)=58 THEN DT=C1:FT=C0
1474 IF PEEK(764)=45 THEN DT=C1:FT=C1
1475 IF PEEK(764)=56 THEN GOSUB 1000:G
OSUB FILL:GOSUB 3000
1477 IF PEEK(764)=42 THEN D1=C:C=D3:D3
=D1:COLOR C:M=C
1479 IF PEEK(764)=190 THEN IN408=8:GOS
UB 4000
1481 IF PEEK(764)=128 THEN GOSUB 5000:
POP :GOTO 470
1485 IF PEEK(764)=139 THEN UPDATE=UPDA
TE+1:FN$(LEN(FN$))=STR$(UPDATE)
1497 POKE 764,255
1500 RETURN
2000 RESTORE 2200:FOR J=1536 TO 1558:R
EAD A:POKE J,A:NEXT J

2200 DATA 104,201,1,208,10,104,104,141
,114,3,162,48,32,86,228,133,213,169,0,
133,212,96,0
3000 SOUND C1,25,C10,C10:FOR J=C1 TO 3
5:NEXT J:SOUND C1,C0,C0,C0:RETURN
4000 TRAP 4400:POKE 764,255:OPEN #3,IN
408,C0,FN$
4010 IF IN408=C4 THEN GET #3,GR:GET #3
,CM:GET #3,OFLG:GRAPHICS GR:X=45:Y=90:
XX=X:YY=Y
4015 IF IN408=C4 AND GR=C9 THEN SETCOL
OR C4,CM,C0
4020 IF IN408=C8 THEN PUT #3,PEEK(87):
PUT #3,CM:PUT #3,OFLG
4025 POKE 891,128
4030 TVAT=PEEK(560)+PEEK(561)*256
4040 RAMTOP=PEEK(106)*256:TVSIZ=RAMTOP
-TVAT
4070 SIZHI=INT(TVSIZ/256):SIZLO=INT(TV
SIZ-256*SIZHI)
4090 TVAHI=INT(TVAT/256):TVALO=INT(TVA
T-256*TVAHI)
4130 POKE 884,TVALO:POKE 885,TVAHI:POK
E 888,SIZLO:POKE 889,SIZHI
4160 TRAP 40000
4200 RES=USR(1536,IN408+C3):CLOSE #3:P
OKE 764,255:RETURN
```

```
4400 ? "THERE WERE PROBLEMS LOADING YO
UR       PICTURE":CLOSE #3:TRAP 40000:G
OTO 5000
5000 POKE 764,255:GRAPHICS C0:? "WHAT
IS THE NAME OF THE PICTURE       TO LO
AD":INPUT A$
5005 IF A$(C1,C1)="C" THEN FN$="C:":IN
408=C4:GOTO 4000
5010 FN$="D:":FN$(C3)=A$:IN408=C4:GOTO
 4000
6000 GRAPHICS C0:POKE 710,15:IF SFLAG=
C1 THEN 6060
6005 UPDATE=C1
6010 ? "THERE ARE THREE MORE CODES THA
T CAN   BE USED IF THE PICTURES ARE TO
 BE"
6015 ? "SAVED EITHER ON DISK OR CASSET
TE"
6020 ? :? "DO YOU WANT TO SAVE THE PIC
TURE ON    ";CHR$(196);"ISK OR ";CHR$(
195);"ASSETTE";:INPUT A$
6030 IF A$(1,1)<>"D" AND A$(1,1)<>"C"
THEN ? :? :GOTO 6020
6040 IF A$(C1,C1)="C" THEN FN$="C:":GO
TO 6060
6042 IF LEN(A$)<C3 THEN 6050
6043 IF A$(C1,C2)="D:" AND A$(C3,C3)>"
@" AND A$(C3,C3)<"[" THEN FN$=A$:FN$(L
EN(FN$)+C1)="]":GOTO 6058
6050 IF A$(C1,C1)="D" THEN FN$="D:PICS
AVE."
6058 FN$(LEN(FN$))=STR$(UPDATE)
6060 ? :? "THE CODE FOR SAVING A PICTU
RE IS      CONTROL-S"
6070 ? :? "THE CODE FOR LOADING A PICT
URE IS      CONTROL-L"
6080 ? :? "THE CODE FOR UPDATING A PIC
TURE IS      CONTROL-U"
6085 ? :? "WHEN A PICTURE IS UPDATED,
A NEW      VERSION WILL BE SEEN"
6090 ? :? "IF A CONTROL-S IS HIT BEFOR
E AN       CONTROL-U UPDATE, THEN THE
OLD PIC-   TURE WILL BE DESTROYED"
6100 REM
6110 ? :? "THE NEXT PICTURE WILL BE SA
VED AS     ";FN$
6400 SFLAG=C1
6410 ? :? "PRESS ";:INV$="RETURN":GOSU
B C2:? " TO CONTINUE"
6800 RETURN
11000 REM
11005 PLOT X,Y:X=X-C1:LOCATE X,Y,Z
11020 IF Z=C0 THEN GOSUB FILL
11030 X=X+C1
11040 Y=Y-C1:LOCATE X,Y,Z
11050 IF Z=C0 THEN GOSUB FILL
11060 Y=Y+C1
11070 X=X+C1:LOCATE X,Y,Z
11080 IF Z=C0 THEN GOSUB FILL
11090 X=X-C1
11100 Y=Y+C1:LOCATE X,Y,Z
11110 IF Z=C0 THEN GOSUB FILL
11120 Y=Y-C1
11130 RETURN
12000 TRAP 40000:GOTO PEEK(186)+PEEK(1
87)*256+20
```

From Page 77

at a location in L$, and puts the next 10 memory units into the screen memory. It skips to the next line and then skips 120 locations in L$ to get the information for the next line of information. This is done by a machine language subroutine so that it is very quick. The subroutine is shown in Figure 3.

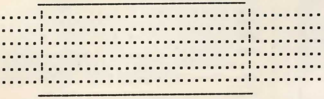
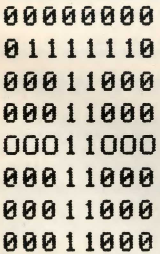


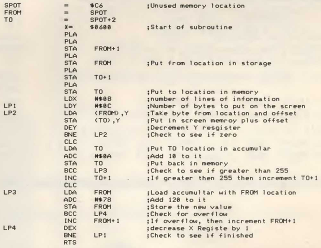
Fig. 1

Let me explain the program quickly. Lines 1-18 initialize several variables that are used in the program. The use of variables rather than numerical constants speeds up the program. The string variables are dimensioned. Subroutine 900 puts values into XSTEP and YSTEP which will be used to vary the joystick movements. This is much faster than a series of IF tests to determine the direction the joystick is pointed. Line 15 puts the value of 0 into the string L$ to make the memory blank. Line 18 determines the address of the string variable L$.



```
00000000
01111110
00011000
00011000
00011000
00011000
00011000
00011000
```

Fig. 2



```
SPOT            =      $C6           ;Unused memory location
FROM            =      SPOT
TO              =      SPOT+2
                *=     $0600         ;Start of subroutine
                PLA
                PLA
                STA     FROM+1
                PLA
                STA     FROM         ;Put from location in storage
                PLA
                STA     TO+1
                PLA
                STA     TO           ;Put to location in memory
                LDX     #$08         ;number of lines of information
LP1             LDY     #$0C         ;Number of bytes to put on the screen
LP2             LDA     (FROM),Y     ;Take byte from location and offset
                STA     (TO),Y       ;Put in screen memroy plus offset
                DEY                  ;Decrement Y resgister
                BNE     LP2          ;Check to see if zero
                CLC
                LDA     TO           ;Put TO location in accumular
                ADC     #$0A         ;Add 10 to it
                STA     TO           ;Put back in memory
                BCC     LP3          ;Check to see if greater than 255
                INC     TO+1         ;If greater then 255 then increment TO+1
                CLC
LP3             LDA     FROM         ;Load accumultar with FROM location
                ADC     #$78         ;Add 120 to it
                STA     FROM         ;Store the new value
                BCC     LP4          ;Check for overflow
                INC     FROM+1       ;If overflow, then increment FROM+1
LP4             DEX                  ;decrease X Registe by 1
                BNE     LP1          ;Check to see if finished
                RTS
```

Fig. 3

Lines 30-120 are used to determine every message. Line 20 and 30 INPUT the message to be put on the banner. Subroutine 1000-1040 determines the size of the letters and if fat or normal size letters are desired.

The major difference between large and small letters is the Graphics mode used, with Graphics 3 using large blocks and Graphics 4 using small blocks with the option of either fat or normal letters. The reason for this has to do with the packing of information in the various Graphics modes.

Graphics 3 is a four-color mode and every byte in memory is made up for four two bits. Each two bits specifies the color register to be used for the color of that square on the screen.

Graphics 4 is a single color mode. Each byte is made up of eight 1 bit portions that specify if the square is to be lit or not. If the same information is used for Graphics 3 and Graphics 4, then the letters will be fat. If the four bites per byte information is changed to eight bits per byte, then the letters will be normal size. Constants are initialized for each situation.

Line 45 puts a statement on the screen and then goes to subroutine 3000. This subroutine reads the data for the machine language subroutine shown in Figure 2 and puts it into hex $0600, a safe place to store subroutines. Lines 50-120 dissect the message into its components. Line 60 determines the letter and subroutine LTT finds the offset from the start of the character memory at 57344.

The exact location of the start of the character is calculated in line 70 and the next eight memory locations are used, one at a time. Line 80 determines the offset from CHARLOC and checks to see what Graphics is used. Y is the value of the location in L$ the next character is to be placed. This value is POKEd into the address plus offset since this is much quicker than using the string transformations. The color of the screen is changed, and a sound is generated, to show that something is happening.

Lines 127-130 determine several variables and the start of the screen memory. Lines 136-190 is the loop controlling the movement. Line 136 determines the value of the joystick and transforms it through XSTEP into a horizontal direction. Lines 138 and 139 determine if the direction is out of bounds. Lines 140-143 do the same thing for the vertical direction. Line 155 is the subroutine that takes a location in the string variable L$ ((LPT + I) and puts it into screen memory at location (T + 10*J). Lines 160-170 determine if the color of the message is to be changed. If the trigger is pressed, then the color will be changed. Line 190 starts everything over again.

This program is a demonstration of some of the capability of the Atari system. By the use of a string variable it is possible to pack much information into a small amount of memory. In Graphics 3, the whole screen can be put into 400 bytes. Ten screens of information can be put into only 4000 bytes. Using a machine language subroutine like the one in this program, the data can scrolled so it appears instantly when it is needed.

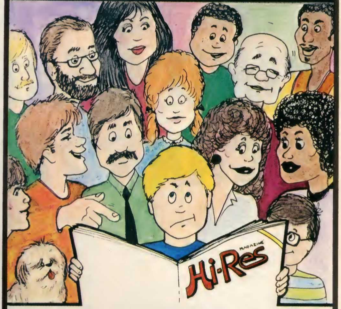
Joystick Controlled Banner

```
0 REM JOYSTICK BANNER C 1982 BY SOL GU
BER
1 GRAPHICS 0
5 C1=1:C16=16:C4=4:C0=0:C7=7:C8=8
6 LTT=600:C31=31:C32=32:C128=128:C96=9
6:C64=64:C127=127:C12=C4+C7+C1
7 C2=2:C120=120:PKR=708:C60=C64-C4:CHA
RLOC=57344
10 DIM L$(960),A$(110),O$(1)
12 DIM XSTEP(20),YS(20)
14 GOSUB 900
15 L$(C1)=CHR$(0):L$(960)=CHR$(0):L$(C
2)=L$
18 LPT=ADR(L$)+C4+C1
20 ? "WHAT IS YOUR MESSAGE"
30 INPUT A$:N=LEN(A$)
43 GOSUB 1000
45 GRAPHICS C2+C16:? #6;"  INITIALIZIN
G":GOSUB 3000
50 FOR I=C1 TO N
60 O$=A$(I,I):GOSUB LTT
70 I3=CHARLOC+X*C8:FOR J=C0 TO C7:M=PE
EK(I3+J)
75 M4=C2:IF FLG=0 THEN M4=C1
80 GOSUB 800:Y=J*120+(I-C1)*M4+C1
82 POKE Y+LPT+C1,FC*T:IF FLG<>C0 THEN
GOSUB 800:POKE Y+LPT,T*FC
85 POKE PKR,T+C16
115 SOUND 1,T+C16,10,8
120 NEXT J:NEXT I
125 SOUND 1,0,0,0
127 LPT=ADR(L$):R=C16-C4-C4-C4:POKE PK
R,R
128 GRAPHICS RG+C16
129 T=PEEK(88)+256*PEEK(89)-C1
130 I=3:J=3
136 S=STICK(0):I=I-XSTEP(S):R=PEEK(PKR
)
138 IF I<C0 THEN I=I+XSTEP(S)
139 IF I>C2*N+C4 THEN I=I+XSTEP(S)
140 J=J+YS(S):IF J<C0 THEN J=J-YS(S)
142 IF RG=3 AND J>15 THEN J=J-YS(S)
143 IF RG=4 AND J>34 THEN J=J-YS(S)
155 A=USR(1536,LPT+I,T+(C8+C2)*J)
160 IF STRIG(C0)=C0 THEN R=R+C16
165 IF R>255 THEN R=R-255
170 POKE PKR,R
190 GOTO 136
500 STOP
600 X=ASC(O$):IF X>C127 THEN X=X-C128
610 IF X>C31 AND X<C96 THEN X=X-C32:RE
TURN
620 IF X<C32 THEN X=X+C64
630 RETURN
800 IF FLG=0 THEN T=M:RETURN
810 R1=C1:T=C0:FOR K=C1 TO C4
820 M=M/C2:IF M<>INT(M) THEN T=T+R1
830 M=INT(M):R1=R1*C4:NEXT K:RETURN
900 XSTEP(5)=1:XSTEP(6)=1:XSTEP(7)=1:X
STEP(9)=-1:XSTEP(10)=-1:XSTEP(11)=-1
910 XSTEP(13)=0:XSTEP(14)=0:XSTEP(15)=
0
920 YS(5)=1:YS(6)=-1:YS(7)=0
930 YS(9)=1:YS(10)=-1:YS(11)=0:YS(13)=
1
940 YS(14)=-1:YS(15)=0
950 RETURN
1000 ? " LARGE OR SMALL LETTER (L/S)"
1003 FLG=C1
1005 INPUT O$:IF O$="L" THEN RG=3:FC=1
:RETURN
1010 IF O$="S" THEN RG=4:FC=3
1020 ? "FAT OR NORMAL LETTERS (F/N)":I
NPUT O$
1030 IF O$="F" THEN RETURN
1040 FLG=0:FC=1:RETURN
3000 FOR I=1536 TO 1600:READ X:POKE I,
X:SOUND 1,X,10,10:NEXT I
3005 SOUND 1,0,0,0
3010 DATA 104,104,133,199,104,133,198,
104,133
3020 DATA 201,104,133,200,162,8,160,12
3030 DATA 177,198,145,200,136,208,249,
24
3040 DATA 165,200,105,10,133,200,144,3
3050 DATA 230,201,24,165,198,105,120,1
33,198
3060 DATA 144,3,230,199,24,202,208
3070 DATA 221,96,0,0,0,0,0,0,0,0
3080 DATA 202,208,215,96,0,0,0,0,0,,24
3090 RETURN
```

"I've heard of computer dating, but..."

Answers to Crossword

	P					R													
	R			A	N	A	L	O	G									I	
P	I	L	O	T		M										P	R	O	M
	N			A				P								I			
	T			S	E	C	T	O	R							X	I	O	
	E		I	C				K		D			H			E			
O	R			I			P	E	R	I	P	H	E	R	A	L	S		
			B	I	T			Y		S			X					F	
	D		Y		E		N			K							R	O	M
P	O	R	T		R		A	S	S	E	M	B	L	E	R			R	
	S		E		M		K			T						D	A	T	A
					I			I	N	T	E	R	F	A	C	E		H	
		J			N	O	P			E		E		N		B			T
		O			A							M		D		U			R
	N	Y	B	B	L	E		S								G	T	I	A
		S				N		A					B			G			C
		T				T		V					A	N	T	I	C		K
		I			S	E	L	E	C	T			S			N			
		C				R							I			G	O	T	O
		K	E	Y						P	A	S	C	A	L				



Truehart of the CIO

KIND HEARTS & CARROT JUICE

Episode 2

by Léo Laporte

When we last joined our hero, Jack Truehart, he was on his way to the dinner table — as the main course.

Fleshy meat, strong and sweet,
Tonight the POKEYs devour a treat!

The song was banging about inside Jack's head like a string of Chinese firecrackers. This was the tightest spot he had been in since his first assignment for the CIO — gunner duty on a Zylon Raider. Those were the days, nothing to do but play games. And now, after years of fighting alien invaders and system glitches, to end it all trussed up like a side of beef on his way to the kitchens of the POKEY people. Jack felt like a sap.

This had to be the doing of that demented dastard who was stealing clock cycles from the System. Jack wondered if they'd meet before the end.

"Hey, you, POKEY," he shouted.

"What do you want, prisoner?" the POKEY holding Truehart's left foot grimaced menacingly.

"I want to meet the Big Guy. I'm on very important business from the CIO, and I think he'll want to see me before you do anything rash."

Another POKEY to Jack's left kept hopping agitatedly from one foot to another. "Do y'hear that, boys? Important business from the CIO! We've got a real high muckamuck here."

A wave of high-pitched giggles washed over Jack's head. The POKEYs kept marching down the carbon black corridor, Jack's bundled form on their shoulders. Time for some fast lip wagging, he thought to himself.

"Really, guys, this is important. We've learned that Atari is planning to replace the POKEY chip on the Mother Board with a new chip from Commodore."

It wasn't true, of course, but Jack figured it would have an effect. And it did. The POKEYs dropped him on his head and began squealing.

"Nerts! They can't do that! We're the most important chip on the Mother Board. Who'll do their I/O? Who'll make their music?"

Jack almost felt sorry for them, but he couldn't stop now. "I hear they're going to move in SID."

Now there are some things the POKEY people hate worse than SID. Baths, for one. Brussell sprouts, for another. But there was no more potent voodoo Jack Truehart could have conjured up just at this moment, than the name of SID, the hated Commodore 64 sound chip, and POKEYs' archrival. His little fib unleashed a maelstrom. It was all he could do to roll out of the way of the stomping POKEY feet. He tried to cover his ears, but his arms were tied down too securely.

"Never, we'll melt the System down first!" the chief constable POKEY screamed, while frothing at the mouth and wildly swinging his nightstick. "No SID will ever be soldered into our slot. Never!"

So violent was their rage, they forgot all about Jack. Now to escape before the POKEYs shrieks caused permanent hearing loss.

Jack began to work his way down

Illustration by Daijo Akutagawa

the corridor, moving like an inchworm, a bit at a time. He had gotten about 10 feet away when his progress was stopped by a pair of legs that definitely were not POKEY shaped. And they went all the way up. A woman!

Jack would have stood and taken his hat off, but it was all he could do, under the circumstances, to keep from tumbling over backwards.

"Hello," she said. Jack's normally cast iron heart skipped a beat.

"Hello, yourself. Jack Truehart, CIO. Get me out of these ropes, willya?"

"Well, I don't know. Did the POKEYs tie you up?" She was grinning, but Jack wasn't sure why.

"Uh, yes, but I'm here on very important business. I've got to see your chief."

"I see." And she walked off toward the still swarming mass of POKEYs.

Jack watched her go with an appreciative eye, and then began worming off again down the hall. He didn't get much farther before everything got very quiet.

"Hot solder, what now?" he thought. He could hear her speaking to the POKEYs, but he couldn't make out the words. Then he heard footsteps as the POKEYs charged down the hall after him. He tried to roll away, but he could no more escape them than a tenpin could escape 100 bowling balls. They came rumbling down all around him. And they weren't happy.

"Tell us stories, willya? Why I ought to bean 'ya with my billy club." The constable was jumping up and down on his little furry feet.

"Come on now, boys, untie Mr. Truehart." It was the lady. The POKEYs grumbled, but they began to loosen the knots on Jack's bonds.

"The old Truehart charm works every time," Jack thought.

Before you could say Beginner's All-purpose Symbolic Instruction Code, Jack was back on his feet. He noticed once again how very nice looking the young woman was. She was Jack's height, with blonde hair like his, and green eyes. She was dressed head to toe in an opalescent pink number hot enough to scald a supernova.

"Did you really tell them that they were going to be replaced with SID?"

Continued on Page 91

SPACE KNIGHTS

Your Passport to the Future

First of a Series

by David Heller

The forty young graduates stood at attention in front of a deserted wooden reviewing stand.

Light wind played across two blue and gold flags planted on either side of a small wooden speaker's rostrum. The rostrum appeared lonely, standing naked in front of the reviewing stand, as if it were waiting for a speaker to fulfill its destiny.

Jake's legs were fatigued, and he shifted his weight imperceptibly from one foot to the other. "Where are they?" he thought. As if on cue, an aged, bent man wearing the blue and gold of an admiral, appeared from behind the stand and walked slowly to the rostrum. He stepped up and blew into the microphone. Two gigantic speakers mounted on poles at the top of the stand squealed and screeched. The admiral reached below the rostrum, then brought his hand up and tapped the microphone. "Testing. Testing, one two three. Testing." He paused to adjust the microphone. "Gentlemen," he wheezed, "can you all hear me?" His voice bounced and echoed off the deserted hangars at the far side of the field.

They were all stunned. His words rang in their ears. No one answered. "Well, very good, very good." He cleared his throat. "I'm glad that you made it." His words were slurred, as if he were on medication, or very old and tired.

The admiral continued, and a tremor of pride entered his voice as he praised their accomplishments and explained their mission. "You young people are our hope for the future. You are the best this society has to offer. You are making a great sacrifice. I praise you, I salute you, and in many ways, I envy you." His voice became loud and strong. "Your individual missions will vary. Some of you will resupply our settlements at the fringes of the galaxy. Some of you will engage in battle, and some of you will be the fathers of new and better worlds.

"All of you" — he paused to clear his throat and lower his voice — "every last one of you, will travel forward in time, never returning to this special place in the universe.

"We are all time travelers." His voice was calm, patient and filled with concern. "Most of us wander through time as I have done, in the company of our fellow beings. This afternoon you will leave us to awaken hundreds of years in the future...our future."

The little admiral walked toward the line of cadets. He paused for a moment to say something to the sargeant, then walked slowly up and down the ranks shaking each man's hand. He stopped in front of Jake and smiled. His eyes seemed alive and young. "Well, young man, I've heard a lot about you." Jake's heart beat faster. "How could he know about me?" he thought. They shook hands, and before Jake had a chance to speak the wizened admiral said, "Godspeed, son," and moved down the line.

Jake lay in a foam-filled container, looking up into the faces of three hovering attendants. Their noses and mouths were covered by white surgical masks. An intense white light scorched his eyes. A strange numbness crept up his legs. His eyelids became heavy, then closed.

Gammalon Encounter

Millions of cilia-like fibers flowed over his body. Each pass lifted more of the gelled silicone casing from his skin. A blast of high-energy ion particles bathed him in blue-white light. The small enclosure sparkled with blue flashes of energy, smelling of ozone. Gradually, semi-darkness returned. A dull humming sound filled the room, and Jake rose slowly from the stainless steel slab like a dead man rising from his coffin.

His bare feet touched the cold metal deck. He flinched, just beginning to realize where he was. He stood, rubbed his eyes with the backs of his hands, nodded to clear his head, and surveyed the room. "Oh, no. Not a transport!"

Jake arrived at the Situation Room in a daze, assaulted by pulsing bright lights and the repetitive scream of a high-pitched claxon. He stepped from the cylindrical elevator hatch and entered.

A small illuminated button pulsed bright red in the center of the control console. He raced forward and slammed his hand against the pannel; the lights stopped flashing and the claxon ceased its wail.

"It's about time." A pleasant bari-

' This afternoon you will leave us to awaken hundreds of years in the future...our future. '

ton voice startled him. "I've been waiting for almost 30 minutes. You know, we've got an emergency on our hands. Sit down, Jake. We'll talk it over."

Jake collapsed into one of the two contoured chairs facing the console. "OK, what gives? What model are you anyhow?"

"I'm CLIDE," the voice answered. "That's short for Computer Linguistic Information and Data Embryo. The embryonic part means that I'm a bit underdeveloped. However, I have been programmed to learn. As I work with you, I'm sure to pick up all sorts of nifty data."

Jake was disgusted. "Are you telling me that you're a MOD-I?"

"Correct, Jake. A MOD-I it is. One of the best units ever developed by R&D."

"Listen, CLIDE," Jake said, "I don't want to hurt your feelings, if you were programmed to have any, but the MOD-I was outmoded before I was born." He shook his head in disgust. "Star Fleet discontinued their use more than two decades ago. How did I ever get stuck with you?"

"It could have been that time you caught Plasma-Ray instructor Squigglie sneaking onto the base with the Fem-Android, Nurse Elems." CLIDE suggested innocently.

"How did you know about that?"

"It's all in my data banks. I've got a complete dossier on you that goes back to your birth. It's always good to know who you're working with."

Jake tapped his fingers against the panel. "OK, what class vessel is this, and what's our situation?"

"No problem," CLIDE began. "You and your assistant, 7EN4, who is currently being revived, are aboard Eggtube Transport 2DELTA76. Your mission was to have been the safe transport and off-loading of supplies on Nystrum IV. However, as you may have surmised, we seem to have en-

About Space Knights

Welcome to the first *Space Knights* episode. You're in for a real treat.

In this, and future issues of *Hi-Res Magazine*, you'll be propelled on a fantastic voyage to the outer limits of the universe. But, before you blast-off with Jake and Lisa, I'd like to tell you a little about the *Space Knights* concept, how I came to write it, and how you can add to your excitement.

Two years ago the idea for *Space Knights* popped into my head as I was playing a computer game on my Atari.

I thought, "I'll write a science fiction adventure novel, team up with an illustrator and a programmer, combine an illustrated novel with computer games, and offer the public the first total package of its kind." By relating arcade style games directly to an illustrated story, the reader relives the adventures of the story's two protagonists and becomes a part of the adventure.

I wrote a sample chapter, teamed up with a brilliant 17-year-old programmer, Bob Kurcina, and started pounding on publishers' doors. I was startled by the positive response. Atari Corporation was excited and agreed to provide technical assistance. Less than a week after mailing my first proposals, Reston Publishing Company sent me a contract!

My *Space Knights* idea had become a project, and once the ball started rolling Frank Cirocco came on board. Frank is a local fantasy artist who started his career in New York working on *Superman* comics. Since then he's become a world renowned illustrator. We were fortunate to have Frank on our team, and I'm sure you'll enjoy his imaginative art.

The computer games took shape as the writing progressed. I'd write a segment, then brain-storm game ideas with young Bob Kurcina. During the development phase, Bob's sister, his two younger brothers, my son and a whole slew of neighborhood youngsters added their input. They told us what they'd like to see in each game, evaluated the prototype versions, and had fun flying spacecraft through the Corridor of Time, landing on Weomby and fighting their way through the War Room.

We started with 12 games, but only nine met our young critics' high gaming standards. These nine games are now included on a disk that's sold with the illustrated novel.

Turning my intangible ideas into reality while forging into uncharted territory made writing and developing *Space Knights* an exciting adventure. I'm sure you'll experience lots of excitement and adventure as you explore uncharted worlds with my two main characters, Jake and Lisa. So, buckle your seat belts, hold on tight, and enjoy your journey through the world of *Space Knights!*

Footnote: Adapted from *Space Knights* by David Heller and Robert Kurcina, copyright © 1983, published with permission of Reston Publishing Company, Inc., Reston, VA 22090. All rights reserved.

'It's just the way I'm programmed. How can I help it? It's just the way I am.'

countered a little problem." CLIDE paused, as if afraid to continue.

"Well, my long-range scan has picked up something unusual in quadrant A23, now exactly 27 Earth minutes distant. The signals are faint, but it's beginning to look like Gammalon pirates."

"So, you're not sure yet." His face turned red. "If you weren't sure there was a threat, why did you wake me up? Why didn't you just take evasive action?"

"Jake, Jake, Jake." CLIDE sounded as if he were lecturing a truant schoolboy. "Article 71Ø,99 specifically states that a human will be present in the Situation Room in the event of any potential danger. Jake, it's just the way I'm programmed. How can I help it? It's just the way I am."

"All right, just give me the armament list and status."

This was a command that CLIDE was well prepared for. He proudly listed the freighter's readiness status. "Two 2ØØØ series photon rays forward, two 1ØØØ series photon rays aft, and a MOD SLP/65 evasive electronic countermeasures set. In addition to these armaments, the Eggtube transport is capable of sublight hyperwarp. All equipment in green status."

"Is that all? I just pray that we're not really going to have to face Gammalons. CLIDE, this is my first mission. I don't want it to be my last."

* * *

Brazt was crunched into a tight ball, squeezed behind a jumble of steaming pipes and wires. His flesh moved, tightened, then convulsed. The contractions were more regular, sharper, excruciatingly painful.

His breathing became faster. His twin hearts pounded a syncopated rhythm, pushing green life-liquid in powerful surges through his contorted body. The final moult was always the most painful, the most dangerous. Brazt's thin body shuddered.

He was barely conscious. He fought to retain his senses. If Clazon found him defenseless, he was dead. And he was so close — so close to total control and absolute power.

He hissed the name Clazon. Hate would keep him awake. The ugly, misshapen image of Clazon twisted

‘Well, Lisa Rome, welcome aboard. I just hope you know your stuff.’

and turned in his fevered brain. He looked contemptuously at Clazon's form from all angles. Gaunt, thin arms and legs covered by translucent blue-tinged flesh, a soft stomach, and cruel gray multicelled eyes. Then his mind's eye focused on Clazon's power. Three slim appendages at the end of his thin wrist. The smooth fingers were broad at their tips and, if Clazon's victim were lucky, white beams of vaporizing instant death spewed from their ends. If Clazon was in an ugly mood, and that was often, the poor underling would squirm and scream at Clazon's feet, engulfed in burning pain.

Brazt shuddered, remembering his experience under the beams. Soon, when Brazt's moult was finished — if they didn't find him — Clazon would feel the pain of his fingers. "They will cringe beneath the sting of my power." His thoughts were interrupted by a wailing alarm. "We've encountered an alien vessel. I'll be missed on the bridge." He willed the moulting to go faster.

* * *

"Don't worry." Jake spun around. "With me on board, we'll make it through unscathed." The voice sounded as confident, in a high, youthful, idealistic way, as his old Nursery Group teacher organizing games.

The girl stood straight, as if on parade, hair and uniform unblemished and unwrinkled. She was tall, almost as tall as he, and for some reason his heart sunk to his boots.

"Who, or rather what, are you?"

"Name's Lisa Rome." She walked up to Jake, formally saluted, and sat down stiffly next to him.

"Well, Lisa Rome, welcome aboard. I just hope you know your stuff. CLIDE's been telling me we're in for a little trouble." Jake kept it formal; he felt that this might be a tedious relationship. And, as if to prove him right, she replied smugly, "Don't worry, ensign, I know my stuff. I graduated head of my class."

"CLIDE." Jake purposely turned away from the girl. "Have you made positive contact yet?"

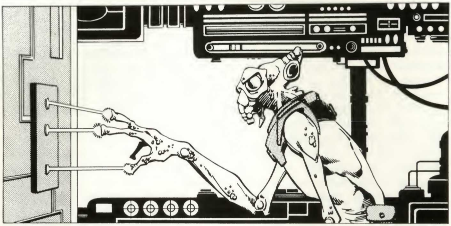
"No, Jake, I should have positive I.D. within 10 minutes. Current data reinforces my initial report. Gammalon pirates."

"OK, CLIDE, just let me know the minute you have the situation firmed up." Jake turned to Lisa, briefed her, then inquired, "You said you graduated head of your class. What's your specialty?"

"Nice of you to ask, ensign."

"This is going to be a long voyage ...I hope. Try Jake, OK?"

"Sure, Jake, anything you say...up to a point. Electronic warfare and countermeasures is my strong suit, but I studied gunnery as well as medicine, with some omni-biology courses."



"I hope we won't need your medical skills in the near future. But, if these are Gammalons, I'll be counting on your E.W. expertise. I understand their electronics are fairly sophisticated."

"You understand correctly," she said. "Their radar, if you can really call it that, operates in the violet spectrum of the light band. They'll use deceptive jamming as well as pulse-node frequency-agility tactics. Very difficult to counter. What type of equipment is this scow fitted with?" "CLIDE tells me our great ship has a MOD SLP/65 set onboard. What's your assessment?"

"It's not going to be easy. But it's not impossible." She shrugged her shoulders. "I'll do my best."

"Right. You'd better get acquainted with the equipment while I check system readiness, then I'll try to take a look at our adversaries on long-range scan."

* * *

"Ahggh...the pain." It spread from Brazt's hooved toes, up his calves, and sliced into his midsection like an upward-thrusting dagger. He doubled over. His skin turned a darker blue as air whooshed out of his collapsing lung. "Got to hold on," he gasped. He felt thousands of needles puncturing his stretched skin. He heard a tearing sound as the raw skin at the front of his head split open. The dry outer skin layer fell back in a straight line that raced up the front of his legs. He held back a scream as the two openings met at his groin and continued racing, as one large fissure, up his stomach and the center of his chest. His body jerked involuntarily straight.

He felt as if he were being flayed alive. The pain reached his hands. They began to shake. Smoke poured from his fingertips as the outer layer of skin began to dissolve. He looked in amazement as his fingers began to change in shape from thinly pointed appendages to fat, mushroom-tipped weapons.

Clazon's voice screamed at him from a wall-mounted loudspeaker. "Bŕạzt!" Clazon spat out his name, making it sound foul and repugnant. "Brazt. You will pay! We have found a fat, juicy Earth ship. You will live to watch our carnage. Then, Brazt, we will make a carnage of you! Desertion in the face of the enemy. I'm going to enjoy watching you suffer!"

* * *

Jake punched a series of numbers into the command console, and the view-screen listed the battle hardware. A green light pulsed next to each red line of text, indicating all systems, GO and in full battle readiness. He punched another button, and the screen showed a long-range view of their sector and blurred view of quadrant A23 at the top of the screen.

"CLIDE. Let's see a readout of distance to target displayed on the righthand corner of the screen."

Tension mounted as they approached the encounter, affecting even CLIDE's long-winded programming. CLIDE went into action without answering, and a string of numbers flowed across the screen.

"Well done, CLIDE. Now, I'd like you to put up our shields, start with low energy, and if we make positive contact, pull them up to full strength."

"Sorry, Jake, but this craft is not equipped with shields."

"Oh, brother. Did you hear that Lisa? No shields! We won't stand a chance."

She thought for a moment. "Maybe we can outrun them."

"Right." Jake turned his attention to the computer, impressed but irritated.

"CLIDE, do you have positive contact yet? We're running out of time. We'll be reaching point of no return in less than a minute."

"Sorry, sir." The computer's use of "sir" worried Jake. "But," CLIDE continued, "even though all the indicators point to Gammalon pirates, I won't have positive I.D. for another five minutes."

"By then it will be too late!" Jake exploded. "If they are Gammalons, we'll be forced to fight."

"Sorry, sir." CLIDE replied.

* * *

Brazt felt a shudder flow through the Dreadnaught as the battledrive engaged. The pain eased, and he sat up and looked down at himself. Dead skin flaked away from his body as he rose, exposing a smooth, radiant, translucent blue-sheened covering. He flexed his cup-tipped fingers, and latent energy snapped and buzzed, tugging to be released.

Brazt held his right arm forward, and pointed his fingers toward the double-thick hatch. He willed the energy to leap, and three beams of solar intense light shot from his fingertips slicing through the thick metal like a hot knife through butter. Smoke hissed from three perfectly round holes in the hatch's center.

Brazt leaned back and savored his power. "Soon, this armada will be mine, and Clazon will entertain me! Entertain his Lord in a method befitting my rank and status. Today ends his reign of insolence. Today marks the ascent of Brazton, the first Lord of Gammala!"

* * *

Lisa and Jake stared at the upper portion of the viewscreen waiting for the blur to change into a recognizable shape. Jake nervously tapped a pencil on the instrument panel. Lisa waited in readiness, her fingers poised before the knobs and switched off the E.W. console.

CLIDE excitedly broke the silence. "Positive I.D., Gammalons! Three of them. Two Regnaught Fighters, and one Dreadnaught. Relative bearing 247 degrees, range 5ØØ kilometers and closing. Intercept at point ALPHA97 at 2145 hours."

"Show it, CLIDE! Put it up on the screen!" Jake was frantic.

The screen blinked, then showed a representation of three Gammalon vessels. Superimposed red lines moved slowly toward a point in the center to indicate calculated point of contact.

Suddenly, the Gammalon vessels vanished, and the three red lines swept aimlessly across the empty screen.

Lisa's fingers scurried across her instrument panel. "Deceptive jamming, Jake. I'm going to counter."

Jake gulped. To regain control of the situation, and himself, he said, "CLIDE, based on your last fix, when can we expect contact?"

"Still looks like 2145, sir. That will be in precisely 5.72 minutes."

"Did you hear that, Lisa? If we can't see them, we can't fight them. We'll be sitting ducks!"

She didn't answer. Her entire being was focused on the small screen and the buttons of her E.W. console.

From Page 85

she asked.

"Of course," Jack replied.

"Liar, liar, from Sigma Tyre!" It was one of the POKEYs and he was hopping mad. Literally. He hopped right onto one of Jack's toes.

"Ow! Get offa there!"

"Now, Twilly, be nice to Mr. Truehart. I'm sure he was indulging in a harmless joke. Weren't you, Mr. Truehart?"

"Well, confidentially, I'm here on a much more important matter. If I could just see Mr. Big, I'm sure I could explain everything."

"Mr. Big?" Linda smiled. The POKEYs giggled. "We have no Mr. Big. There's just the POKEYs and me."

"And who are you?" Jack asked.

"I am Linda Darling. And this is Twillnap, he's our security chief."

The POKEY bowed. "Twilly, this is the famous Jack Truehart of the CIO."

"Pleased to make your acquaintance, JT." The POKEY stuck out a furry hand.

"Pleased to...why you flea flecked fuzz ball, you were about to eat me for dinner a minute ago."

Linda tried to look stern. "Twilly, you didn't tell Mr. Truehart you were going to eat him, did you?"

The POKEY feigned a sudden interest in the buckle on his sash.

"Mr. Truehart, my apologies." She hardly could keep from laughing. "My friends have a somewhat childish sense of humor — particularly when it comes to strangers.

"Actually, the POKEYs are strict vegetarians. They live on carrot juice and guava jelly. I promise you, agents from the CIO are not on their menu."

The POKEYs tittered.

"Humph." Jack was feeling like a sap again. "Be that as it may, I'm here on official business, and someone will have to answer my questions."

"We would be very happy to. Perhaps you would like to join us for dinner — strictly vegetarian, of course? We could answer your questions then."

The invitation was welcome. It had been many hours since Jack's last meal. He wondered if he could get some Chinese food.

The POKEYs, his captors moments before, transformed into an honor guard to escort Jack and Linda into the heart of POKEY. As they walked, Linda told Jack about their duties inside the chip. Her remarks were punctuated by occasional high-pitched exclamations of agreement from the POKEYs.

Illustration by Daijo Akutagawa

"POKEY is one of the three original chips on the Mother Board. We were designed to relieve the CPU of some of the more mundane tasks within the System. Those giant halls you saw when you came in are the I/O ports. In one chamber we pull in the signals from the disk drives or the keyboard or whatever else is hooked onto the System. The data comes in, we parcel it up, and send it off down the data bus for processing.

"Whenever data is to be sent outside the System, the POKEYs prepare it and pass it through the output chamber to the printer or disk drive or whatever. It's very hard work, we handle thousands of bits every second."

"You make a devilish lot of noise while you're doing it, too," Jack said.

"You noticed," Linda agreed. The POKEYs like to sing and make noise. It was built into them when the Mother Board was designed. You can see over here the choir halls where the POKEYs generate the music and sound effects for the whole System. There are four halls, one per channel, and each of them can make a different sound."

Jack was awestruck by the size of the halls and the power of the sound that came from them. He almost could see the music as it pulsed through the air around him.

In fact, he became so engrossed in the multi-colored strains of a Bach Cantata, that he nearly ran into a plate-glass window. "Oof."

His attention switched from the music to the view through the window. He saw a cavern seven-stories tall, filled with hundreds of thousands of POKEYs, bouncing against the walls, and each other, with abandon. Every so often one of the POKEYs would randomly bounce out through a hole in the cave's roof.

"Great gamma guns! What's going on in there?" Jack asked.

"This is the Random Number Generating Room. Here, we're constantly generating random numbers to be used by programs. The numbers we make here also are used by

Illustration by Daijo Akutagawa

the distortion counter in the music halls. Each POKEY inside is assigned a number. Every few thousandths of a second, one of them flies out into the room above, where his number is recorded for use by the OS. The POKEY is then sent down that chute over there and back into the cavern. It's really kind of fun. You should try it."

"Uh, yeah." Jack was having a hard time taking it all in. But he managed to swallow his confusion long enough to say, "To tell you the truth, I'm more interested in the interrupt handling."

Jack and OS had guessed that whoever was using the System was taking control of the interrupts through POKEY. By issuing interrupt commands, someone with a penchant for villainy could take over the System for short periods of time, without permission from CIO or OS.

Linda looked at him strangely. "Why do you ask about the interrupts?"

Jack wondered, could he trust her? There was only one way to tell. He told her the story and closely watched her reaction. The tale had its effect. The POKEYs began to shiver and Linda's brow creased with worry.

"I had a feeling that's why you were here," she said. "I'd better tell you what I told your friend, McGinty."

So! McGinty had been here. And whatever he had learned must have led to his electrifying mishap in the halls of the data bus.

The group entered one of the vast dining halls, now empty but for a few food-prep POKEYs who were cleaning up. As Linda and Jack sat down, large plates of green jelly and carafes of orangy liquid materialized on the table. The POKEYs bellied up to the food and began eating with gusto. Jack grimaced. Didn't look like sweet and sour shrimp to him.

As the POKEYs ate, Linda told her story.

"It's been happening for several months now," she began. "Every night, just about this time, an interrupt command comes through. We are to stop everything and wait for I/O. Usually a large packet of information will go out through the serial port. Sometimes more information comes in. It goes very quickly, and after it's over, it's as if nothing had ever happened.

"At first we assumed that these were authorized System calls. But recently we have begun to think otherwise. When your man McGinty came here earlier this evening, he asked us about the I/O. He told us he was from the CIO and that he was investigating illegal accesses. That's when we began to worry. I decided to have Twilly patrol the I/O hall after hours. That's where he caught you."

Twilly wiped at his green-stained whiskers and nodded vigorously. "At first, we thought you were the intruder."

"I'd like to see this interrupt. Is there somewhere we can watch?" Jack asked.

"Well, we could watch from the catwalk on the serial port, but it's dangerous."

"Danger was my mother's maiden name. Let's go."

Jack jumped up and took Linda by the hand. It was warm to his touch, and so much friendlier than the printed circuits he was used to. Jack had no doubt he could trust her, now. She was too beautiful to be lying.

Jack and Linda left the dining hall. The POKEYs were too busy guzzling guava jelly to notice.

They turned south and began walking down a forest green hall lined with oddly-shaped green and yellow bushes. Every other bush had a snoozing POKEY underneath it.

"Shh," Linda whispered, "they're sleeping. When the interrupt request comes we'll have to be very careful. Whoever's doing it takes over all the POKEYs in the I/O chamber. We will be terribly outnumbered."

They had come to a small gap carved into the silicon wall. Upon climbing through, Jack and Linda found themselves on a catwalk bolted to the mirror smooth walls of the serial bus. To their left stretched the output port, a dark pipeline to the outside world. To their right lay the I/O hall, carpeted with a thousand POKEYs, resting now after their hard day's labor. The overhead lights had been turned down low; the hall glowed a mellow gold.

"Be very careful, Jack, we're at the mouth of the output line. If you were to fall off here you would be sucked out into the disk drive. Every once in a while, we lose one of the POKEYs that way. When the data bits come through, press your back against the wall and they'll glide right by."

"How long will we have to wait?" Jack could hear the faint sound of snores coming from the POKEYs hundreds of feet below. The evening was cool and very calm. He was beginning to feel a bit more relaxed.

"Here it comes now!"

There was an electric buzzing in the air. The sleeping POKEYs awoke. Their brown eyes grew as big as dinner plates. From the back of the room came a whine, then a red sphere crackling with static electricity rolled into the center of the chamber.

"That's the interrupt. Now he'll take control," Linda whispered.

In the silence, Jack could feel Linda's heart pounding as she stood next to him.

A disembodied voice came from the red sphere, crackling as it spoke. It sounded a little like Rudy Vallee.

"Good evening I/O POKEYs. This is Hi-Top Harry. Prepare to receive data."

The POKEYs scrambled to position.

"Begin to receive data."

Jack leaned over the flimsy rail as the POKEYs began to pull in the blue data bits from the input port across the hall. He just could barely see from where he was standing, so he leaned a little farther out.

"Be careful, Jack," Linda whispered urgently, "you'll fall off."

Jack hardly heard her, he was too busy trying to see the data the POKEYs were onloading. He leaned farther.

Now he just could make it out. It seemed to be made up of invoices and bills of lading for some kind of parts. He leaned farther still, and as he did, the thin metal of the rail gave way. Jack tried to grab it, but it was too late, his body fell, tumbling to the surface of the serial bus below.

Linda screamed. Suddenly the chamber grew very quiet, except for a crackle from the red sphere pulsing in the center of the room.

The sphere crossed the floor to where Jack lay crumpled. Its electric tentacles reached out to touch his face. He moaned.

"He's alive!" Linda ran from the catwalk and raced down the corridor toward the floor of the I/O chamber.

The sphere spoke. "Jack Truehart, my old friend. We meet again. I should have thought that what happened to McGinty would have served as a warning to you. Ah, but then you were always the impetuous fool."

The sphere backed up and issued a command to the POKEYs.

"Prepare to send data."

The POKEYs moved into position at the head of the tunnel where Jack lay.

"Send data."

The POKEYs pulled at the packets of data that lay at the rear of the hall. A stream of data bits began to flow along the tunnel, gathering force as it approached Jack's body. It soon surrounded him in a rising tide of green and blue, pulling him from view down into the black hole of the serial bus.

Jack, barely conscious, opened his eyes in time to see the light of the I/O hall fade to darkness as the data stream carried him toward a place from which no man had ever returned alive.

Episode 3: Next Issue

True Confessions of a Computer Mama

by Sandra Markle

I never intended to know anything about computers.

Eight years ago when my husband, Bill, went back to school to get his master's degree, computers were different. The machine he worked with was a giant residing in a giant-sized apartment in another building. Now, that was a computer that fit its brain.

I have trouble trusting a computer that doesn't even fill up the top of my coffee table. Particularly when it can convert my grandmother's recipe for chicken and dumplings into proportions large enough to serve the entire Boy Scout troop faster than I can get my pencil to my lips. That's my beginning thinking position for math problems. You can't see a computer puzzling away.

That big computer didn't try to be chummy or even slightly user friendly either. Bill had to take courses to learn its alien tongue. And then to get it to perform any tasks, he had to spend long hours putting his messages on punched cards. These he fed into a card reader — a go-between machine that decided if the message was in a form worthy of the computer's time.

Whether the program worked or not, when the computer was done with the punched cards, they were still useful. I was teaching fifth grade at the time, and I loved to save up boxes full of cards for Christmas. Sprayed green, spindled and stapled, they made wonderful wreaths. What can you do with leftover printouts? Not a lot.

Those were quieter, more peaceful times. Disks were still used by strong-armed men in olympic competitions and keyboards were played by musicians. My house wasn't filled with beeps and blips, pops and peeps. *Pac-Man* hadn't been born yet, let alone taken a wife, had a baby and become supercharged.

No one was concerned about defending Earth from communist mutants from space, safely guiding a frog to a lily pad or building burgers — unless they worked at the local fast food chain. My watch still needed to be set and was perfectly capable of gaining or losing time unexpectedly. Machines never talked to me. They certainly never called me on the phone to ask my opinion.

Just a few short years ago, computers fit the old adage "bigger is better." Now, a little computer is faster and has more memory than its giant ancestors. Who knows how many computers I'm really living with? There's the one I can see with its disk drive, its monitor and its keyboard. But what about the little special-purpose computers that each do only one job. They're hiding in my dishwasher, my microwave oven, my car's engine, my camera.

There has been no stopping it. Byte by byte, I've become computer literate without ever planning to. I first realized what was happening to me when I started putting slashes through my zeros. When my son used the wrong word in a sentence the other day, I told him he had a "syntax error." When the girl at the deli sliced the smoked turkey breast too thin, I told her there was a "type mismatch." When a car suddenly pulled in front of me on the expressway, I felt automatically for a firing button on the knob of my stick shift.

"Look how well it goes with the room now, honey!"

So okay, computers are no longer super giants that are kept in separate buildings and used by specially trained technicians. They're small enough and cheap enough to be brought home and simple enough for me to use. I can live with that, but I'd like to make one request to all the many computer manufacturers.

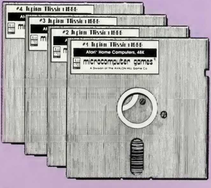
Could you please give those machines some style.

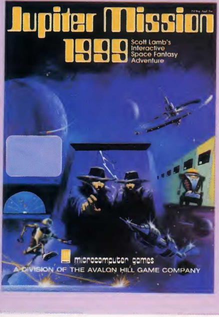
I mean, do we have to go through the Model T stage again? Can't I have my home computer in eary American with ceramic knobs? Or at least, how about some color? I'd like the case in something besides drab — something that would go with the wallpaper in my family room. Can't my computer come in wild plum?

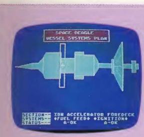
Jupiter Mission 1999™

Scott Lamb's Interactive Space Fantasy Adventure

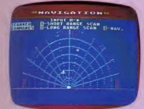
Jupiter Mission 1999 is Avalon Hill's Trademark Name
for its Microcomputer Space Adventure Role Playing Game.



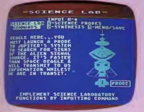

Jupiter Mission 1999
Scott Lamb's Interactive Space Fantasy Adventure
microcomputer games
A DIVISION OF THE AVALON HILL GAME COMPANY



Ship Status Display:
Damage report



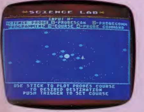
Navigation Display:
Used to plot course



BEAGLE HERE...YOU MUST LAUNCH A PROBE TO JUPITER'S SYSTEM TO SEARCH FOR SIGNS OF THE ALIEN SIGNAL SOURCE. IT'S FASTER THAN SPACE BEAGLE & WILL TRANSMIT TO US INFORMATION WHILEST WE ARE IN TRANSIT.

IMPLEMENT SCIENCE LABORATORY FUNCTIONS BY INPUTTING COMMAND

Science Lab Display:
Probe report



USE STICK TO PLOT PROBES COURSE TO DESIRED DESTINATION PUSH TRIGGER TO SET COURSE

Science Lab Display:
Jupiter system diagramatic

® Trademark of
Warner Communications

RRRING!!! RRRING!!! RRRING!!!

Too early on a chilly January morning, I was jarred awake by the noxious blaring of my traitorous doorbell. As I moved to accost the unknown aggressor, with the full force of semi-conscious wrath, I pulled on my robe and lost my dignity to the pain of a stubbed toe. Now fully awake, I opened the door, prepared to educate the mysterious interloper on the meaning of manners. My determination to this end was somewhat shattered when I saw two large men clad in long overcoats and wide-brimmed hats. Instinctively, I tried to slam the door. My retreat to safety was denied by the advance of the strangers. Before I could protest their entry, my vision was drawn to the shining silver badges that hung from their now unfolded wallets. They were government agents.

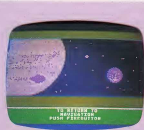
Hesitantly, trying to remember any crimes that I had ever committed, I invited them into my home. At their request, I produced my driver's license and other forms of identification. After examining these credentials, they asked me to pack a bag for an extended journey. After some protest and argument, I was made to understand that my options in this matter were less than limited. My country needs me, they said—with the clear implication that either I pack and dress or I take an extended journey in my robe.

This is how my adventure began. From my cold apartment, I was taken to a towering vehicle for an emergency mission to Jupiter. My very life on the line and, possibly, the survival of the planet Earth as well, and only God knows what other kind of perils await.

JUPITER MISSION 1999 is a highly detailed role-playing space adventure game that includes challenging arcade segments and mystifying puzzles. **Four separate program disks** are enclosed to test the creative imagination of the advanced computer gamer. **JUPITER MISSION 1999** is ready to run on your Atari® Home Computer with 48K memory and one disk drive. **$50.00**

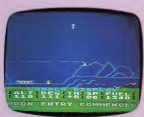
Available at leading computer software dealers, or call toll-free for further information: 1-800-638-9292

AH QUALITY
microcomputer games®
A Division of **The Avalon Hill Game Company**
4517 HARFORD ROAD, BALTIMORE, MD 21214 1-800-638-9292



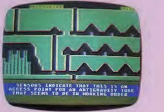
TO RETURN TO NAVIGATION PUSH FIREBUTTON

Porthole View
of Jupiter and a moon



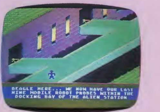
MOON ENTRY COMMENCED

Lander Approach Display:
note descending spaceship



SENSORS INDICATE THAT THIS IS AN ACCESS POINT FOR AN ANTIGRAVITY TUBE THAT SEEMS TO BE IN WORKING ORDER

Exploring an
Alien Complex



BEAGLE HERE... WE NOW HAVE OUR LAST NINE MOBILE ROBOT PROBES WITHIN THE DOCKING BAY OF THE ALIEN STATION

Exploring an
Alien Space Station

Joystick required