INSIDE: PUBLIC DOMAIN SOFTWARE

ATARI® CONNECTION™

Arcade Game Design!

FALL 1983 THE HOME COMPUTER MAGAZINE $3.00

LOGO vs. BASIC

WHICH IS BEST ?

Seymour Papert
Logo Creator

Arthur Luehrmann
Language Expert

SPECIAL EDUCATION ISSUE

TRUNKS FOR THE MEMORIES.

Introducing the most logical place to store Elephant Memory Systems® (or lesser brands of disks): The Trunk.

With its alphabetized library index, you can file or retrieve up to 60 disks, instantly.

The Trunk is made of durable molded plastic with a hinged, one-piece lid, to keep disks safe from dust, dirt, and other detriments which disks despise.

And, it's portable. Because the lid doubles as a carrying handle so your Elephant Memory Systems® disks can go anywhere you do.

There's a model for 5¼" and 8" floppies, as well as a cassette-and-game file and a special Atari® version.

So if you're looking for the best disk storage system on the market . . .

The Trunk is an open-and-shut case.

THE TRUNK. ENDORSED BY ELEPHANTS.

Elephant Memory Systems® Disks

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a life-time of heavy-duty use).

Marketed exclusively by Leading Edge Products, Inc., Information Systems and Supplies Division, 55 Providence Highway, Norwood, Massachusetts 02062. Call: toll-free 1-800-343-8413; or in Massachusetts call collect (617) 769-8150. Telex 951-624.

FALL 1983 VOL. 3, No. 3

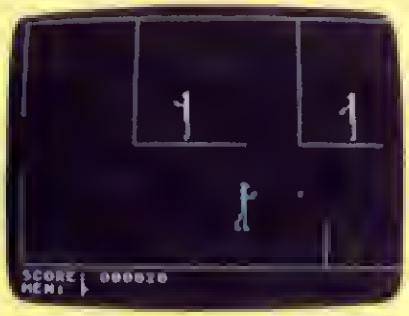
CONTENTS

FEATURES

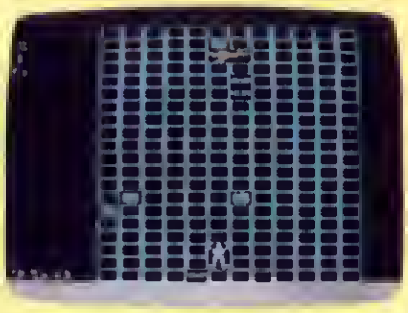
DEPARTMENTS

Cover photography: Seymour Papperr by Ted Horowitz. Arthur Luehrmann by Jeffery Newberry. Computer enhanced photography by Jerry Blank.

ATARI CONNECTION is published quarterly by the Atari Products Company of Atari, Inc., 60 E. Plumeria Drive, San Jose, CA 95134. Third-class postage paid at Sunnyvale, CA and at additional mailing offices.

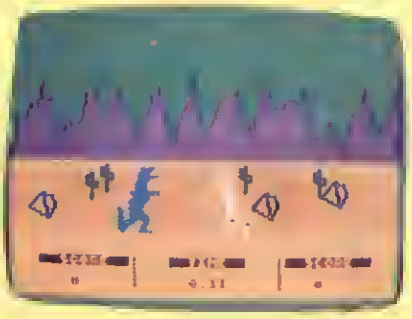


Maniac!

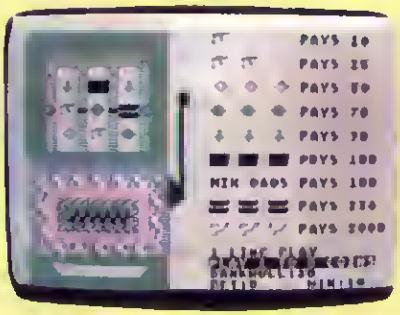


Stuntman

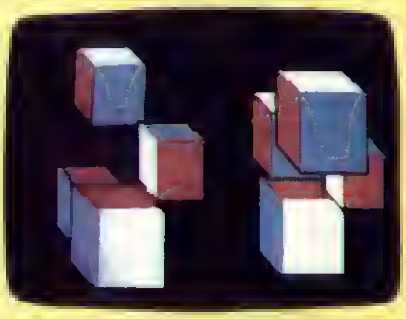
Fill 'Er Up



Dino Battle

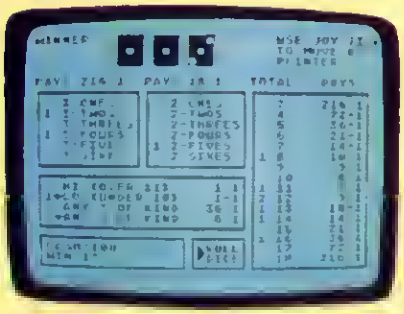


Color Slot Machine

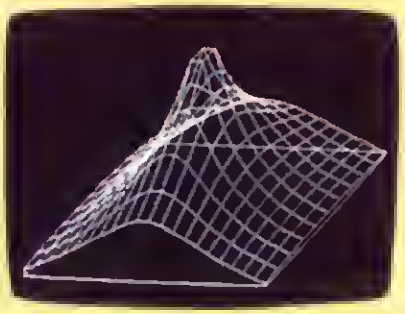


Cubes

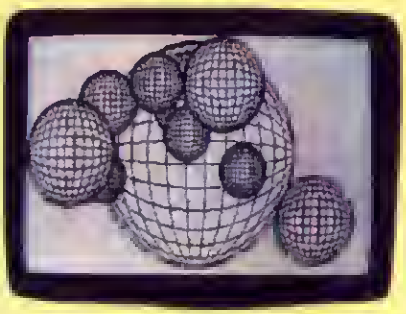
Where can you get all of these programs (and dozens more!) for only $14.95?



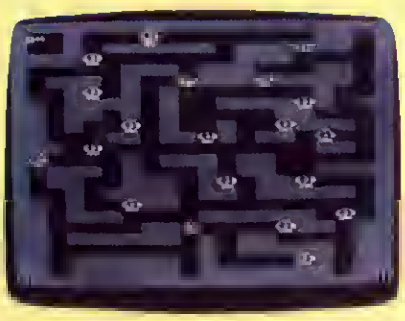
Triple Threat Dice



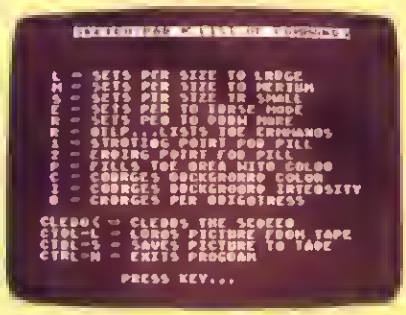
3-D Graphs



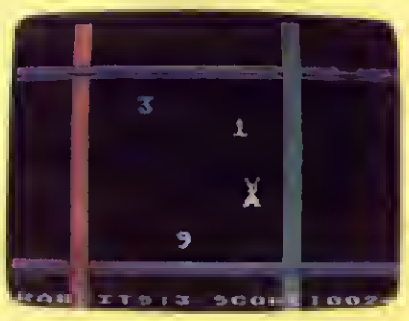
Sphere Demo



Leprechaun King

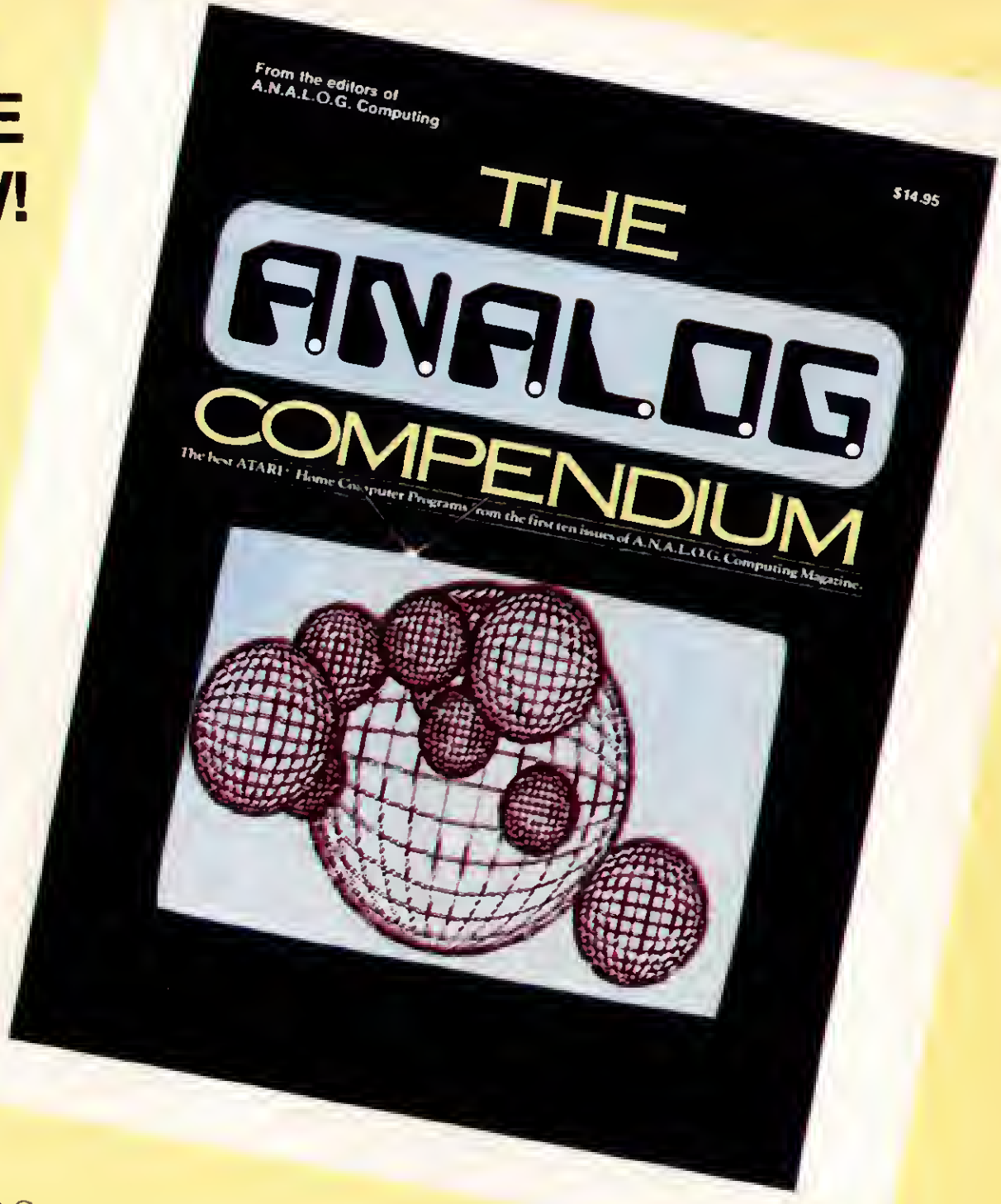


Sketch Pad



Harvey Wallbanger

ON SALE NOW!



The **A.N.A.L.O.G. Compendium** contains the best ATARI programs from the first ten issues of **A.N.A.L.O.G. Computing** Magazine, including new material not previously published. It is packed with utilities, educational and business programs, including an updated version of Tony Messina's "Disk Tool." And, to top it off, The **Compendium** contains the best game programs ever published in any magazine, including *Maniac*, *Stuntman* and the **A.N.A.L.O.G.** favorite, *Fill 'Er Up!* (All shown opposite page, top.)

The **A.N.A.L.O.G. Compendium** is available at selected book and computer stores, or you can order it direct. Send a check or money order for $14.95 + $2 shipping and handling to:

A.N.A.L.O.G. Compendium
P.O. Box 615
Holmes, PA 19043

Or you can order by phone with MasterCard or VISA. Call toll free: 1-800-345-8112 (in PA, call 1-800-662-2444).

For orders outside the U.S., add an additional $5 air mail, $2 surface.

SUPPLIES ARE LIMITED, SO ORDER YOUR COPY TODAY!

ATARI CONNECTION

The Home Computer Magazine

EDITOR
Ted Richards

CIRCULATION MARKETING AND ADVERTISING MANAGER
Julie Clerou Riegel

MANAGING EDITOR
Paul Cohen

ART DIRECTOR
Charles Rush Denson

PRODUCTION MANAGER
Thomas Clark

EDITORIAL ASSISTANT
Myrna Rae Johnson

CONTRIBUTING EDITORS
Markene Kruse-Smith, Elizabeth Metzger, Dr. Steven Young

SENIOR TECHNICAL EDITOR
David Heller

STAFF WRITERS
Jim Carr, Teddi Converse, Carlos Greth, Stephen Englehart, Gary Paul Fox, Paula Polley

CONTRIBUTORS
Bill Bartlett, Jason Gervich, Margaret Harrison, Dorothy Heller, Herb Kohl, Janet Littlefield, Dave Menconi, Bruce Parker, Earl Rice, Lee Sherman, Kent Smith, Jane Sokolow, Cassie Stahl, Stephen Vincent

MANAGER, WRITING STAFF
Jim Inscore

ADMINISTRATIVE ASSISTANT
Paula Surian

CIRCULATION ASSISTANT
Hydee Decker

TYPOGRAPHY
CBM Type

PRINTING
W.A. Krueger Co.

ADVERTISING REPRESENTATIVES
Parris & Company
2058 20th Stteet
San Francisco, CA 94107
(415) 648-6660

Please submit articles to Editor, ATARI CONNECTION, Atari, Inc., P.O. Box 50047, San Jose, CA 95150. Subscriptions: $10.00 for one year, $19.00 for two years, $25.00 for three years. For information on subscriptions and changes of addresses, please write Subscription Department, ATARI CONNECTION, Atari, Inc., P.O. Box 50047, San Jose, CA 95150.

Letters From Our Readers

Dear ATARI CONNECTION,

What language does BASIC use in foreign countries such as Germany, France and Spain? Must programmers use English or can they program in their native tongues? Does the BASIC cartridge have different versions for different countries?

J.P. HAMMERTON
Tucson, Arizona

Simply put: English, English and no. To our knowledge, BASIC in English is the only programming BASIC available for the Atari Computer.

■

Dear ATARI CONNECTION,

ATARI CONNECTION has been helpful in my adventure into the 6502 chip, but I'm having trouble with player/missile graphics—please help.

JOHN DECLAIR
Hyde Park, New York

You're in luck! Dr. C. Wacko's Miracle Guide to Designing and Programming Your Own Computer Arcade Game, *published by Addison Wesley, $12.95, is being serialized for the next three issues. Dr. Wacko is filled with semi-serious graphics and detail, which we are sure will help you out of the depths of your computer confusion. Also in this issue,* Compurer Classroom, *by Bill Bartlett, features player/missile graphics. Then check out the "Bits and Pieces" section article about player/missile graphics by the real Dr. Wacko, David Heller. By the time you finish all of that, your players will be dodging your missiles.*

■

Dear ATARI CONNECTION,

Our family of three acquired an Atari 800 about 15 mnnths ago, with accessories slowly following. Elizabeth, who is nearly five, received her own Atari 400 a month ago. She was thrilled. So was I. Peanut butter goes better with an Atari 400 than an Atari 800.

As I am a publicly employed attorney, the computer had no direct value to my work. However, when our office was directed to automate, my experience with the Atari Computer turned out to be a big help. No one else had had any significant hands-on experience with computers!

WILLIAM L. HENSON
Woodland, California

■

Dear ATARI CONNECTION,

There were several technical errors in the article "Mumbo Jumbo" written by Jim Carr (Spring 1983 issue):

Mark and space frequencies are not limited to those mentioned in the article. They are different for varinus modem types. Acoustically coupled modems and direct connect modems are similar in that both are digital on the terminal side and analog on the TelCo interface side. The only difference is in how they connect to the TelCo line —one is plugged directly into the line and nne is indirectly connected. Contrary to the article, the direct connect modem does nnt use "vnlrage levels that are the equivalent of the 1270 or 1070 Hz signals." Both types of modems transmit and receive audio tones on the TelCo side of the modem. The reason a direct connect modem may be hooked directly to the TelCo line is that a DAA (data access arrangement) has been built into such modems.

The upper limir for data transmission over the switched network is no longer 300 bps, nor is special conditioning needed to push the transmission rate to 1200 bps and beyond. The 1200 bps modems are in common use all over the world and 2400 bps modems are now available from Concord and Codex, with others sure to follow. In the last two years, the price of 1200 bps modems has dropped to the cutrent low of around $500, which makes them competitive with the now only slightly cheaper 300 bps models.

Asynchronous modems do not insert start and stop bits. The terminal equipment does that.

Synchronous modems are not necessarily expensive. Most Bell 212A-compatible modems can be run in 212 mode either synchronously or asynchronously. A reasonably good modem of this type should cost no more than $600.

Synchronous data transmissions up to 4800 bps half-duplex take place over the switched network every day. The Bell 208-comparible modem does the job nicely.

HENRY N. MANZ
Alexandria, Virginia

You didn't give us your credentials, but we assume you must have "Ma Bell" experience at your end of the line. Thank you for taking the time to correct us and sharing your expertise.

AMDEK COLOR-I MONITOR . . .
IT'S ALL ATARI COMPATIBLE!

Amdek's Color-I monitor is designed to enhance the performance of your Atari 800 or 1200 personal computer. It's quality 260(H) x 300(V) line resolution means you'll get more realism with games . . . improved, richly colored graphics with business programs. And all you need for full compatibility is an Amdek AC-300 interface cable!

There's a built-in speaker and audio amplifier . . . and front controls for easy adjustment. It's UL and FCC approved and newly warranteed for a full two years! See your authorized Amdek dealer for a demo . . . or write Amdek for complete specifications & prices . . . and don't forget to ask about the Color-I PLUS monitor that has a special non-glare screen and headphone jack with a FREE headset.

2201 Lively Blvd. • Elk Grove Village, IL 60007
(312) 364-1180 TLX: 25-4786

AMDEK

REGIONAL OFFICES: Calif. (714) 662-3949 • Texas (817) 498-2334

Amdek . . . your guide to innovative computing!

War Games: A Lesson in Futility

WE HAVE A NEW "computer character": W.O.P.R., an acronym for War Operating Planned Response. W.O.P.R. dethrones the obsolete HAL of *2001 A Space Odyssey* fame. W.O.P.R. starred as the "leading computer" in this summer's hit movie, *War Games**. Unlike HAL, who portrayed the monster computers of the sixties, W.O.P.R. is a prodigy of the home computer age—besides having "legitimate" database management responsibilities as North American Defense Command's (NORAD) master computer—it's also the ultimate video game.

W.O.P.R. was originally named *JOSHUA* after the deceased son of its creater/programmer, Prof. Stephen Falken. Only when it was assigned its role as NORAD's master computer did it assume the identity of W.O.P.R. Beneath its cold, logical computer exterior lies a big kid —JOSHUA—who's been programmed to win every game it plays. Its "secret identity" is known only hy Prof. Falken and a young 17-year-old computer whiz kid, David Lightman.

When David Lightman, in his not-so-above-board quest for the hottest new computer games, stumbles into W.O.P.R.'s game menu, the password "JOSHUA" triggers its latent identity. The last game listed is "Global Thermonuclear War."

Although both computer movies, *2001* and *War Games*, are decades apart, they share a modern 20th-century theme: Humanity losing control of its machines—in this case, our computers going hezerk. But, in *War Games* we have a new twist: The human interface has evolved —David Lightman and his mentor, Prof. Falken, beat the computer at its own game.

In the climactic scène of *War Games*, as the seconds tick away towards a nuclear holocaust, David, at the behest of Prof. Falken, coaxes JOSHUA to try a game of tic-tac-toe.

JOSHUA begins playing tic-tac-toe against itself, while W.O.P.R. prepares to launch the nation's arsenal of Minuteman missiles. After a few quick games, the futility of tic-tac-toe dawns upon the schizophrenic computer. Once the players become familiar with the game's simple strategy, no one can win. Suddenly, W.O.P.R./JOSHUA makes the ultimate Artifical Intelligence correlation: Thermonuclear War is like tic-tac-toe: "It's futile . . . Would someone like to play a nice game of chess . . .?"

What does all of this have to do with computer education? I think the movie raises two important issues.

First, I thought it was ironic that the computer was the one who learned the lesson. We were led to believe, from that moment on, all computers, when asked to "play" Thermonuclear War, will respond on their screens (or speech synthesizers): "Sorry, that game is futile." That is, if computers learn from their mistakes as humans do. . . .

Second, the character David Lightman represents a modern "computer education paradox": Here's a bright, computer-literate kid who's a poor student, an underachiever, a disciplinary problem—he even changes his grades by accessing his school's computer. Yet, in the course of his adventure, he becomes "human literate." He learns there is a larger world in which he lives. A world of people with human feelings, desires, aspirations and dreams. Lest we forget, with all the current emphasis on "computer literacy," there should be a human quality to the "information" we process and send flitting across the nation's networks. If we forget to teach our humanity, then the machines will indeed rule our lives. —Ted Richards

*A Leonard Goldberg Production. © 1983, United Artists Corporation. All Rights Reserved.

wabash® diskettes for as low as $1.29 each!

Now get Wabash Quality at a CE Price

For over 17 years, Wabash has been making high quality and dependable computer products. Wabash diskettes are made to provide error-free performance on your computer system because every diskette has been totally and hypercritically tested. Since you can now buy Wabash computer products directly from CE, the world's largest distributor of magnetic media, you can now get maximum savings on every order. You can even order toll-free.

New Wabash Six Year Warranty

The quality of Wabash diskettes is stressed throughout the entire manufacturing process. After coating, all Wabash diskettes go through a unique burnishing process that gives each diskette a mirror-smooth appearance. Wabash then carefully applies a lubricant that is specially formulated to increase diskette life. Then, to keep out foreign particles, a unique heat seal bonds the jacket and liner together to help prevent contamination. After 100% hypercritical testing and certification, Wabash then packages each diskette, (except bulk pack) in a super strong and tear resistant Tyvek® evelope. The final Wabash product is then shrink-wrapped to insure cleanliness and reduce contamination during shipment. Wabash diskettes are so very reliable that Wabash now offers a six year warranty in case of defects in materials or workmanship on all diskettes purchased directly from Communications Electronics.

New...Wabash Diskette Duplication Services

Communications Electronics has teamed up with Wabash to provide a single-source solution for the diskette duplication requirements of software developers, OEM's and distributors. All service is in-house, to give you fast, dependable service. In most cases, delivery can be completed in five days. Whether you require 100, 1,000, or 10,000 copies per week, call CE first for a no obligation price quote. For additional information, please write us on your letterhead with your requirements.

SAVE ON WABASH DISKETTES Product Description	Part #	CE quant. 100 price per disk ($)
8" SSSO IBM Compatible (128 B/S, 26 Sectors)	F111	1.89
8" SSSD Shugart Compatible, 32 Hard Sector	F31A	1.89
8" SSOD IBM Compatible (128 B/S, 26 Sectors)	F131	2.39
8" DSDD Soft Sector (Unformatted)	F14A	2.99
8" OSOO Soft Sector (256 B/S, 26 Sectors)	F144	2.99
8" OSOD Soft Sector (512 B/S, 15 Sectors)	F14S	2.99
8" OSDO Soft Sector (1024 B/S, 8 Sectors)	F147	2.99
5¼" SSSD Soft Sector w/Hub Ring	M11A	1.49
5¼" Same as above, but bulk peck w/o envelope	M11AB	1.29
5¼" SSSO 10 Hard Sector w/Hub Ring	M41A	1.49
5¼" SSSO 16 Hard Sector w/Hub Ring	MS1A	1.49
5¼" SSOO Soft Sector w/Hub Ring	M13A	1.79
5¼" Same as above, but bulk pack w/o envelope	M13AB	1.59
5¼" SSOO 10 Hard Sector w/Hub Ring	M43A	1.79
5¼" SSOO 16 Hard Sector w/Hub Ring	MS3A	1.79
5¼" OSOO Soft Sector w/Hub Ring	M14A	2.69
5¼" Same as above, but bulk pack w/o envelope	M14AB	2.49
5¼" OSOO 10 Hard Sector w/Hub Ring	M44A	2.69
5¼" OSDD 16 Hard Sector w/Hub Ring	M54A	2.69
5¼" SSQD Soft Sector w/Hub Ring (96 TPI)	M1SA	2.59
5¼" OSQO Soft Sector w/Hub Ring (96 TPI)	M18A	3.69
5¼" Tyvek Diskette Envelopes - Price per 100 Pack	TE5	12.00

SSSO = Single Sided Single Oensity; SSOO = Single Sided Oouble Oensity; OSOO = Double Sided Oouble Oensity; SSQO = Single Sided Ouad Oensity; OSQO = Oouble Sided Oued Oensity; TPI = Tracks per inch.

Quantity Discounts Available
Wabash diskettes are packed 10 disks to a carton and 10 cartons to a case. The economy bulk pack is packaged 100 disks to a case without envelopes or labels. Please order only in increments of 100 units for quantity 100 pricing. With the exception of bulk pack, we are also willing to accommodate your smaller orders. Ouantities less than 100 units are available in increments of 10 units at a 20% surcharge above our 100 unit price. **Quantity discounts** are also available. Order 500 or more disks at the same time and deduct 1%; 1,000 or more saves you 2%; 2,000 or more saves 3%; 5,000 or more saves 4%; 10,000 or more saves 5%; 25,000 or more saves 6%; 50,000 or more saves 7%, 100,000 or more saves 8%, 500,000 or more saves 9% and 1,000,000 or more disks earns you a 10% discount off our super low quantity 100 price. Almost all Wabash diskettes are immediately available from CE. Our efficient warehouse facilities are equipped to help us get you the quality product you need, when you need it. If you need further assistance to find the flexible disk that's right for you, call the Wabash compatibility hotline. Dial toll-free 800-323-9868 and ask for your compatibility representative. In Illinois or outside the U.S. dial 312-593-6363 between 9 AM to 4 PM Central time.

Buy Wabash Diskettes with Confidence
To get the fastest delivery from CE of your Wabash computer products, we recommend you phone your order directly to our Computer Products Division and charge it to your credit card. Be sure to calculate your price using the CE prices in this ad. Written purchase orders are accepted from approved government agencies and most well rated firms at a 30% surcharge for net 30 billing. For maximum savings, your order should be prepaid. All sales are subject to availability, acceptance and verification. All sales are final. All prices are in U.S. dollars. Prices, terms and specifications are subject to change without notice. Out of stock items will be be placed on backorder automatically unless CE is instructed differently. Minimum *prepaid* order is $50.00. Minimum *purchase* order $200.00. All shipments are F.O.B. Ann Arbor, Michigan U.S.A. No COD's please. Non-certified and foreign checks require bank clearance.

For **shipping charges** add $8.00 per case or partial case of 100 8-inch flexible disks or $6.00 per case or partial case of 100 5¼-inch mini-diskettes for U.P.S. ground shipping and handling in the continental U.S.A.

Mail orders to: Communications Electronics, Box 1002, Ann Arbor, Michigan 48106 U.S.A. If you have a Visa or Master Card, you may call and place a credit card order. Order toll-free in the U.S. Dial 800-521-4414. In Canada, order toll-free by calling 800-265-4828. If you are outside the U.S. or in Michigan dial 313-994-4444. Telex anytime 810-223-2422. Order your Wabash diskettes today.

 Ad # U12483

MasterCard VISA MEMBER DIRECT MAIL MARKETING ASSOCIATION

Order Toll-Free!
800-521-4414
In Michigan 313-994-4444

wabash error-free diskettes

COMMUNICATIONS ELECTRONICS™

Computer Products Division

818 Phoenix ☐ Box 1002 ☐ Ann Arbor, Michigan 48106 U.S.A.
Order TOLL-FREE 800-521-4414 or outside U.S.A. 313-994-4444

NEWS

HOME COMPUTER

FLYING HIGH WITH SUPERMAN

Atari Animation Wizards Rescue Super Hero

Fans of the *Superman III* movie will remember the computer animation scene as one of the best of the film's many special effects: The Man of Sreel is flying through the Grand Canyon while the evil genius Ross Webster tracks him with a supercomputer. He attacks him with missiles, and watches the action on his video monitor. The sequence, painstakingly designed to look like a video game, is the product of a unique computer animation technique developed by Atari's Special Programs group.

The trick was making an extremely complicated process look easy. "Our objective was not to create realistic, high-resolution compurer animation, but to produce a scene that looked like an advanced video game, something you might see five years from now," says Director of Special Programs Steve Wright. "The animation is far more sophisticated than what is possible with today's video game technology," says the team's computer graphics expert, Pat Cole. Wright and Cole are part of an advanced design group at Atari's new 3D Computer Animation Facility, which will be used to develop laser disk animation for the next generation of computer games.

The 26-second animation sequence in *Superman III* represents three-and-a-half months of work and a $125,000 budget. The scene was shot in classic stop-frame animation fashion, using a Dunn Instrument fitted with a 35mm Mitchell camera.

"It took two to five minutes to compute each frame," says Cole, who handled the sophisticated mathematics involved, and whose credits include computer animation for Lucasfilm and Carl Sagan's *Cosmos* series for PBS Television. More than 600 frames were shot to create the *Superman III* sequence.

The computer animation system, under the guidance of Paul Hughett, allowed programmers to "paint" images into the computer's memory. The system included a "Paint Program," an "Animation Editor" and a "Final Rendering Program" which blended the graphics and motion together.

Vicki Parrish wrote the scene's computer script, which, says Director Steve Wright, was based upon storyboard frames that resembled engineering graph paper.

The scene has earned kudos from *Variety*, and is to be screened at the Special Interest Group on Computer Graphics (SIGGRAPH) conference. This may be a harbinger of things to come. But, home programmers who have been studying computer animation techniques in these pages, beware: you'll need an expensive Symbolics LISP computer and a few fancy peripherals to approximate the super effects produced by Atari's animation wizards.

. . .

In a related story, Hollywood sources inform us that an Atari 800 Computer played a supporting role in a

Edited by Elizabeth Metzger

film last spring. You won't actually *see* the computer in *Deal of the Century*, scheduled for release this fall. What you will see is a monitor inside the cockpit of a fighter plane. The Atari 800 was the brains behind that display, relaying messages to the pilot.

"Our group wrote the software program for the display," explains Steve Davis, director of Atari's Los Angeles Lab, Corporate Research Division. "When Burbank Studios asked us if the Atari 800 could do the job, I answered with a strong affirmative."—Paul Cohen and Elizabeth Metzger

GAME DESIGNING 101

Joysticks Required

In the '60s the "hot" college courses were French Existentialism and American Foreign Policy; in the '70s students flocked to Eastern Religions and Humanistic Psychology. This past summer, the University of Wisconsin–Milwaukee's Computing Services Division debuted its crowd-pleaser for the '80s—"Computer Game Design" and "Intermediate Graphics and Animation," both designed with the Atari 800 Home Computer in mind.

"These are the first courses we're offering with the Atari Computer," says Thomas Krischan, a management information specialist who teaches short courses at UWM when he isn't working as a consultant on microcomputer problems. "It's part of a general trend here to offer more courses using microcomputers."

The two non-credit games courses taught students how to plan a game strategy and to solve the nitty-gritty technical problems of game design. Both required some prior familiarity with computers.

"These courses will be part of a continuing program," says Krischan. "If the response is positive, we'll offer them again during the regular academic year."

For more information, contact The University of Wisconsin–Milwaukee, Computing Services Division, P.O. Box 413, Milwaukee, Wisconsin 53201.

Win Big in APX Contest

$29,000 in Cash and Prizes for New Peripherals-Users

Home programmers with especially useful, original or well-designed software know of the Atari Program Exchange (APX) as an important outlet for their ideas. With more than 200 titles in its catalog, APX has always rewarded programmers with royalties, as well as quarterly contest prizes for outstanding programs.

Now, beginning in January, software designers can win prizes worth an additional $6,000 if their programs make use of at least one of these new Atari peripherals: the *Atari Touch Tablet*, the *Atari Trak-Ball*, the *Atari 1020 Color Printer*, or the Speech Synthesizer feature of the Atari 1400 and 1450XL Home Computers. Programmers who adapt their current APX software for use with peripherals are also eligible for prizes.

Three contestants each quarter who make the best use of these new peripherals will have a chance at prizes of $3,000, $2,000, or $1,000 in Atari and APX products. Winners will then be eligible for a super cash prize of $5,000.

For complete information, check the Fall issue of the APX catalog, or phone (800) 538-1862, or (800) 672-1850 in California.

Nuclear Troubleshooter Stays Sharp with Scram

Video Game Imitates Life

Harold Denton tries to beat the reactor at its own game.

When Harold Denton, chief of reactor regulations for the Nuclear Regulatory Commission, bought his Atari Home Computer a year and a half ago, one of the first programs he tried was *Scram*, the Atari game that simulates a nuclear power plant.

"It's a great game," says Denton, a physicist who earned prominence during the Three Mile Island accident four years ago. When not keeping the heat on the nation's nuclear industry, he may be found entertaining fellow scientists with a friendly simulation of some worst-case scenarios. The game offers an "excellent representation" of the effects of an earthquake on a reactor, says Denton.

While it can't replace the $3 million simulators (used to train reactor operators), "*Scram* is a great tool for learning what goes on in a reactor," says Denton. "It's a fascinating game."

Denton is equally impressed with the rest of his system. After checking out other computers, Denton settled on the Atari 800 and a cassette recorder. Since then he's added a disk drive, printer, voice synthesizer and modem. "It's kind of like having a piano. It's become a piece of standard equipment around here," he says of the system in his Rockville, Maryland, home. "My son's in high school and uses the computer routinely. It's improved his schoolwork, especially his writing. My wife teaches school, and I've written a computer gradebook for her, using VisiCalc. It keeps individual test scores for 80 students as well as cumulative averages. I use the system for home finance and word processing."

By Denton's account, programmer Chris Crawford (see page 40) has scored another hit with *Scram*. So, has the nation's most renowned nuclear regulator ever experienced a meltdown? "Once or twice," jokes Denton, "deliberately." It's the one experiment he can't duplicate in real life.—Paul Cohen

PILOT TAKES OFF IN PENNSYLVANIA SCHOOL

Along with readin', writin', and 'rithmetic, students at Yeadon, Pennsylvania's Evans Computer Magnet School, are getting a strong dose of computer learning by way of Atari PILOT.

Now in its second year, the computer program at the Philadelphia-area elementary school offers classes in computer literacy and applications in a special "computer center." All 400 students and 15 teachers at the school participate in the program.

The school chose Atari PILOT because it allows students to manipulate the computers, "not the other way around," explains principal Thomas G. Kerr. "We also needed a program that would tie in closely with our other instructional areas."

The Evans School staff went one step further. They redesigned the Atari PILOT student manual to fit teachers specific lesson plans. "This became our curriculum guide for all the teachers, from first through sixth grades," says Kerr.

Students were introduced to the computers as "another curriculum area," explains Kerr. "This way they knew it would be hard work, not just fun and games."

"What's exciting is that every student and teacher is involved in this program. Every teacher has had to learn Atari PILOT. At first they were frightened, but after a year behind them, they're looking forward to working with the computers again."

The Evans Computer Magnet School welcomes inquiries from other elementary schools. For more information, contact Dr. Thomas G. Kerr, The Evans Computer Magnet School, Church Lane and Baily Road, Yeadon, Pennsylvania 19050.

The Atari Institute:

Computers and Lifelong Learning

Since its founding in 1981, the Atari Institute for Educational Action Research has awarded $1 million in equipment and cash grants to groups as diverse as the Rehabilitation Institute of Chicago, for work with people suffering brain damage and requiring rehabilitation; Greenfield Community College, for a computer-integrated simulation of a 72-hour flight aboard the NASA Space Shuttle, and San Quentin State Prison in California, for providing prisoner education.

The Institute's Board of Advisors is especially interested in fostering computer-related education in non-formal—that is, non-school—learning situations, emphasizes Dr. Ted Kahn, the Institute's founder and director. As a result, grants have gone to such dissimilar institutions as Stanford University in Palo Alto, Calif., and the Center for the Development of Non-Formal Education, a community-based bilingual learning center for early-childhood, adult, and family education in the barrios of Austin, Texas. (See page 32 for the complete story.)

Another area of interest for the Institute lies in education and the arts—music, theater, visual art, design, and choreography.

The Institute hopes to use the experience of its project participants as a resource to help Atari develop new equipment and meaningful software programs, particularly those related to education. "The profile of the 'consumer' is a statistical summary, hut there's no such individual," Kahn explains. "We're covering the entire spectrum to try to understand what they need—what kinds of software they use in their environment."

Kahn sees the Institute as a vehicle for developing an electronic network that will, in his words, "create a community." The planned electronic bulletin board system, accessible via modems and normal phone lines, will allow the passing of information—student- and teacher-written programs, for example—from one part of the country to another.

We're going out and creating mini-computer lahs all over the country," Kahn stresses. "The projects we support are, in effect, the largest computer lab Atari has for research and development."—**Jim Carr**

For information and grant applications, write to Sandra Williams, Manager of Program Development, Atari Institute, 1196 Borregas Ave., Box 427, Sunnyvale, Calif. 94086.

Recent Atari Institute Grants

The Exploratorium, San Francisco, California for extending and enhancing exhibits

George Washington High School, Denver, Colorado for the computer lab

Rowland High School, Rowland Heights, California for a nationally acclaimed, computer-generated animation program

San Francisco Education Fund, San Francisco, California to bring the computer age to inner-city school children

Alternate Media Center, New York University, New York, New York to be used in non-broadcast applications of computers

Information Space, Albuquerque, New Mexico to provide American Indian and Chicano high school students with computer science

Computer Catch-Up for Educators at Fordham U.

On-Line 17 Hours a Day

A new Fordham-Atari Center at New York's Fordham University is now the home of 20 Atari Home Computer systems. The computers, peripherals, and software donated by Atari to the Graduate School of Education will be used for teaching educators the ways of computing.

College of Education and Continuing Education classes, along with elementary school students invited into the labs during off hours, have logged an average of more than 17 hours a day of computer time.

The Center offers courses in computer programming and in research and educational applications. It will also serve as a center for testing educational software and developing computer literacy training materials in addition to teaching programming languages.

"Almost half of all high schools in this country currently have computers, making it cnicial that teachers be familiar with basic computer concepts," says Dr. Alfred Moye, Atari's Manager of Educational Sales. "This center will be a big boon to Fordham."

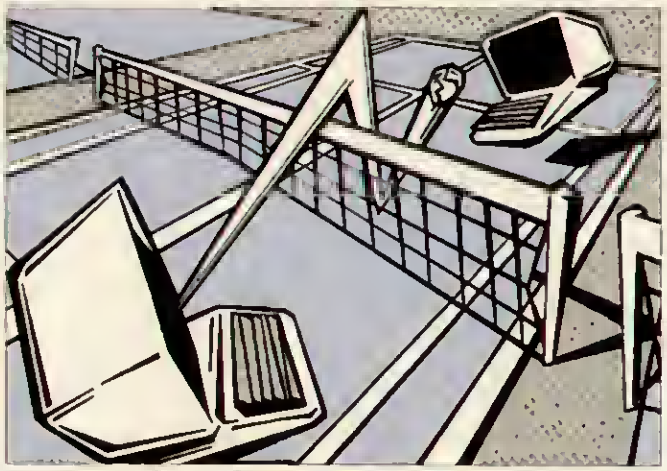
ATARI COMPUTER-ATP PROGRAM RANKS MEN'S PRO TENNIS CIRCUIT

Complete Player Information To Be Available on CompuServe

The Association of Tennis Professionals (ATP), an organization that represents the world's best male tournament players, has teamed up with Atari to develop the ultimate tennis ranking and information system. Using an Atari 800 Home Computer and software developed by Atari programmers, the system calculates rankings for more than 1,000 singles and doubles players each week, and provides biographical information, match and tournament histories and comparative data.

Players, tournament directors, the press and fans around the world will soon be able to call up this information when it is added to the CompuServe Information Service.

According to Ray Moore, president of the 350-member organization, "The ATP computerized rankings revolutionized pro tennis when they began in 1973. Today, we're not only upgrading the system, but we'll also be providing home computer users with immediate access."



Results of all Volvo Grand Prix and other ATP-sanctioned tournaments, along with player entry requests for future tournaments, are sent to the Association's headquarters in the Dallas suburb of Garland, Texas. There, the information is entered into an Atari 800 Home Computer system, which automatically revises all player data including rankings for the prior 12 months.

Updated weekly, the Atari-ATP system provides more player information than ever before. Over 20 item entries on each player—including his ranking, winnings, success on various surfaces, even whether he is right- or left-handed—can be called up on the system. Additional historical and comparative data organized by player and tournament, and entry lists for future tournaments are also compiled.

—Gary Paul Fox

Look for the Atari-ATP Tennis Rankings in the CompuServe index soon.

Instant Entrees in a Box

Everything You Need to Get Into Telecommunications or Accounting

After talking to friends in your Atari Users' Group—or reading an article in ATARI CONNECTION—you've realized that you'd really like to experiment with programming, business applications, telecommunications, or some other specific aspect of home computing.

So you go out and purchase the items you think you'll need and race home bristling with enthusiasm. You're ready to go. Or so you thought.

The salesman at the store forgot to ask you if you owned the right accessories. Or he didn't mention that you'd need a particular diskette program. So now you have to wait for your next free Saturday and next paycheck to put it all together.

Now Atari gives you a way to assemble a complete application system in one simple purchase—for less than you'd pay if you bought each part separately. It's the Atari Add-A-Paks line, and it provides software and hardware packages for everything from business applications and telecommunications to programming and home entertainment.

The Communicator II Atari Add-A-Pak

If you read the Spring 1983 issue of ATARI CONNECTION, you're aware of the almost infinite number of resources available through telecommunications: news services, bulletin boards, home banking and shopping, specialized information services, on-line CB-like personal exchanges, electronic mail, stock market and business news updates, games, novels and cultural reviews.

The Atari Communicator II Add-A-Pak comes with all the accessories and software you need to delve into this ever-expanding world of telecommunications.

The Atari 835 Direct Connect modem plugs right into your phone jack, so there's no fumbling with telephone receivers and couplers. The Telelink II program lets you program in phone numbers and access codes, so there's no more dialing and typing, redialing and retyping frequently used numbers and passwords.

Communicator II includes an easy-to-follow manual as well as special discount offers from Dow Jones News/Retrieval Service and CompuServe Information Service.

Package includes Telelink II program cartridge, Atari 835 direct-modem, and users' guides. Suggested retail price: $279.95. Available now.

The Accountant Atari Add-A-Pak

If you have a small business, you know how important it is to have well-managed, accurate bookkeeping—not just for audits and taxes, but for securing financial backing as well.

With the Atari Accountant, you don't have to be a computer expert to generate an impressive variety of financial reports, because this Add-A-Pak includes the easy-to-learn, double-entry Atari Bookkeeper program. For quick and efficient numerical entries, it comes with the CX85 Numerical Keypad. And the VisiCalc® Handler program enables your CX85 to work perfectly with VisiCalc and other electronic spreadsheets.

The Bookkeeper program

helps you generate all types of professional accounting reports: Profit and Loss Statements to track monthly, quarterly, or annual earnings; Balance Sheets to determine the most profitable way to manage assets and liabilities; General Ledger, Accounts Payable, and Accounts Receivable statements to monitor cash flow; plus Cash Received, Invoices Written, Checks Written, and General Journal print-outs.

—Paula Polley

Package includes The Bookkeeper by Atari program diskettes, CX85 Numerical keypad, keypad handler diskette to run with VisiCalc and other programs, and users' guides. Suggested retail price: $249.95. Available now.

COMPUTER SWAP AMERICA

Computer enthusiasts won't want to miss COMPUTER SWAP AMERICA, "The Personal Computing Bargain Show," on September 10th from 10 a.m. to 6 p.m. at the Santa Clara County Fairgrounds.

Buyers will find complete personal computer systems, peripherals, software, supplies, publications, and consumer electronics—all at bargain prices.

The fairgrounds are located at 344 Tully Road, San Jose. Admission is $5. Sellers call 800-221-SWAP or 415-327-7810, or write to Box 52, Palo Alto, CA 94302.

CALLING EDUCATORS!

The Atari Teachers' Network

Now, elementary school teachers and administrators who use Atari Computers can exchange ideas, experiences, and software through the Atari Teachers' Network.

The newly formed organization publishes a quarterly newsletter which reports on educational programming activities with Atari PILOT and Atari Logo, computer happenings in the classroom, questions and answers about classroom use of computers, and more.

Highlights of a recent issue include a look at voice synthesizers for the Atari Home Computer, and a discussion of ways to ensure that all students get equal computer time.

The Network's co-founders are Nancy Austin Shuller, a computer specialist at the Day School in New York City, and Curtis Springstead, an educational computer consultant.

"I work two days a week in a computer lab with elementary school children," says Nancy. "I'm also working on my dissertation for a doctorate in Mathematics Education."

Curtis is the local Atari PILOT "Expert" for his Users' Group, and is teaching business programming languages at a community college while he pursues his MA in Education with a specialization in computers. "I enjoy teaching as much as doing programming and system design work," he says.

Network membership (newsletter included) is only $4.00 annually.

For more information write to:

Atari Teachers' Network
P.O. Box 1176
Orange, NJ 07051

—Dorothy Heller

Computers Go Preppie

Computers Replace Slide Rules for College Freshmen

Last fall, we told you about a scholarship program at New York State's Rensselaer Polytechnic Institute which awarded Atari Home Computer Systems to 21 top freshmen in the class of '85. Well, the Atari Home Computer seems to be the hot machine on campus these days. Last year, Stevens Institute of Technology in New Jersey required all incoming freshmen in the Science and Systems Planning and Management curricula to own an Atari Home Computer.

The computer requirement, the first in a U.S. college, affects 80 of the 500 members of the class of '86 at Stevens; an additional 40 students and faculty members have also bought the systems. Beginning with courses in "Introduction to Computing" and "Matrix Algebra with Computers," students will continue to take one microcomputer course every semester until they graduate.

"The idea behind the computer requirement is to help the faculty teach better, to help the students learn the material better, and to prepare students better for professional careers," according to Joseph Moeller, Jr., Associate Dean for Educational Development. "Most of our graduates go into engineering, science, and management fields. We think it's important that they be 'computer fluent' so that they will be better professionals."

One impetus for the requirement was the "bottleneck" in Stevens' time-sharing system. "Student use was growing at a rate of 30 percent a year," explains Dean Edward A. Friedman, committee chairperson of the Ad Hoc Computer Planning Committee. "We didn't want our students to hate computing because they could only get on a computer between one a.m. and three a.m.!"

Students purchased the computers at Stevens when they arrived on campus in the fall. To keep costs down to a minimum, students were required to buy only the minimum they would need for a system—a computer, program recorder, and small black-and-white television set—at an approximate cost of $750. A special Peripheral Center, open seven days a week, was set up for students to enhance their systems.

Other colleges across the country have been watching Stevens with interest. "This past year we've received inquiries from over 200 colleges," says Friedman.

As for the not-so-distant future, Friedman predicts that within five years, similar requirements on other campuses will be "commonplace." In the meantime, the Stevens program has been "extremely successful," says Friedman. "We're very, very pleased."

College freshman line up to claim Atari 800 Computers.

ATARI SINGS YOUR FAVORITE SONGS!!!

THE Original VOICE BOX Speech Synthesizer by the ALIEN GROUP has received rave reviews:

MICRO COMPUTING–"The VOICE BOX injects an endearing personality to your computer. The possibilities are enormous."

COMPUTE–"The VOICE BOX offers more human-like tones and does not blank out the screen."

CREATIVE COMPUTING–"English text and phonetic code may be freely intermixed rather than requiring separate modes as is the case without exception with every other speech system. A mode called talking face displays an animated face with impressive lip sync animation."

ANTIC–"There is a great potential for teaching children to spell and an added dimension to games overall. I believe the VOICE BOX is well worth the price tag."

ANALOG–"For ATARI owners who want to add speech to their programs, the Alien Group VOICE BOX is probably the best choice."

POPULAR SCIENCE–"The speech quality is excellent. Besides creating speech, the software has a bit of fun with graphics."

and on the new VOICE BOX II.....

TIME MAGAZINE–"Machine of the Year" "The VOICE BOX by the Alien Group enables an ATARI to say aloud anything typed on its keyboard in any language. It also sings "Amazing Groce" and "When I'm 64" or anything else that anyone wants to teach it.

INCORPORATE THE SINGING HUMAN FACE
INTO YOUR PROGRAMS AND GAMES

VOICE BOX II
Speech & Singing Synthesizer

The New VOICE BOX II for ATARI plugs into the serial port of the ATARI 400/800 with sound coming out of the TV/monitor. 48K DISK is required. It has all of the features of the original VOICE BOX plus many exciting new hardware and software features:

- The ability to sing with voice and 3 part music.
- A library of 30 famous songs.
- A comprehensive music system that allows the user to easily enter or modify new songs.
- Software that can convert the bottom two rows of the ATARI keyboard into a piano with a range of 3½ octaves using the shift and control keys.
- Programmable musical sound effects such as tremolo, vibrato, glissando and click track.
- A singing *human* face with lip-sync animation designed by Jerry White.
- A talking or singing *ALIEN* face with software that allows the user to change the face and 8 mouth patterns as he sees fit.
- The ability to speak with inflection and feeling.
- Can speak in a foreign language with correct foreign spelling as input.
- A talk and spell program by Ron Kramer. Users can program any vocabulary for this spelling game. In fact, this program can even speak in a foreign language like French, where the user must spell the correct word in English, or vice versa.
- GREEN GOBLINS–A talking arcade game by John Wilson.
- Random Sentence Generator–An amusing grammar game that helps teach school children to identify parts of speech and recognize a variety of sentence structures.
- NUMBER SPEAK–A subroutine by Scott Matthews that converts up to a 9 digit number into normal English pronunciation. Ideal for building your own math games.
- STUD POKER–A talking poker game by Jerry White.
- The screen never blanks out while talking or singing.
- Singing or speaking subroutines can be incorporated into your programs, requiring as little as 100 bytes of RAM plus 5 bytes for each word.
- Price $169.00 includes VOICE BOX II and all of the above software.
- Inquire about our discounts for educational institutions.

To order by mail send a check or money order to the ALIEN GROUP for $169. Then, try the VOICE BOX II for 10 days, and if it isn't the finest value you've ever seen in a computer peripheral, the most challenging and provocative addition you've ever made to your system, return it in its original condition for a full refund.

THE ALIEN GROUP
27 West 23rd Street (212) 741-1770
New York, NY 10010

ALSO AVAILABLE AT LEADING COMPUTER STORES THROUGHOUT THE WORLD.

Public Domain Software

How to Find Those Hard-to-Get Special Interest Programs!

by Dorothy and David Heller

BY "NETWORKING" with your Atari Computer, you can discover special interest groups for scientists and ham radio operators, writers, educators and teachers, "Dungeons and Dragons" fans and more. We're talking about an incredible variety of infonnation services and electronic bulletin boards (BBS) that cost no more to use than the price of an ordinary phone call and are available as a resource for you at this moment.

These services include "Dial-Your-Match," to find the computer romance of your life; "The Living Tree," a vast and ever-expanding information bank; and hundreds of other sources for *free public domain software.*

Public domain software is software that has been donared by its authors for public use. Many hobbyists, educators, and amateur software developers aren't primarily interested in making money from their original programs—most public domain software has been written for special applications and system utilities. A special interest bulletin board is the most likely place you'll find those hard-to-get programs no one else seems interesred in publishing.

Public domain programs may not be as polished as rhe best commercial software packages like AtariWriter, but they can be entertaining, useful—and phenomenally inexpensive!

With the right kind of modem and communications software, you can *UPLOAD* programs you want to conrribute to other Atari Computer systems *and bulletin board services, or DOWNLOAD* their programs to your Atari Computer and SAVE them to diskette or cassette.

Get On-Line Today

You'll need some extra equipment hooked up to your system to begin downloading *free software:* a modem and an Atari 850 Interface Module.

The MODEM (MOdulator/DEModulator) is an inexpensive electronic device that lets you use your Atari Computer to conmunicate with other computers through regular telephone lines. The Atari 850 Interface Module enables the modem to work (interface) with your computer. The proliferation of low-cost modems—first the acoustical then the newer direct connect and, now the Atari 1400 and 1450 XL computers with built in modems—has literally brought the electronic information age into homes throughout the United Srares and abroad.

If you already own a modem and an Atari 850, the following terminal program, *MINIATRM*, will get you on-line today. You can communicate with Atari BBSs in just the time it takes to type in *MINIATRM* and dial one of the phone numbers we've provided.

MINIATRM was contributed by Mr. Jim Steinbrecher of the Michigan Atari Computer Enthusiasts (MACE). Jim is also the author of AMODEM, a more advanced terminal program, and co-author wirh Craig Chamberlain of AMISTERM, a terminal program designed specifically for communicating with Atari BBSs.

For Those with Disk Drives: "Open 'em Up"

Before you ger started—are you using a modem that requires the Atari 850 Interface Module? (The Atari 1030 Direct-Connect Modem does not require an Atari 850 Interface Module.) If your answer is "yes," you need special *RS-232* software to "open up" your interface unit for communications software.

Note: *Cassette users* don't need this software. The Atari 850 boots up as soon as you turn your Atari Computer on. Skip this section and go on to the *MINIATRM* program listing.

The RS-232 handler software you need comes with the purchase of the *Master Diskette, Atari Disk Operating System* (DOS 2.OS).

Step 1: Fonnat and write a DOS 2.OS file on a blank diskette.

Step 2: Label this diskette. It will become your "Communications Terminal" software package.

Step 3: With the BASIC cartridge plugged in your computer, type and enrer the program exactly as shown below. Check it carefully ro eliminate typing errors.

Step 4: Assign the name "AUTORUN.BAS" to this program and save it to your diskette.

Step 5: RUN the program. It will create an "AUTORUN.SYS" file on your disk.

How to Use Your New AUTORUN.SYS

1. Turn on your disk drive and wait until the "busy light" goes off.

2. Insert your "Communications Terminal" diskette.

3. Turn on both the modem *and* the Atari 850 Interface Module.

4. Turn on your Atari Computer. The AUTORUN.SYS software you've created will automatically open your interface module. You'll hear a tone from your TV speaker and the word "READY" will appear on your screen to tell you that your Atari 850 Inrerface Module is up and running.

WARNING: If you go to DOS to check your file, you'll have to repeat steps 1 through 4!

This article was adapted from Free Software for Your Atari Computer *by David and Dorothy Heller, copyright © 1983, Enrich/Ohaus, San Jose, CA, $8.95.*

Open 'em Up Program

```
20 OPEN #1,8,0,"D:AUTORUN.SYS"
30 FOR I=1 TO 88
40 READ D
50 PUT #1,D
60 NEXT I
70 CLOSE #1
80 END
100 DATA 255,255,0,56,75,56,169,80
110 DATA 141,0,3,169,1,141,1,3,169
120 DATA 63,141,2,3,169,64,141,3,3
130 DATA 169,5,141,6,3,141,5,3,169
140 DATA 0,141,4,3,141,9,3,141,10
150 DATA 3,141,11,3,169,12,141,8,3
160 DATA 32,89,228,16,1,96,162,11
170 DATA 189,0,5,157,0,3,202,16,247
180 DATA 32,89,228,48,6,32,6,5,108
190 DATA 12,0,96
200 DATA 226,2,227,2,0,56
```

MINIATRM

```
10 REM * MINIATRM.BAS:
   REM BY JIM STEINBRECHER,
   ARCADE BBS 313-978-8087
20 REM * For a new or 16K user to DOWNLOAD
   Terminal software or other programs.
30 REM * Run in present form when connected
   to BBS using ATASCII <EOL=155>
40 REM * To use on CIS or ASCII BBS change
   Line 70 XIO-38 to "XIO 38,#2,0,0, etc."
50 C=FRE(0)-100:DIM D$(15),BUFF$(C):
   BUFF$=" ":BUFF$(C)=" ":
   BUFF$(2,LEN(BUFF$))=BUFF$:? "BUFFER=";C
60 OPEN #1,4,0,"K:"
70 SAV=0:CLOSE #2:XIO 34,#2,192,0,"R:":XIO
   38,#2,32,0,"R:":OPEN #2,13,0,"R:":
   XIO 40,#2,0,0,"R:"
80 SETCOLOR 2,7,4:? "* TERMINAL MODE *"
90 STATUS #2,C:IF PEEK(747) THEN GET #2,C:
   ? CHR$(C);:IF SAV THEN POKE ADDR,C:
   ADDR=ADDR+1:GOTO 90
100 IF PEEK(764)<255 THEN GET #1,C:
    PUT #2,C:GOTO 90
110 IF PEEK(53279)=7 THEN GOTO 90
120 IF PEEK(53279)=6 AND SAV=0 THEN SAV=1:
    ADDR=ADR(BUFF$):SETCOLOR 2,2,4:? :
    ? "* SAVE ON *":GOTO 90
130 IF PEEK(53279)<>5 OR SAV=0 THEN 90
140 CLOSE #2:? :? "SAVE DEVICE";:INPUT D$:
    IF LEN(D$)=0 THEN 70:REM ENTER C,P,D:
    FILENAME.TYP
150 TRAP 140:OPEN #2,8,0,D$:TRAP 160:
    ? #2;BUFF$(1,ADDR-ADR(BUFF$)):
    GOTO 70
160 ? :? "BUFFER EMPTY":GOTO 70
```

MINIATRM: Your First Terminal Program

MINIATRM turns your computer into a communications terminal. Used with your modem, it enables you to communicate with distant computers to download any program, then add it to your diskette or cassette software library.

MINIATRM can also be used to download and save more sophisticated Terminal programs—you might only use it once, to capture and save AMISTERM for example, but it will get you on-line today! So, type in and SAVE this valuable *MINIATRM* program to your new "Communications Terminal" diskette or cassette.

Note: If you are using *MINIATRM* with a cassette and a 48K Atari Computer, change line 50 as follows:

From: "C=FRE(0)–100"
To : "C=FRE(0)–2200"

Now you have the software to begin your adventure into telecommunications: an Interface AUTORUN.SYS program and the *MINIATRM* terminal program!

How to Use MINIATRM and Get On-Line

First, make sure that both the AUTORUN.SYS program and MINIATRM are saved on the same diskette. Now follow the steps below:

Step 1: Turn on your disk drive, modem, and Atari 850 Interface Module.

Step 2: Insert the ATARI BASIC cartridge into your computer.

Step 3: Insert your "Communications Terminal" diskette into your disk drive and turn on your computer. Note: If you are using a cassette with a modem that does not require the Interface Module, simply type "CLOAD: MINIATRM" into your computer.

Step 4: Type RUN"D:MINIATRM" and press [RETURN].

Note: If you are using a cassette, turn on the Atari 850 Interface and your computer and type RUN"C:MINIATRM" and type [RETURN].

Your computer is now a *Communications Terminal*! You can begin accessing all the *FREE* software and services available to you! After you run *MINI-*

ATRM, you'll see "BUFFER=29213" appear on your monitor's screen. "BUFFER=29213" means that you can download and save programs that are as long as 29213 bytes (or letters). The screen also tells you that you are now in the *Terminal Mode* of operation.

Downloading

After you've established contact with a BBS and are ready to download a program, press the START key.

After pressing START, your screen will turn red and " *SAVE ON* " will appear to tell you that your computer's buffer memory is storing the program being sent to you by the BBS.

Storing Your Downloaded Program

Press the SELECT key when you are ready to save your newly acquired program. After you press SELECT, a "SAVE DEVICE?" prompt appears to ask if you want to store your program on diskette or cassette, and what you'd like to name it.

Cassette:

If you'd like to store your new program on cassette, answer the "SAVE DEVICE?" prompt by typing:

C:"FILENAME" and press [RETURN]

Now depress the RECORD and PLAY buttons on your Program Recorder, press [RETURN] on your computer, and your new program will be transferred automatically from your computer's buffer memory to the cassette tape.

Diskette:

To store your new program on diskette, answer the "SAVE DEVICE?" prompt by typing:

D:"FILENAME" and press [RETURN]

Your new program will be transferred automatically from your computer's buffer memory to your diskette.

That Was Easy!

That's all there is to it! If you'd like to use *MINIATRM* to get some free communications software, just look for abbreviations like "JTERM," "AMISTRM" and "JNSTRM" on a BBS list of available software. When you find one of these Terminal programs, simply download and store it on your diskette or cassette. Building a great software library is really easy!

Dial Up Some Free Software

As a special supplement for ATARI CONNECTION, we have included seven of the more than 500 sources for free software listed in our book, *Free Software for Your Atari Computer*. These seven phone numbers will give you a good introduction to the quality and variety of free software, electronic mail, and information services that are available to you.

And remember—this is only the beginning! With the *Free Software* book, you can also access the public domain software libraries of hundreds of users' groups by mail or modem. If you are a parent or a teacher, you can get in touch and on-line with special educational resources and software libraries, including services for the handicapped and learning disabled.

Here are the seven phone numbers that will get you started in telecommunicating:

M.A.C.E. (Michigan Atari Computer Enthusiasts) is an active, international users' group with three great electronic bulletin boards:

M.A.C.E.: 313-544-0885 (AMIS)
M.A.C.E. WEST: (AMIS)
313-274-3940
A.R.C.A.D.E.: (AMIS)
313-978-8087

These three BBSs offer you an impressive selection of amateur games, science programs, educational software, home management programs, computer utilities, and graphics demonstrations from their 30-disk library.

The people at M.A.C.E. are pioneers in computer modem communications. M.A.C.E. members, such as Jim Steinbrecher, created *MINIATRM* and A.M.*I.S.* (Atari Message and Information Service). This extensive program, which turns your computer into an electronic bulletin board, is available for downloading at selected times during the week. You must call the special phone number:

313-544-0885

It takes approximately one hour to receive the program's many sections. If you are interested, leave a message on one of their three bulletin boards to find out when the A.M.*I.S.* program is available for download.

* * *

A.C.E. (Atari Computer Enthusiasts) is a users' group based in Eugene, Oregon, which has thousands of active members worldwide. The group has a special interest in programming and educational software. They are also world-renowned for good arcade game software. A.C.E. has a 24-hour ARMUDIC bulletin board that you can dial at:

503-343-4362

* * *

If you would like to dial up a number in California, you can talk to a "homegrown" BBS, *Itsy Bitsy Bulletin Board*. Jeff Bell enjoyed networking with his Atari Computer so much that he decided to set up his own bulletin board for the weekend. After lots of discussions with SYSOPS (system operators) of other bulletin boards, Jeff went on-line. People called all weekend and Jeff has been hooked on networking ever since.

Jeff is happy to share his experiences with you about the challenges and rewards of starting your own bulletin board (see "How to Start Your Own Electronic Bulletin Board" in *Free Software for Your Atari Computer*). To communicate with *Itsy Bitsy Bulletin Board*, dial:

408-298-6930

* * *

The Downtown Washington, D.C. Atari Computer Users' Group has 20 diskettes full of public domain software with a special emphasis on business programs and telecommunications. The group developed the *ARMUDIC* Electronic Bulletin Board. The name "ARMUDIC" was derived by transposing the board's phone number into "ARM-UDIC."

This excellent Bulletin Board System, which offers a breakdown of software on file by category, is available commercially to those who want to start their own base station. For more information, dial (you guessed it):

202-276-8342 (ARM-UDIC)

Telecommunications opens up business opportunities for enterprising Atari Computer owners. The GRAFEX Bulletin Board in California hosts a mail-order business for Atari equipment and software. You can find out about prices and new products, place your orders electronically, or simply exchange software and information by dialing:

408-253-5216

Happy dialing! Enjoy your telecommunications adventure! ■

KIDBITS

by Myrna Rae Johnson

FIND THE BUG WINNER

SOLICITOUS SALUTATIONS, sovereign insecta seekers! I am the glorious Ms. Mella Dramatick, and I am here to tell you the sad news about my pugnacious protégé, Lord Motley Bugnut. While marauding marvelously through the jubilant jungles of Pago Pago, on the prowl for the coveted Creeping Ceediwackwack, his mind wandered off, and then so did he. It took weeks to locate him in that dreadfully deadly dark forest. When we finally found him, we discovered that he had suffered a severe lack-of-sun stroke. We had to rush him to the nearest native resort, where he is now sequestered, sunning himself to solid salubrity. Meanwhile, I will be officially offering my sporting services in the hunt for bodacious bugs, being an avid arachnid enthusiast.

Now then. This contest was even more excrutiatingly exacting than the last one. It seems that something sat on the "perfect circle," causing a summarily squashed specter. I believe that the species involved in this vicious violation of perfect programming is known as *Menacingii menconii*, a two-footed species of the Computerus Creepus family.

Illustrated by Ward Schumaker

According to some of our confused contestants, the circumference of the circle was correct as programmed. Little did they know, the piquant pixels definitively dance in different dialects, depending on the television set used. For those of you with perfect pixels, line 50 had to be changed. When working perfectly, the program calculates the coordinate of the vibrant vertex. It does this by masterfully multiplying the length of tbe legitimate line (L) by the SIN of the angelic angle (AN) for Y, and by the cosmic COS of the angle for excellent X. The bug had eaten away at the COS of the angle, so it was not being multiplied by languishing L. This left the figure limply long, rather than round.

Sly Stanley Halbert had the wondrously winning answer for tbe fastest fixing of our flat-footed bug, and here's his short but sweet letter:

ATARI CONNECTION,

My name is Stanley Halbert, and I live in Lawrence, Kansas. Our family has had an Atari 800 Computer for three years. My favorite hobby is programming.

To make a perfect circle, I added an extra line 55:

Perfect Circle

```
10 GRAPHICS 8:POKE 710,0:COLOR  3:
   POKE 752,1
20 TRANX=159:TRANY=79:SIZE=50
30 ? CHR$(125):? :? "
         THE":? "
         CIRCLE?"
40 FOR THETA=0 TO 179
50 Y=SIN(THETA)*SIZE*0.9:X=COS
   (THETA)*SIZE
55 Y=Y*0.95
60 PLOT X+TRANX,Y+TRANY:PLOT
   X+TRANX,TRANY-Y:PLOT TRANX-X,
   Y+TRANY:PLOT TRANX-X,TRANY-Y
70 NEXT THETA
80 GOTO 80
```

To make the circle draw faster, rewrite line 60, keep line 50, but repeat it as line 35, then add a new line 36 as follows in the new program below:

Fast

```
10 GRAPHICS 8:POKE 710,0:COLOR 3:
   POKE 752,1
20 TRANX=159:TRANY=79:SIZE=50
30 ? CHR$(125):? :? "
         THE":? "
         CIRCLE?"
35 Y=SIN(THETA)*SIZE*0.9:X=COS
   (THETA)*SIZE
36 PLOT X+159,Y+79
40 FOR THETA=0 TO 179
50 Y=SIN(THETA)*SIZE*0.9:X=COS
   (THETA)*SIZE
55 Y=Y*0.95
60 DRAWTO X+159,Y+79
70 NEXT THETA
80 GOTO 80
```

Bravo, Stanley. You will rapidly receive your new wits wrangling game, *Getaway*, by tbe Atari Program Exchange (APX).

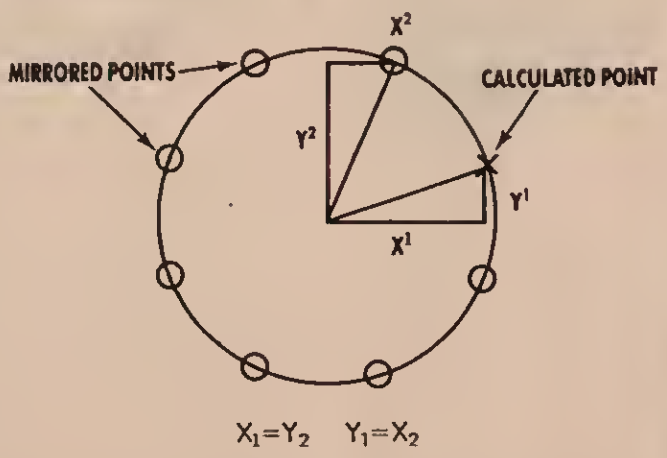
BONUS!

To miraculously make our flat-footed bug into a fleet-footed bug, Peter Ghiloni of Granville, Ohio, sent a correct collection of calculus, and an excellent explanation:

THE BUG,

The key in speeding up the program is in lines 40 and 65. In line 40, the "FOR" loop ranges from 0 degrees to only 45 degrees. All points on the circle can be derived from Y and X values calculated from 0 to 45 degrees. This is probably best explained by the following diagram.

Once a value for X and Y is calculated in the 0 to 45 degree range, transposing these values will provide the necessary data for the 45 to 90 degree range. As Mr. Menconi mirrored his calculated points to derive the bottom half of the circle, line 65 in my program mirrars points to derive the remainder of the circle.



Line 75 has been added to help in measuring the time to draw a circle. Time to draw was measured from the time "RUN" was entered to the time "DONE" appeared.

Following are the results:
Original Program: 37.3 seconds
Submitted Program: 11.1 seconds

We are a two Atari Computer family. My Atari 800, with a host of peripherals, is tied into a computer network associated with my professional activities. My wife's Atari 400 spends the school year in the fourth grade classroom where she teaches. My wife wrote and used a 16-lesson unit to teach Computer and Pragramming Literacy to the fourth grade this past school year. I find video games most relaxing, and I delight in the cleverness of graphics, sound and strategies appearing in many of the new games and adventures.

Sincerely,
PETER A. GHILONI

Faster

```
10 GRAPHICS 8:POKE 710,0:COLOR 3
   :POKE 752,1
20 TRANX=159:TRANY=79:SIZE=50
30 ? CHR$(125):? :? "THE":?
   "CIRCLE?"
40 FOR THETA=0 TO 45
50 Y=SIN(THETA)*SIZE:X=COS
   (THETA)*SIZE
60 PLOT TRANX+X,TRANY-Y:PLOT
   TRANX+X,TRANY+Y:PLOT TRANX-X,
   TRANY+Y:PLOT TRANX-X,TRANY-Y
65 PLOT TRANX+Y,TRANY-X:PLOT
   TRANX+Y,TRANY+X:PLOT TRANX-Y,
   TRANY+X:PLOT TRANX-Y,TRANY-X
70 NEXT THETA
75 ? "OONE"
80 GOTO 80
```

Well, precious Peter, for your diligent and brilliant efforts on behalf of our confused camaraderie of slightly buggy bug finders, you will be rewarded with a positively perplexing new game from APX —delightfully delectable *Dandy* (reviewed this issue).

And, for the grand finale, with thanks to all of the slightly slower programs sent in with inventive ideas, our little ol' messmaker Menconi spliced together the best of these programs and came up with a spectacularly speedy *5.5* second circle:

Fastest

```
10 GRAPHICS 8:POKE 710,0:COLOR 3:
   POKE 752,1
12 OIM CH$(1)
15 RATIO=1.1:INVRATIO=1/RATIO:
   TWOPI=6.28318531:NPTS=360
20 TRANX=159:TRANY=79:SIZE=50/
   RATIO:OTHETAX=RATIO*TWOPI/NPTS
   :OTHETAY=INVRATIO*TWOPI/NPTS
30 ? CHR$(125):? :?
31 ? "           THE"
32 ? "         CIRCLE?"
35 Y=0:X=SIZE
40 FOR I=1 TO 93
50 PLOT X+TRANX,Y+TRANY:PLOT
   TRANX+X,TRANY-Y:PLOT TRANX-X,
   TRANY+Y:PLOT TRANX-X,TRANY-Y
60 X=X-Y*OTHETAX:Y=Y+X*OTHETAY
70 NEXT I
80 GOTO 80
```

FIND THE BUG

OUR ARCANE ARACHNID, Spiff Spider, wove a wonderful web, replete with a proliferation of polygons—which are flamboyant figures with three or more sides. But that horrible heathen, Malingering Menconi, sent his bonsai bug, Buford, to sew up the sides of Spiff's web. How rude! Worse, it seems that Buford is some sort of math whiz and used a tirade of triangulation to tangle poor Spiff's territory. Can you help calculate the web's release?

SPIFF'S WEB

by Dave Menconi

```
10 DEG :TX=159:TY=95:GRAPHICS 24:
   SETCOLOR 2,0,0:AOJ=0.9
20 FOR SIOES=3 TO 22:L=30
30 X=TX-L/2:Y=1:GOSUB 50:NEXT SIOES
40 GOTO 40
50 PLOT X,Y:OAN=360/SIOES:AN=0
60 FOR I=1 TO SIOES:X=X+COS(AN):
   Y=Y+AOJ*L*SIN(AN):ORAWTO
   X,Y:AN=AN+OAN:NEXT I:RETURN
```

FIND THE BUG CONTEST

If you can Find The Bug and fix Spiff's web, then send us your corrected program along with a short story about yourself. We're offering a reward of the new Atari Home Computer game, *Eastern Front (1941)*, for the first right answer that undoes the damage.

If you're the winner, we'll print your story along with your corrected program in the next issue of ATARI CONNECTION.

Please send your entry to:

Find The Bug
c/o ATARI CONNECTION
P.O. Box 50047
San Jose, CA 95150

Atari Computer Weathers West Virginia

High School Senior Airs Forecasts Based on His Own Atmosphere Theories

Lee Smith can predict your weather.

by Teddi Converse

"TODAY WE'LL SEE mostly sunny skies with a few scattered clouds," reports Lee Smith, a 17-year-old high school senior and president of his class in Williamstown, West Virginia. "Temperatures will range from a high of 75 to a record low of 51 for this time of year. This low temperature is the result of that cold Canadian air from the high-pressure system we've been seeing."

Four times a day his weather report—aided by a forecasting system he designed on his Atari Computer—is aired on WMOA AM and FM radio stations. He also does a daily weather column for the *Marietta Times* newspaper, and reviews home computers for a weekly computer program on WMOA.

Lee has been interested in meteorology for several years. After reading about a BASIC program designed to predict the weather, Lee decided he would expand on the program and create one of his own.

Using national weather maps, weather radio and his Atari 400 Home Computer, Lee designed GRASP (Graphics-Related Atmospheric Synopsis Program). It includes all of the latest theories of jet-stream and frontal movements along with his own unique "checkpoint city" theory. It takes into consideration conditions in surrounding cities as well as in the immediate forecast area. Through his own testing he found the system not only better than local newspaper, television, and radio forecasts, but 20% better than National Weather Service predictions. He claims 96% accuracy for forecasts up to 12 hours and up to 90% for forecasts up to 36 hours in advance.

"My program makes something very complicated like weather forecasting easy," says Lee. "Basically, you just input information about four different cities surrounding your area. Then you input local conditions—cloudiness, humidity, barometric pressure and the like. You can gain information from a weather radio [a device sold for about $10.00 which carries a high-band frequency, 24-hour weather report from the National Weather Service] or from the local papers about the closest weather system. After you put all the information in your computer, it spits out a weather report."

Lee plans to submit GRASP to the Atari Program Exchange (APX) in the near future. "I'm changing it a little to use the OPTION and SELECT keys right now," he says. "I read the section in *De Re Atari* about human engineering and it inspired me to add some more components to the program. I'm also working on putting more graphics into the program."

Lee would like to create software programs that would make long-range weather predictions for farmers. "I don't know too much about agriculture right now, but I'm learning," he says.

After Lee graduates from high school he intends to go to college and, of course, study meteorology. Who knows? Someday, all of us may be able to use Lee's software weather programs to plan our weekend skiing or camping trips.

ATARI CONNECTION™
THE HOME COMPUTER MAGAZINE

DELIVERED TO YOU AT HOME

Subscribe Now And Save 50¢ Off The $3.00 Cover Price

Every issue of ATARI CONNECTION is filled to the brim with fun-filled, educational, practical articles, features, cartoons, and hints to help you and your entire family get more and more fun and work out of your ATARI Home Computer.

☐ **YES,** enter my subscription for one year (4 issues) for ONLY $10.00

☐ Payment Enclosed ☐ Bill Me Later

☐ MasterCard ☐ VISA

Acct. No. _____

Expiration Date _____

Signature _____

Regular subscription rate—$10 for 1 year. 3BF3

Name _____

Address _____

City _____

State _____ Zip _____

This rate limited to USA and its Possessions.
For foreign subscriptions, add $5 for postage and handling.

Subscription Department
ATARI CONNECTION
P.O. Box 50047
60 E. Plumeria
San Jose, CA 95150

ATARI®
A Warner Communications Company

ADD A WHOLE NEW WORLD OF SPECIAL PRODUCTS

This 70-page catalog offers just about everything you could desire for your ATARI Home Computer system—exciting new games, educational software for the classroom and the home, computer graphics and business programs for the home office. Plus, hardware, accessories and publications that add more power, more conveniences and more versatility to your ATARI Home Computer.

Order your ATARI Special Additions now and add a whole new world of special products just for your ATARI Home Computer.

To order see back of card.

*Plus 55¢ postage and handling.

ONLY $4.95*

Order the latest ATARI® Special Additions™ Catalog of fascinating new programs and products created by independent manufacturers especially for ATARI Home Computers.

ATARI
THE HOME COMPUTER MAGAZINE
CONNECTION™

Subscriber Benefits & Privileges

As a Subscriber to ATARI CONNECTION you are GUARANTEED a subscription price far below the cover price.

As a Subscriber to ATARI CONNECTION, you are GUARANTEED the privilege of immediate cancellation for any reason. [Or for no reason at all.] You will receive a FULL REFUND — by return mail — for the undelivered portion of your subscription.

☐ **YES! I want to order my copy of the ATARI Special Additions catalog. Here's my name and address and my check or money order for $5.50 ($4.95 for ATARI Special Additions plus 55¢ for postage and handling).**

ATARI® SPECIAL ADDITIONS™

Name _____

Address _____

City _____ State _____ Zip _____

Make your check or money order payable to: Atari, Inc.
Place your payment and this card into an envelope and mail to the address on the right.

ATARI SPECIAL ADDITIONS
P.O. Box 50047
60 E. Plumeria
San Jose,
CA 95150

Please allow 4 to 6 weeks for delivery.

ATARI® DOES MORE THAN ANYONE ELSE TO GIVE YOU TWO HELPFUL KINDS OF COMPUTER SERVICE.

LOCAL. LONG DISTANCE.

If you need someone to fix your ATARI® Video Game or Home Computer, you'll find the best place is also the closest.

We have over 1,600 ATARI SERVICE℠ Centers coast to coast; just look in the Yellow Pages under Video Games or Computers.

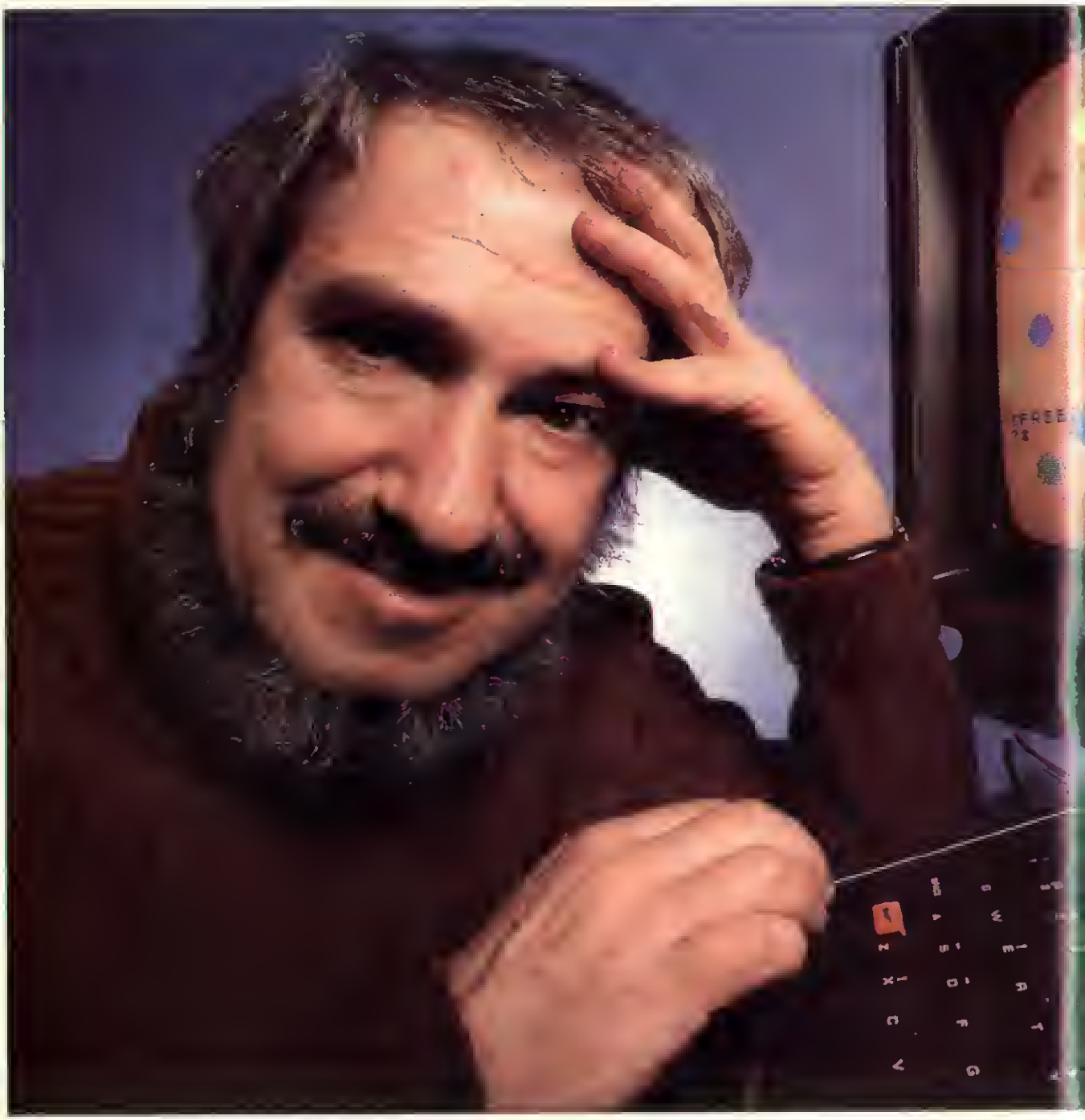
And if you have any kind of question about your ATARI Home Computer—how to do something new with it, how to debug one of your own programs, what kind of peripherals are best—call the ATARI Help Line and talk to an ATARI expert. Our toll-free number is 1-800-538-8543.*

At ATARI SERVICE, we take care of you. As well as your ATARI system.

ATARI SERVICE
FACTORY AUTHORIZED NETWORK

WE ANSWER YOUR CALL FOR HELP.

*California: 1-800-672-1404



SEYMOUR PAPERT—*technological seer, outspoken social and educational critic, self-styled computer poet—exerts a profound influence in the microcomputer field. One journalist recently called him "the spiritual father of the Logo movement," and referred to his book,* Mindstorms (Basic Books, New York, 1980), *as "the movement's bible." He is best known for inventing Logo, a versatile computer language that provides a friendly introduction to programming, a serious tool for advanced programmers, and a medium for educational discovery.*

Papert says his goal was to create a language "simple enough so a five-year-old could write a program in the first few minutes of contact with a computer, and sophisticated enough so a computer scientist would find the system congenial and rich."

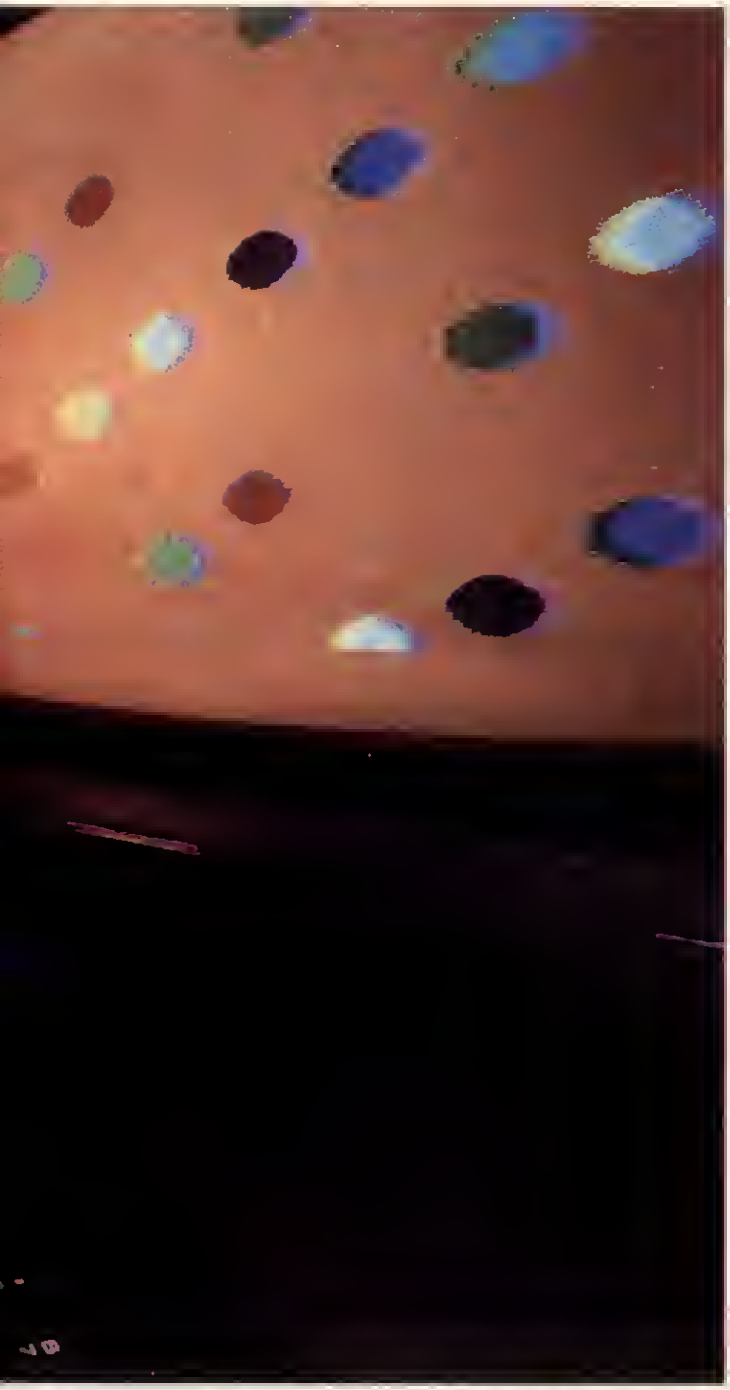
The resulting language might have been created in Papert's own image. It is friendly, accessible and as complex as you want to make it. But, like any language, Logo is difficult to describe in a few words. "Because it can take on a thousand forms and can serve a thousand functions," Papert says, "it can appeal to a thousand tastes."

Papert's unaffected clothing and meditative air bespeak his position as MIT professor and programming guru for the current generation of computer educators. In personal conversation, Papert is reflective and almost otherworldly, but charming. He delivers carefully considered statements in a South African accent, a relic of his Johannesburg upbringing, tempered by years of study at Cambridge and the University of Paris.

Mathematical studies and years of work with renowned child psychologist Jean Piaget in Geneva prepared Papert for his work in the development of educational computing.

"Personal computers will soon become the most important medium for learning," Papert says, "because of the ways they affect how people think and learn." He suggests that computers are like pencils; you need more than one or two per class. He estimates that, at current prices, every student could have a personal computer for approximately five percent of the cost of

PHOTOGRAPHED BY TED HOROWITZ



Learning to LEARN *with the* FATHER OF LOGO

Seymour Papert on Education and Language

by

Carlos Vidal Greth

each student's total education.

"I dream of using this technology not to improve the schools we have always known, but to replace them with something better," Papert says. "It will be like the growth of a new culture in which computers will be so integrated into ways to think about ourselves and the matters we learn that the nature of learning itself will be transformed."

On a recent visit to Atari corporate headquarters in Sunnyvale, California, Logo inventor Seymour Papert shared his views about Logo and its revolutionary implications for our schools and society.

CONNECTION: Did you expect to develop such a large, devoted following when *Mindstorms* was published?

PAPERT: I wrote *Mindstorms* as a criticism of our educational system. I've been incredibly, pleasantly surprised by the number and diversity of people who have felt something personal in the book. People have found in it a validation of themselves in the face of an educational system that is a virtual put down. I wrote what people don't want to admit —even to themselves—about how they feel about our schools.

CONNECTION: In *Mindstorms* you compare Logo environments to Brazilian samba schools (social clubs whose members prepare music, dances, and skits for the annual parade during Carnival in Rio De Janeiro). Why do you make this analogy?

PAPERT: I've done the samba in Brazilian samba schools. Samba schools provide a context where people enjoy themselves while learning, and where experts and beginners interact. This never happens in American schools, where learners are segregated by levels of ability in classrooms. Samba schools are in the culture, a part of people's lives.

CONNECTION: Will there be a role for teachers in the future?

"With Logo, I use computers the way a poet uses words."

PAPERT: Absolutely. When people claim that my vision of education does away with teachers, I get very upset. That turns on its head what I've been trying to say. Schools give teachers vety little opportunity to teach. They spend most of their time brainwashing or forcing children to do rote activities nobody believes in. In the kind of learning environment I envision, teachers can really teach.

CONNECTION: Can schools be transformed to accommodate the style of learning that you and others propose?
PAPERT: Twenty years from now, there will be very different learning environments involving computers. Schools will successfully transform themselves into these new environments or they'll die out like the dinosaurs.

CONNECTION: That's a disturbing concept, in part because so many people are involved in education in some way. Is it a case of program or perish?
PAPERT: The role of learning professionals in society is going to grow. Rapid technological changes mean that people will have to or want to do more learning. Educators will have to adapt to new contexts. If they can't teach themselves to teach in new contexts, they are pretty lousy teachers, and should be doing something else.

CONNECTION: In a symposium several years ago, you said that if you had $10 million, you would spend it on creating "conditions for the emergence of computer poets." What are computer poets?
PAPERT: A poet is someone who establishes a relationship with his readers that's outside the logical, discursive relationship we usually classify as science. With Logo, the computer can be used similarly, to touch on the deeper, non-logical dimensions of self and the personal aesthetic. As a programmer, I use the computer in the same way a poet uses words, to touch on intimate and individual aspects of life.

CONNECTION: Is BASIC bad for you?
PAPERT: BASIC is a bad language. BASIC exists for the same reason we use the QWERTY [the top row of alphabetic keys on a standard typewriter] keyboard. It was created for historical reasons. The only reason it continues to exist is that lots of people know it, and lots of programs have been written in it. If BASIC were wiped out by some kind of cosmic chemical, I'm quite sure it wouldn't be reinvented.

Logo will replace BASIC altogether or in a very large proportion of homes. Certainly, Logo will be the language home computer owners use for pleasurable programming. There are very few projects that most home computer owners are likely to want to do that couldn't be done better on Logo than on other available languages.

CONNECTION: Most Logo applications are geared toward youngsters. How can an adult use Logo?

PAPERT: We've created a wrong impression by de-emphasizing the value of Logo for adults. Logo is a particularly good introduction for adults who want to learn to program. Experience shows that it's easier to convert from Logo to other programming languages that it is to learn those languages directly. Logo is also good for business and record-keeping applications. Logo is being pushed by a number of people as the small business applications language of the future.

CONNECTION: Why are so many people surprised afrer experiencing Logo to find that computers can be so accessible?
PAPERT: The easy answer is that many people grew up thinking of themselves as nontechnological.

There are deeper explanations having to do with the

"A child might use dynaturtles to create pleasing patterns, but he applies the same laws that govern the paths of planets. Call it participatory physics."

"If BASIC were wiped out, it would not be reinvented."

subconscious. There are many fears coming to light in terms of our relationship to computers. These are fears about whether we are self-determining, free creatures, or whether we are machines. The computer forces us to face some very basic elements of human nature. Any confrontation with the depths of what one is has to be frightening.

CONNECTION: Why are the turtles valuable as learning tools?
PAPERT: Graphics is one of the easiest ways into the world of programming. If you are a beginner, your first program should probably be one that draws something on the video screen. If you're an advanced programmer and want to learn a new style of programming, getting the computer to draw something for you is a good way to start. Graphics programs let you see immediately what your program is doing.

CONNECTION: How is turtle geometry different from other types of geometry?
PAPERT: It's process-oriented or dynamic geometry. In Cartesian geometry, one thinks of a circle as the set of points that makes a certain equation come true. That equation defines the points. In turtle geometry, you define a circle by thinking about how the turtle moves on the circle. The turtle is based on knowledge that most people have: how to physically move around in the world. The main thing is the process of making the circle rather than the properties of the finished circle.

CONNECTION: What purpose do dynaturtles serve?
PAPERT: Dynaturtles are turtles whose laws of motion are like those of physics. Using these mobile turtles gives the user a chance to learn physics by actually playing with fundamental concepts. A child might use dynaturtles because he likes to create pleasing patterns on the screen. In order to create those pleasing patterns, however, he must apply the same laws that govern the paths of planets circling the sun. You might call it participatory physics.

CONNECTION: How can the animation capabilities of Atari Logo be used?
PAPERT: The animation in Atari Logo concretizes the idea of motion for learners. Motion is traditionally one of the hardest concepts to learn in science. Paradoxically, knowledge of how to move around in the world is learned very early, but the study of motion in science is delayed until advanced levels of schooling. With Logo, very young children learn to work with concepts of motion, which places it in its rightful fundamental place in early education.

CONNECTION: What is list processing, and why does it make Logo so special?
PAPERT: In Logo, a program is a list of instructions, or a list of lists. Each instruction is a list of Logo words. If you're creating a program to draw a square, for example, you type *FORWARD 100, RIGHT 90* four times. *FORWARD 100* is a list of two words in a special order. The relationship between the program and its instructions is the same as that between each instruction and the words that make it up.

When "new math" came out in the 1960s, many people learned for the first time the concept of *set*. A list is like a set, with one addition—it involves process-oriented set theory. The set of objects in a list is organized in a definite way so that you come to each object in a special order.

CONNECTION: What is recursion? Why is it important to Logo?
PAPERT: Recursion is a process that refers to itself. Since it refers to itself, it can refer to itself referring to itself.

I'll make an analogy. One of the most important things about thinking is that you can think about thinking. You can even think about thinking about thinking. Similarly, Logo has programs that act on programs, and programs that act on programs that act on programs.

CONNECTION: Why is recursion powerful?
PAPERT: Because it is infinite. When the way a higher level acts upon a lower level is also the way a yet higher level acts on the first level, then there are no limits how far you can expand a program.

In the history of philosophy, the problem of self-reference has always come up. Consider the Greek invention of the paradox of the liar from Crete. For argument's sake, say that Cretans always lie. But if a Cretan says "I'm lying," wouldn't he be telling the truth? It's always puzzling when someone or something refers to itself.

Recursion offers a solution to the puzzle. It gives you a way of thinking about things referring to themselves.

CONNECTION: You said that computers will help bridge the gap between technical-scientific and humanistic cultures, by putting "powerful ideas in computational form, ideas as important to the poet as to the engineer." Will the bridge be built by those on the technical-scientific side?
PAPERT: Undoubtedly that is where the initiative is going to come from, but that's not where it *has* to come from. One of my most important tasks is opening doors for people other than those who are taking the lead now. I want to see more people get involved with computers and help shape the culture of the future. The power of computers is presently in the hands of a small minority—largely white, male, and Californian. A tiny subculture is shaping the computer culture for us all. It's scary in terms of developing social inequalities. ■

Carlos Vidal Greth is a Senior Writer in The Atari Products Company.

IN DEFENSE OF BASIC

by Carlos Vidal Greth

SEYMOUR PAPERT'S ENTHUSIASM for his language, Logo, is shared hy many programmers and educators. There is, however, some controversy about specific aspects of Logo and Papert's strongly held beliefs about BASIC. The contrasting views of Arthur Luehrmann, another major influence in educational computing, are presented below.

Luehrmann is co-founder of Bekeley-based Computer Literacy, a three-person partnership designing and writing curriculum materials for beginning computer users. Formerly a director of the Computer Education Project at the Lawrence Hall of Science in Berkeley, Luehrmann is co-author of the book, *Computer Literacy: A Hands-On Approach*.

Luehrmann shared rhe following observarions with the ATARI CONNECTION in an interview at his home in the hills overlooking Berkeley.

"Covered with Warts, But It's There"

Whether you like it or not, millions of home computers speak BASIC. Only a tiny percentage speak other languages. BASIC is covered with warts, but it's there. Current versions of BASIC on microcomputers are overly complex, tasteless, and full of inconsistencies. There's no easy way to define a procedure and use local variables and pass parameters. BASIC doesn't have tight control structures; you have to huild them out of pieces. It docsn't have a loop structure or a multi-statement branch structure.

Despite the language's drawbacks, however, one can learn to read, write, and do structured and top-down programming with garden-variety BASIC.

If the educator teaches a language besides BASIC, will the kid get any practice on it outside of school? Not likely. If schools teach languages that kids only have access to in schools, kids won't learn much.

If BASIC isn't taught well in schools, the children learn street BASIC. Street BASIC is an unstructured, hacker approach where programs are built like houses of cards. You keep adding statements to a wobhly structure—a program—until it collapses into a heap.

Logo—the Pluses and Minuses

Current versions of Logo for schools are pretty good, considering that they're shoehorned into a pretty tight box [a microcomputer]. Logo's primary advantage is that it teaches how to break a problem down into manageable pieces and how to deal with rhe whole problem in an absrract way. This is called the "top-down" method in computer science. The top-down method doesn't come naturally; it needs to be taught. There's nothing magical about any language in that regard. It takes a teacher to do the job.

It's hard to teach structured programming in Logo because there's no conditional loop for repetitive functions, and branch construction is limited. A language designer should provide a simple method of telling the compurer ro do somerhing repeatedly until some condition hecomes true. That can he done in Logo, but only by a recursive procedure. It's easier to teach beginners repetitive procedures than it is to teach them recursion.

Branches are a problem as well. The data that you type until you press RETURN is the logical line in Logo. That's constraining when the branch structure you need to do something is rather lengthy. It forces programmers ro make two choice blocks into separate procedures and give each one a name.

That decision was made in part to protecr users from errors that can propagate from multi-line statements. If you close off each line by pressing RETURN, when you do error testing you've got a known world where mistakes can be flagged. The unfortunate effect is to make lengthy if/then constructions difficult.

The Transcendent Issues

We shouldn't ger bogged down in details of languages when there are transcendent issues like learning how to solve a problem with the top-down approach and how to use control structures. These are fundamental issues, and it doesn't matter a damn what language you teach them in.

If you know any programming language, and have heen correctly taught about procedures, control structures, dara and variables, and lists, learning a new language should be a piece of cake. ■

PHOTOGRAPHED BY JEFFERY NEWBURY

LOGO vs. BASIC
WHICH LANGUAGE IS BEST?

A Comparison Reveals The Real Question: What Do You Want To Do With Your Computer?

FOR THE PAST NINE YEARS, we've all witnessed how computers have changed our daily lives. But one thing will never change: Whenever a new computer language is introduced, its advocates will begin challenging the other languages. Inevitably, battle lines are drawn and a furor erupts.

The current furor has to do with the differences between Logo and BASIC. Logo advocates extol its sophisticated learning philosophy, contrasting Logo's educational applications to BASIC's pedestrian applications. BASIC's champions, on the other hand, point to BASIC's "tradition"—its vast collection of software and universal appeal, not to mention its ability to efficiently program a complete microcomputer system of printers, disk drives, and other peripherals.

When microcomputers were first introduced to the public, people had few languages from which to choose. BASIC was the predominant language available because it was so easily adaptable to small computers. Choosing a language wasn't an issue. But now we are faced with having

ILLUSTRATED BY J.T. MORROW

to make decisions which are often based on very little understanding or experience. Novice owners of home computer sysrems, individuals and educators alike, are likely to have questions such as: Which programming language is best? Which language can do the most? Do I need more than one language? Which language is the easiest to learn? Which is the easiest to use?

In the case of Logo and BASIC, there are no simple answers to these questions. Each programming language was designed to fill particular, and often different, needs.

For most home computer applications, many different programming languages can be used to accomplish the same goal. However, different languages can make it easier or harder to accompish some goals depending on the task ar hand. The *task environment*—what you want to do with your computer—is therefore an important consideration when choosing a language. Some of the most common and interesting tasks you can perform on the Atari Computer include:

- Graphic Designs—creating color drawings and designs on the computer screen or for color plorting and printing.
- Animation—moving shapes, figures or objects on rhe computer screen.
- Music—using computer-generated sounds to create melodies or sound effects.
- Data Processing—the manipulation, storage and retrieval of information to and from the computer and its peripherals.

The Task Environment

Graphics

ONE OF THE MORE endearing features of Logo is its "Turtle Graphics." Logo's structure evolved from *LISP*, a powerful list processing language used by computer scientists and researchers in the field of Artificial Intelligence. When Logo was first designed by Seymour Papert there were no graphics. The first experiments with children involved processing lists of words. Although Papert's initial experiments were successful, he felt that younger children would find programming easier to comprehend if they could actually *see* their program at work.

The "turtle" was originally a *mechanical* robot with a plastic domed top; its appearance resembled a turtle, hence its Logo namesake. The turtle has since evolved into four computerized turtle-shaped cursors residing in Atari Logo. To program the turtles, you simply tell them where to go.

This graphics system is known as *relative geometry*, or "turtle geometry." The

Logo

analogy of giving directions helps illustrate this concept. If a stranger were to ask you how to get to a certain gas station in your town, you could give him directions in one of two ways: (1) You could tell him the gas station is on the corner of 3rd St. and Main St.; or (2) you could tell him to go down three blocks, then turn left onto Elm St. until he reaches 3rd St; turn left onto 3rd St. and drive three blocks until he arrives at the corner of 3rd and Main.

The second example is an illustration of relative geometry. The first example illustrates *Cartesion Coordinates*. In Logo, you have access to both types of graphics systems; thus, in a sense, the programmer is placed on the screen along with the turtles.

The BASIC graphics system uses *Cartesian Coordinates* to set up a "pixel grid" of X and Y coordinares thar plot each graphic point.

Although its graphics are not nearly as sophisticated as Logo's "Turtle Graphics," BASIC offers a broad range of graphics capabilities that include multiple colors and a diverse selection of graphics resolutions. With ATARI BASIC, you can access all 20 Graphics Modes of rhe Arari Compurer (26 with the new Atari XL Home Computer series). Logo's Turtle has only three —Graphics Mode 0 and 7 (full-sceen) and Graphics Mode 7 (split-screen).

Animation

One of the features that sets the Atari Computers apart from other home computers is their built-in player/missile graphics animation system. This feature allows programmers to move shapes quickly and smoothly across the screen. Although player/missile graphics are not fully supported in ATARI BASIC, you can use simple machine language subrourines to access this animation system. The experienced ATARI BASIC programmer can create entertaining arcade-style game animation using player/missile graphics and character set animation.

In Logo, the turtles are players. You can redefine the four turtle "players" on the screen with four different shapes and colors. You can move rhese shapes smoothly—even rapidly change them for animation. Logo is the only Atari language that directly accesses player/missile graphics.

In addition to moving the Logo turtles around, we can tell a turtle to start moving in a direction at a cerrain speed and it will keep moving until it's told to stop. This becomes a very useful and powerful programming feature when combined with the use of "WHEN demons."

Using the WHEN demon feature allows us to have the program execute special instructions WHEN the turtle hits something. An easy way to picture this is to imagine the *WHEN command* calling on a WHEN demon, who watches for a condition to be met and then performs the operation. Once the WHEN demon has its instructions, the program can relax and continue without worrying, confident that someone's "on the job" watching for collisions or conditions to arise. In BASIC, the program has to be either continually on guard or must periodically check for collisions.

Music

Logo has two distinct tones that can be simultaneously turned on, with control over their frequency, volume and duration. With Logo, you can create simple melodies and game sound effects.

BASIC offers four distinct tones, with control over their frequency, volume, and even a distortion factor for creating sound effects. The skillful programmer can create four-part harmonies, from those used in barbershop quartets to New Wave rock, not to mention the complete complement of standard arcade game sound effects.

Data Processing

In most computer languages, data is represented by single numbers, or string arrays of characters. In Logo, however, we have *numbers*, *words* and *lists*. A word consists of characters, while a list is a group of numbers, words, or lists separated by spaces.

Usually, we think of *text* as groups of words, sentences and paragraphs. Logo mimics our concept of "text" which makes the handling of words very simple and natural. On the other hand, if we want to break a word down to its individual characters, we can treat a word much

THE ATARI
GIFT CATALOG

ATARI®
GIFTS FOR THE ENTIRE FAMILY

(See Page 44 and 45)

like a list.

Lists are also useful for "normal" applications. Obviously, a mailing list would be pretty simple, but what about a word processor that "reads" entire words instead of individual characters? Anything that can be handled using strings and arrays can be handled using a list processing routine. "Thought processing" designed for human interaction is much more efficient in Logo than in a language using data arranged in strings.

On the other hand, BASIC is well suited for programming an entire computer system to process data—retrieving the data from a disk drive, updating it, printing it, and putting it back on diskette. Atari Logo would also be a great language for true data processing applications because of its list processing capabilities, but you can't easily get the data to and from the storage and printing devices. When Atari Logo was designed, I/O (Input/Output) simply wasn't part of its repertoire, because of the excessive amount of memory required for system programming.

The Programming Environment

Another consideration when choosing a language is what we'll call the *programming environment*. Different languages often require a different user approach, even when the task is the same. Most programming languages are constructed in a way that makes them very difficult to use if you try to program in a style or approach that conflicts with the way the language is designed. However, you can if you understand what a language is good at doing and capitalize on its strengths. The following are examples of programming environments:

- Interactive—allowing the user to sit down at a terminal, "talk to the computer," and receive immediate responses.
- Modular—each task can be written expressing as an individual piece of a program that can later be used as a building block in larger programs.
- Extensible—creating new programming commands that become part of the langauge and, in effect, make the language grow.
- Recursive—allowing people to define a task or concept in terms of itself.

Logo Programming

Logo was conceived and designed as a computer langauge that could help children learn how to structure their thinking and discover new thought processes. A child learning to program in Logo also becomes familiar with basic problem solving techniques as well. As a result, Logo's strongest point is its learning structure.

BASIC

The research and development work in Artificial Intelligence, from which LISP grew, has revealed how some of our human thought processes and "belief systems" are structured. Logo has inherited this tradition and structure, and therefore, more closely approximates the structure of our natural human thinking.

This capability makes Logo a truly *extensible* language, which means it can "learn" new commands in ways that are similar to how people learn. In most languages we have a fixed set of commands. In Logo we start with a fixed set of commands (called primitives) but we can write our own commands as well. This feature creates a rich environment for exploring how people think and solve problems. Logo programmers will also find the language to be extremely interactive, allowing ideas to be created and immediately tested and changed if so desired. Each task can be solved independently and later incorporated into a larger program. This is a result of the modular structure of the language.

One of Logo's most powerful features is its *recursive* problem-solving capability. Recursion provides you with a programming tool that not only helps you solve problems in a program, but teaches you a sophisticated problem-solving technique you can use in your daily life. Although recursion is a simple technique once learned, it's a difficult concept to grasp and requires learning a new way of thinking.

Recursion encourages you to look for a simple solution to a complex problem. To apply recursive problem-solving, you first see if you can break a complex problem down to a simpler version of itself. You then look for a solution to this simple problem. Once you've discovered a solution, you use it to solve the more complex problem. This may all seem "simple," but you'll find this a difficult technique to apply in practice. Still, Logo's recursion feature can be used to write programs that solve problems created by a larger program. The applications possibilities are indeed very powerful.

BASIC Programming

BASIC is not extensible, modular, or recursive. Its strength lies in its use as an all-purpose tool that is especially tailored to solve simple or complex computational problems.

Unfortunately, programs written in BASIC are not easy to modify. The interactive nature of BASIC does allow a programmer to test pieces of a program, but this facility is not extremely useful because of the lack of structure built into the language.

Many experienced BASIC programmers will argue that this structure (called "structured programming") is possible in BASIC. This is true, but it takes a great deal of self-discipline to write structured code in BASIC, while in Logo you structure your program naturally along the lines of your own thinking. Thus, Logo teaches good programming habits and, therefore, some say, good thinking habits.

We would like to warn the reader at this point to be careful when comparing these languages by their commands alone. It is true that there are some *literal translations* of Logo and BASIC commands. But, for the most part it's a frustrating approach, because the structures of both languages are so different. You're faced with a similar problem when comparing different human languages—especially in the case of languages from Western and Eastern cultures. Certain social habits and customs that are peculiar to each culture simply cannot be translated literally.

The User Environment

Finally, before choosing a computer language, it is important to consider your purpose in using the computer—the environment in which you will be working. Following are several distinct *User Environments:*

- Computer Literacy—the user wishes to learn how to use computers.
- Problem Solving—the user who is developing logical skills and creative thinking.
- Computer Science—the user who wants to learn about the design of computers and programming languages.
- Computer Enthusiast—the person who loves to "compute" and who learns everything he or she can about

(continued on page 72)

The Fine Art of Publishing

High School Students Make Print

by Stephen Vincent

THERE ARE THREE WEEKS left in a busy high school semester. All year long you have prompted, provoked and pleaded with 40 creative writing students to tap their potential. And, everyone in the class has responded with one goal in mind—to see his or her work published, just like a professional author. As a result, you have a cabinet full of poems, stories, ditto masters, and multiple copies of work you've reprinted for discussion—all of which now looks like a cramped paper monster ripe for the school dumpster. What's a teacher to do?

This spring Judy Bebelar, the creative writing teacher at Galileo High School in San Francisco, and John Marron, the California Arts Council Poet-in-Residence, faced just this dilemma. Fortunately, teachers, students and poets were rescued from the chaos by the timely arrival of an Atari Computer system.

As part of a program developed by San Francisco's Momo's Press, with a grant from the Atari Institute for Educational Action Research, I helped organized a network of writers, teachers and computers in the schools. Working in a special computer room, students learned how to use the AtariWriter word processing system. In the meantime, staff and students had picked the year's best poems and stories to be stored on diskettes.

As they entered material in the system, students and staff alike—all computer novices—were amazed at how simple it was to correct spelling errors, alter syntax, and change the spacing between words and lines in a poem. A book that would have taken hours to compose and correct on a typewriter, and cost a lot of money to professionally typeset, was taking rapid shape.

Since the computer was centrally located, students could come in at various times during the day and continue the work begun by others. And the addition of another computer accelerated the input-process, as two students at a time could create a file and, at the end of the day, save their work on a single diskette.

After the initial input of the material came the multiphase work with the Atari 825-80 Column Printer. When each poem or story was entered, it was given a File Name, and was Chain Filed to indicate the piece that would follow (ANTH001, for example, would have [CTRL]V D:ANTH002 as its bottom line). Thus, the printer reproduced the entire work of 88 pieces in succession onto the fanfold paper. Right off, two copies were given to proofreaders for corrections. (One of the nice things about being able to print multiple copies is passing out copies to several people—we had access to a printer, after all, but not to a Xerox machine.)

After the corrections had been made on the computer screen and the revisions were saved on diskette, it was time to make a mock-up version of the book. We took a stack of 8½-by-11-inch paper—the page size of the book—and taped on the stories and poems from the print-out. What we did, essentially, was book design. We decided where we wanted the poems to appear on the page (right side, left side, or middle) so that we could set the margins for the printer. We also wanted to make sure that long poems and stories did not conclude with design "hangers," that is, two or three lines left hanging on a page. In making this mock-up, students learned what it is to give a book its character by seeing how the poems and stories fit together visually.

The completed mock-up of *Facets* (the title of our anthology) gave us a real sense of excitement. An actual book now looked possible! But it was here that the hardest work began. We had to learn how to format the text with the fonts available on the Atari 825 Printer.

We discovered, even with only four fonts (*Proportional Spacing, Condensed Print, 10 Characters Per Inch,* and *Elongated* Print) that there were more options than we thought. After some experimentation, we decided on making the titles a mix of Elongated Capitals and Condensed Print, centered over each piece. For the text, we decided Proportional Spacing provided a nice contrast to our title face.

The only problem we encountered was that our printer initially did not want to follow our commands. The titles refused to properly center. And the [RETURN] command, used to create the space between titles and text, interfered with the command to change fonts. We were quite confused until the school computer teacher and technician, Phil Lawson, consulted the printer's manual and figured out a simple set of commands. Instead of using the [RETURN] command for blank lines, we used the printer command [CTRL]0 and the number *10* for each blank line that we wanted. To solve the centering problem, which again seemed related to the problem of having so many diverse printing commands, we set the paragraph indentation [CTRL]P8, for eight spaces, and [RETURN] to establish a consistent, if not centered, margin for the titles over the poems. But once we had our formulas and techniques perfected, we were ready to go to the final step.

Camera-ready copy was produced by feeding single sheets of glossy paper into the printer. (Glossy coated stock makes better photo negatives for printing than ordinary fiber stock.) Page by page we went through the book. Each poem was brought up on the screen and given its set of printer commands; each piece of paper was fed into the printer. Because our work had been chain filed, it was possible to prinr out two successive poems on the same page. Our

preliminary work was beginning to pay off. Of course, it took awhile to get our rhythms down. But, if the pages didn't look quite the way we wanted them to, we could always paste in any tricky design corrections.

Finally, much to our excitement, the book was in hand, and ready to go to an "Instant Press" for a quick edition of 200 copies. In three weeks' time the Galileo creative writing class had edited and input manuscripts, proofread and revised copy, and produced camera-ready work for a 50-page book. Through direct involvement in the book-making process, each participant learned valuable skills which led to a real sense of personal accomplishment.

In making *Facets*, the Atari 800 Computer at Galileo High School established itself as a wonderful collaborative tool through which creative voices can gather and be organized into a lovely and powerful shape.

As the school opens this fall, creative writing teacher Judy Behelar, poet John Marron and the writers connected with Momo's Press are using the AtariWriter to create original poems and stories. Judy Bebelar is already looking at her classroom cabinets with a sigh of relief; all this new work will be kept on diskettes (protected, of course, from electro-magnetic interference). In her mind's eye, the huge and cramped stacks of student work and ditto masters have already been computed into oblivion. ■

Stephen Vincent is Editor and Publisher of Momo's Press, a fiction and poetry press in San Francisco.

THE POEMS AND STORIES *here are selected from* Facets, *the annual magazine published by students in creative writing classes at Galileo High School in San Francisco. The classes are led by teacher Judy Bebelar and John Marron, the California* Arts Council Poet-in-Residence. *The work here reflects Galileo's multi-cultural character. Publication of* Facets *was made possible by a computer equipment grant from the Atari Institute for Educational Action Research and a grant from the San Francisco Education Fund.*

CHINATOWN/TONG-YUN-FOW

The live catfish
still in their small tanks,
squished together
in a crowd, like
the streets
with people–
Chinese and American tourists,
the smell reaching from
Jackson Street to Broadway:
the odor of fish, turtles,
live frogs, oysters,
and any kind of sea creature
you can name.
The grocery stores,
Bak Choy and Guy Lan vegetables.
The only heaven there is;
the cooked food market
where the roast ducks hang,
the Chow Dow Foo Guak(fried soybean cakes)
and the Yim Suey Gi(salt water chicken)
are behind the glass windows.
The oriental food aromas
leak out the crowded doorway
and mix into the crowd...

Lisa Lee

FAREWELL, EMPTY SAIGON HOUSE

In the summer of 1978, a slight wind blew a coconut into a fast flowing stream. The current carried the floating coconut far from the fertile soil of its mother tree. The current took the nut to an unknown rocky and muddy delta where it was to be nurtured and where it would grow.

Leaving the place where I had lived for almost fifteen years was a sadly memorable experience for me. This departure would determine my fate forever. Leaving, I regained my freedom. Otherwise I would have faced a life sentence in a labor camp.

At this moment, I'm trying to remember. The years inside my father's red brick-color four story house; the rose garden; several fish ponds; the elegant marble stairway; the fireplace and the furniture; the pink living room; my healthy German Shepherd, Dowrly; my lovely gray Persian cat. My home was occupied and confiscated a long time ago. I imagine this building empty now and echoing with hollowness.

I'm also trying to be in the house where I spent my childhood. In the evening, when I walked down the long stairway, I could hear my mother sobbing in a candle-lit chapel. Looking down the hall, I could see the dim light coming from the slightly opened door of Dad's room where heavy cigar smoke floated around a light bulb. On the opposite side of the hall, a small suitcase was placed outside my sister's room.

Early in the morning, we left secretly. Dad, motionless, took a last glance at the house. His face was dry and pale. This building contained the history of my dad's successful business. He left his home before the sun came out. When dawn arrived, only a house remained.

An Hung Kha

lizard tail

lonely morning glory

spiders

eucalyptus blossom

delicate wing

free

Beth Wise

BEAUTIFUL DAY

It's raining
I overslept
Mom's a grouch
My hair is wrong
Nothing fits
Rotten day
Already

Late to school
My hair got wet
Teacher's a grouch
At lunch, he sat with her
Rotten day
Still

He walked me home
Held my hand
Said he liked my hair
Mom smiled at me
Then left me alone
He kissed me
Beautiful day
Now

Sharon Dea

CLOSING

CEDEN Project Brings Bilingual

IS AMERICA BECOMING A NATION OF computer "haves" and "have-nots"?

The "haves," naturally, are affluent, usually suburban youngsters with a computer at home and a programming class on their school schedule; many have written their own database management programs by the time they're 13 years old.

The "have-nots," on the other hand, typically are children living in poverty-stricken inner-city areas; merely learning to read and write English as well as Spanish is all the challenge they can handle. The schools they attend are usually just as poor as the children's parents, and a microcomputer is an unthinkable luxury that would supplant vital necessities. This makes for an especially tragic situation in light of recent studies which have shown that it's precisely such children who benefit most from the mental stimulation, hand-eye coordination, and decision-making abilities that microcomputers foster.

The irony of all this is not lost on Emily Vargas Adams, a former computer consultant to UNESCO (the United Nations Education, Scientific, and Cultural Organization) and now the director of a community-based research and service center that's bringing the microcomputer into the Mexican-American barrio of East Austin, Texas.

At CEDEN (the acronym is derived from the Spanish words for Center for the Development of Non-Formal Education), Adams and a group of dedicated colleagues are striving to improve the lives of low-income Mexican-Americans through three projects: A *Parent-Child Program* seeks to prevent or reverse developmental delays caused by improper nutrition or neglect in very young children. A *Family Assistance Program* provides food, clothing, rent, and counseling on additional financial help to indigent families. And the *Computer Education Program* gives children two to six years of age and their parents an opportuniry to use Atari Home Computers, and promises to narrow the gap between the computer "haves" and "have-nots."

One of the basic goals of the CEDEN computer program is to stimulate learning among "at-risk" children—those two-to-three-year-olds from impoverished and undernourished families. These children tend to fall behind in personal development—oral communication, interpersonal skills, and so forth—and, when they're older, in schoolwork as well. The CEDEN staff also works with the parents of such children, so that the parents can in turn help the children develop perceptual and social skills and improve their fine- and gross-motor abilities.

CEDEN's approach to "computer literacy" is a unique one: Rather than setting up computers in schools or a mobile unit, Adams and her principal colleagues, Patricia Platt and Mary Donley, decided to create what they call a "computer learning house," for which they chose a small house located in the center of East Austin's barrio.

"When we first opened the center in January, we wondered what the response would be," Adams recalls. Families that had taken part in previous CEDEN programs were invited to participate in the first six-month (January to June) computer session. "Almost everyone showed up—we were delighted with the enthusiastic response."

A NEW GENERATION OF COMPUTER ENTHUSIASTS:
Children in CEDEN's computer learning center develop problem-solving skills.

PHOTOS BY PATRICIA PLATT

THE GAP

by Jim Carr

Computer Education to Texas

"We asked the families to respond to a lot of questions about how they felt about computers," the CEDEN director says. "Virtually all the parents were enthusiastic because they see computers as a vehicle for improving their economic status, and they believe computers will aid their children in learning."

When they visited the center for computer lessons, children and their parents were guided by Mary Donley —who had a child of her own in the program. A flexible routine was developed to prepare the families for working with the computers. The group began by "playing" with such concepts as relationships and colors, then moved on to well-planned, on-computer activities. "We used a full range of preschool and elementary school programs such as Atari PILOT," explains Adams. "Our primary goal was to make the children and parents enthusiastic about each activity—art, music, Video Easel and other paint-type programs, and simple type-and-tune programs—all things that are fun. The children became totally engaged by the computer." Each lesson closed with a post-computer session designed to reinforce what the children had learned.

The Center personnel hoped to accomplish three things during these lessons. "First, we wanted to motivate the children," says Adams. "Second, we wanted to introduce them to fundamental problem-solving techniques. Finally, we worked on basic skills such as math, reading, and a lot of poetry writing and storytelling."

"It's especially important for young children to become engrossed," notes Adams, "because when they can put their own content into learning, it becomes theirs. The minute they don't have something invested in it, then you have the potential for losing them."

But there was little cause for worry on that account. "Families would stay for many hours," laughs Adams. "They didn't want the sessions to end." Post-computer activities, such as drawing pictures and storytelling, often served as gentle ways to terminate the sessions.

One of the major benefits that young children gain from using a computer, confides Adams, is that computers give them the opportunity to make "decisions that they otherwise wouldn't have had." Nonviolent games such as PAC-MAN and *Frogger* let the children make decisions in a relaxed environment. Learning which are the most effective and ineffective moves in such games, they build "a reservoir of understanding of what are the best decisions in any particular circumstance," adds Adams. "They can then apply that decision-making method to other problems."

This was one of the main findings of Adams' experience with UNECSO in Spain. "In the late 1960s and early 1970s, I had the opportunity to develop a computer-education program in Spain," she explains. "We found that children learn very quickly to utilize the computer, and they were ahle to turn around and teach their illiterate parents to read and write using the computer.

"That experience led me to want to develop a similar program here," she adds. Fortunately, the "computer revolution" began about that time, making low-cost microcomputers readily available.

The Atari Home Computer was selected for use in the CEDEN program for several reasons, notes Adams. "I did a study on various types of computers, and decided that the Atari Computer would be the best computer for our purpose, given its graphics and music capacity in addition to its memory size."

So, in 1981, she applied to the Atari Institute for Educational Action Research, asking for four complete Atari Home Computer systems (computer, disk drive, printer, program recorder, and various software).

This was just the sort of project the Institute wants to support, notes Institute founder and Director Ted Kahn. "It's probably the best single model program we have for introducing very young children to computers," says Kahn.

The Institute awarded funding to CEDEN for three major reasons, he explains. "First, this is definitely a case of an audience that would not under other circumstances get access to computers," Kahn explains.

"Second is Emily Vargas Adams. She's very well-known for her work in informal education with the Ford Foundation in Latin America and with UNESCO.

"Third—and probably, most importantly—this was the first opportunity in the U.S. to support a computer project in a multicultural setting. The children and their parents are bilingual (English and Spanish), but they don't necessarily read English, so it's an excellent testing ground for software in a bilingual setting."

One result of the program is that CEDEN staff members have adapted and translated Atari software programs into Spanish for international distribution, Kahn says.

In addition to the Atari Home Computers, the Atari Institute provided funding that allowed Patricia Platt to produce a videotape documentary of CEDEN's computer project. "The videotape allows us to show how a good informal educational program can be implemented—how it can relate to the culnire of the Spanish barrio," Kahn says.

The Institute this summer supplemented its original grant to CEDEN by contributing another four Atari Computer systems to the program. This will allow CEDEN to expand its computer project to include more families and more children this fall, says Adams, though more staff must be trained and additional funding acquired. "It's not entirely clear how many more we can handle," she explains. "It'll be a function of staff time and capacity—we don't have space for a thousand."

Adams hopes CEDEN's computer program will be duplicated by other organizations planning similar projects. She spent the summer months evaluating the program's first six months of operation.

Meanwhile, Adams promises that CEDEN will continue trying to make computers "a living part of the Mexican-American community."

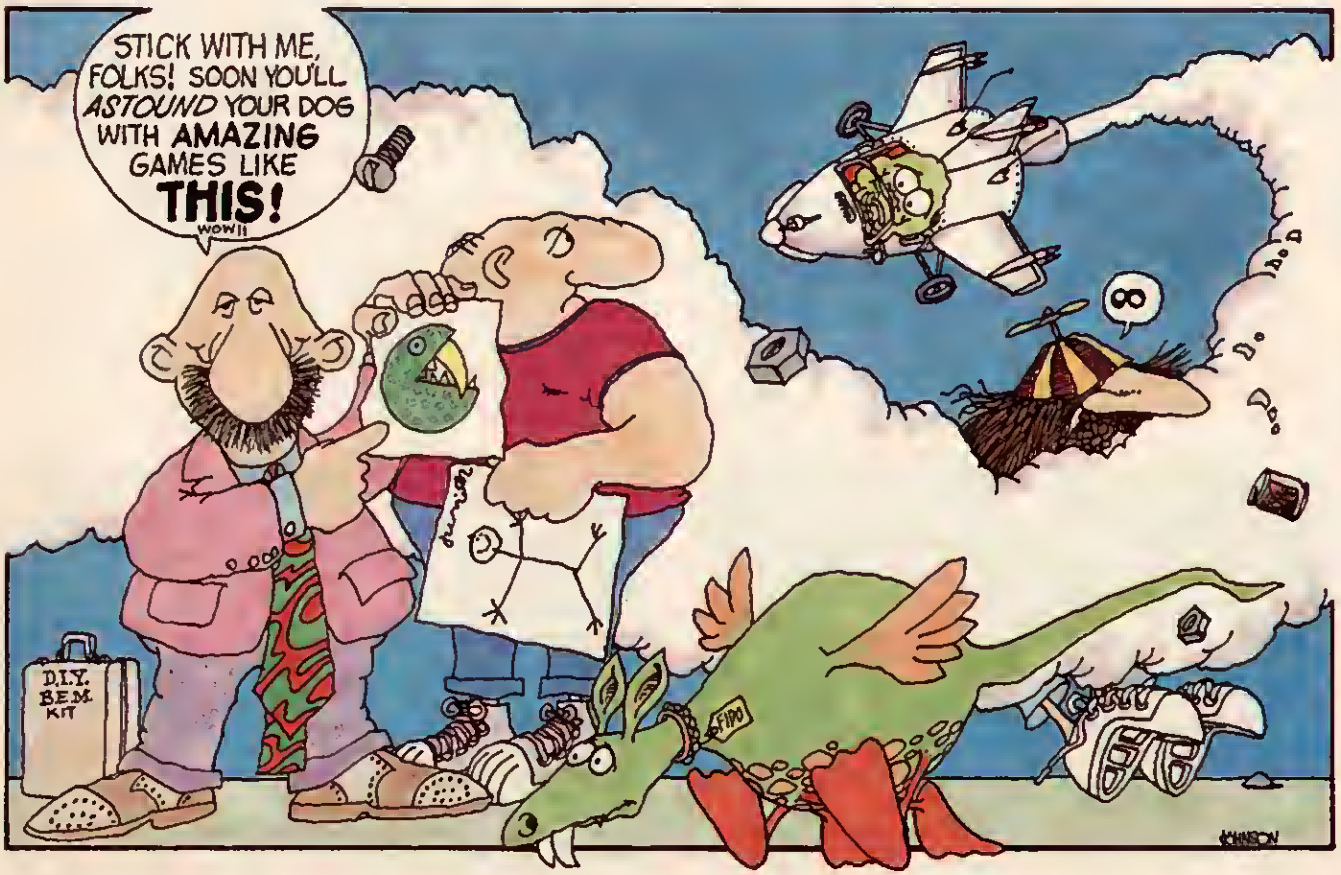
A lot of "have-nots" are counting on it. ■

Jim Carr is a Senior Writer in the Creative Services Department of the Atari Products Company.

Dr. C. Wacko's Cure for the Game Blahs

☞ *Have You Played Every Video Game?*
☞ *Have You Become a Chronic Critic?*
☞ *Think You Can Do Better?*

Then Design Your Own Computer Games with Dr. C. Wacko's Game Design Elixir!



ILLUSTRATED BY JOHN JOHNSON

HOLY ZANZIBAR! Are you going to be glad that you bought this issue of ATARI CONNECTION! In this issue, and the next two issues, I'll show you how to make chartreuse creatures squirm obnoxiously under the control of your joystick and rocket ships whiz across your brilliantly colored screens. In short, you'll be on your way towards designing and programming your own computer arcade games in ATARI BASIC! The internationally famous Dr. C. Wacko will tell all in three articles here in ATARI CONNECTION magazine. Trust me—you'll soon be astounding your friends and neighbors with your own action-packed arcade games!

It's exciting, and dangerous, to be on the stage again. This is where illusion becomes reality—where wackos like me get the chance to perform, be applauded, booed, hissed and get downloaded . . .

I know you're anxious to try out your super arcade game ideas so you can bask in the sunshine of success and fame, like me—so stay weird, buckle your seat belts, and here we go!

Causing a Scene

I always get a kick out of writing games that play on the Atari Computer. This great computer offers so many different stage settings (called Graphics Modes), that I can always find the right one for any arcade game I've dreamed up—the weirder, the better!

You too can use the Atari Computer's screen as the stage for all your arcade games. But, before you can astound the peanut gallery with your genius, you'll have to select the proper scenery, then do some set design and construction.

Constructing a stage set and backdrop for your arcade game takes a lot of backbreaking work, so to make things easier, you first need to get acquainted with the Atari Computer's world-famous Graphic Modes. The first pearl of wisdom I'm going to impart is this: to select a Graphics Mode, use the BASIC command, *GRAPHICS* (or its abbreviated form: GR). For example, this short direct command selects and displays Graphics Mode 3:

GRAPHICS 3 (RETURN)

Since a picture is worth a megabyte of words, check out your screen after you type in and RUN the following program. It shows you all the "stage settings" your Atari Computer offers. (Note: Your Atari Computer may have as many as *twelve* Graphics Modes. To keep things simple, let's only discuss and use Graphics Modes 0 through 8. Not that I'm lazy. It's just that most BASIC arcade games don't use Modes 8 through 11 because each Mode consumes 7,900 bytes of memory—which is a lot of valuable memory that can be used for making game players such as bouncing fuzzballs stick to walls.)

Dr. Wacko's World-Renowned, Selecting-the-Stage Program

After you RUN this program, just press START to view each of the Graphics Modes. (SAVE this program to diskette or cassette.) As you flip through each of the nine Graphics Modes presented in this neat program, you'll notice a few important differences between each screen.

```
10 FOR X=0 TO 8
20 IF X=0 THEN GRAPHICS X:
   POKE 752,1:POSITION 13,11:
   PRINT "GRAPHICS ";X:X=1
30 FOR A=1 TO 100:NEXT A
40 IF PEEK(53279)<>6 THEN GOTO 40
50 GRAPHICS X:POKE 752,1:PRINT:PRINT
   CHR$(127);CHR$(127);"GRAPHICS ";X:
   POSITION 5,5:PRINT #6;"GRAPHICS ";X
60 FOR A=1 TO 100:NEXT A
70 IF PEEK(53279)<>6 THEN GOTO 70
80 NEXT X:GOTO 10
```

Text Modes: The first three screens (Graphics Modes 0, 1, and 2) are known as "Text Modes" and are normally used (you guessed it!) to display text.

Pixel Modes: The remaining six screens (Graphics Modes 3 through 8) display colored squares in place of text. Each colored square is called a "pixel."

The Thin Blue Strip

A thin blue strip runs mysteriously across the bottom of Graphics Modes 1 through 8. (In Graphics Mode 8 the blue strip is hard to see because the entire screen is the same blue color as the strip—but it's there. Trust me . . .) Anyway, you can use this area to display information such as "Rate of Descent" in a Lunar Lander game, "Bearing to Target" in a battleship gunnery game, or "Earth Calling Dr. Wacko" in a *Come Back to Reality* game.

You enter text into this window with the PRINT command. Here's an example:

```
10 GRAPHICS 3:POKE 752,1
15 . 'POKE 752,1' gets rid
   of the cursor
20 PRINT "Earth calling Dr. Wacko!"
```

If your game doesn't use the text window, you can destroy it—ZAP and create a *Full Screen* display by adding *16* to the graphics designation number. Here's how to get rid of the "thin blue strip": To change Graphics Mode 3 from a Split Screen to a Full Screen display, use the command, GRAPHICS 19(3+16=19). Try this two-line program:

```
0  GR. 19
20 GOTO 20
```

You've eliminated that pesky blue strip, and the screen's grown! Do you remember the *Selecting the Stage* program you just SAVEd? Well, bring it back! Cleverly modify the program by deleting line 50 then replace it with:

```
50 GRAPHICS X+16
55 POKE 712,X+10*INT(RND(0)*15)
```

Now, press the [START] key and flip through each Graphics Mode without having to watch that persistent "thin blue strip" run around across the bottom of your screen!

The Wacko Unified Hole Theory: Columns, Rows and Coordinates

Take a magnifying glass (or squint a lot) and look at the color cover of this magazine. You'll see thousands of tiny colored dots, right? Graphics are displayed on your computer's screen in exactly the same way—each Graphics Mode contains hundreds of "holes" (pixels) waiting to be filled in with color. In this article we'll be working with *Low Resolution* Graphics Mode 3. Graphic Mode 3's screen size is 40 columns across and 20 rows high—that's a total of 800 (40×20) empty holes waiting to be filled with color. In contrast, *High Resolution* Graphics Mode 7's screen size is 160 columns across by 80 rows high. That's a whopping 12,800 empty holes!

"How do I fill all of those holes with color?"

I'm glad you asked. It's simple! Each hole (or pixel) is assigned its own two-number location called a *coordinate*. Here's how this nifty system works:

The numbers across the top of the illustration assign a value to each column. These are called "X" locations. The numbers down the side assign a value to each row and are called "Y" locations. *Each pixel is identified by its two-number location coordinate.* The pixel's column location is *always* stated first, followed by a comma, then the pixel's row location, like this: "X,Y."

I've taken the liberty to fill in the pixels at locations 15,10 (X,Y) and 24,15 (X,Y) to show you this simple concept.

Using the COLOR & PLOT Statement

Graphics Mode 3 or 19 (3+16) is a four (4) COLOR Graphics Mode. This simply means that you have one (1) background color and three (3) pixel colors to work with. COLOR and PLOT go together like R2-D2 and C-3PO—they're inseparable. In Graphics Mode 3, you use COLOR 0 to "paint" the background of your playfield and COLORs 1, 2 and 3 to "draw" your playfield in vibrant color!

The PLOT command fills a pixel on your screen with the COLOR of your choice, at the coordinates of your choice. COLOR and PLOT are used together like this:

COLOR 1:PLOT 5,10

Here's a short program that PLOTs an orange pixel (COLOR 1) on your screen in Graphics Mode 3:

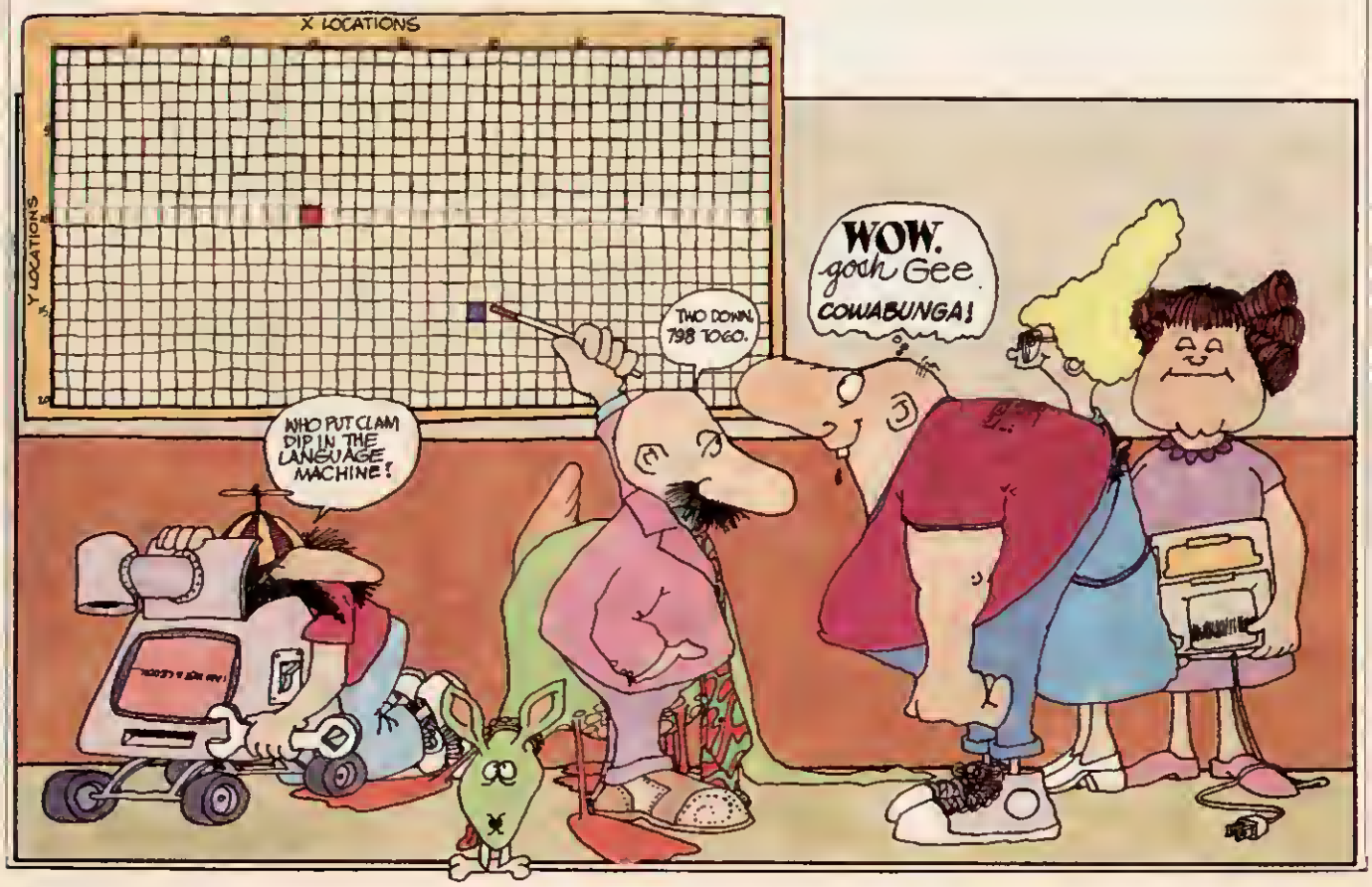
COLOR & PLOT

```
10 GRAPHICS 3
20 COLOR 1
30 PLOT 0,0
```

"But, what about COLORs 2 and 3?" you ask. Replace line 20 first with COLOR 2 then with COLOR 3. COLOR 2 appears as light green and COLOR 3 as a small blue box—amazing! You guessed it. The four COLORs are preset:

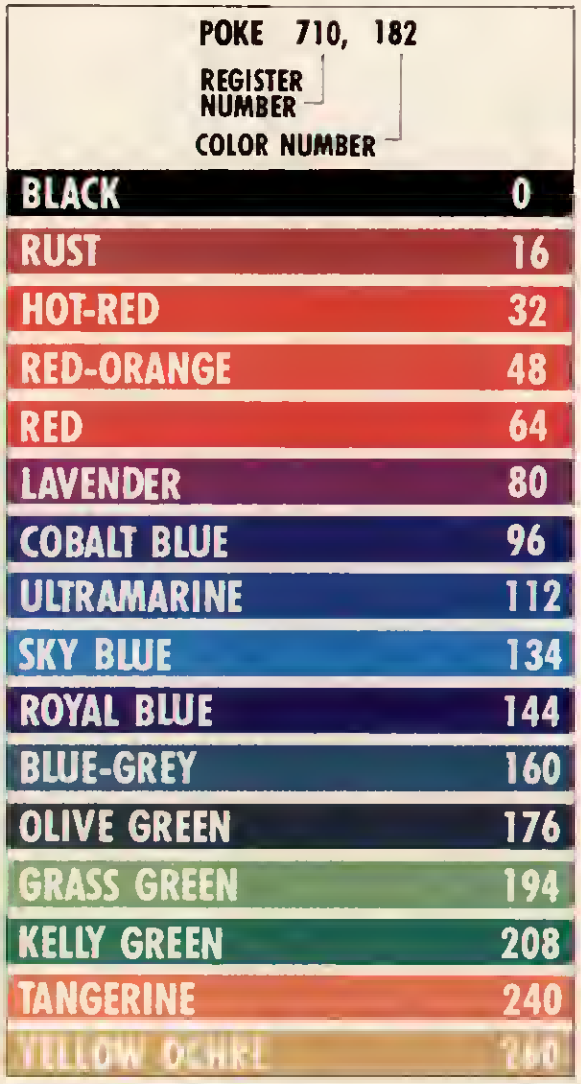
COLOR 0—Black (background)
COLOR 1—Orange
COLOR 2—Light Green
COLOR 3—Blue

"But, what if I want to change those colors?" you



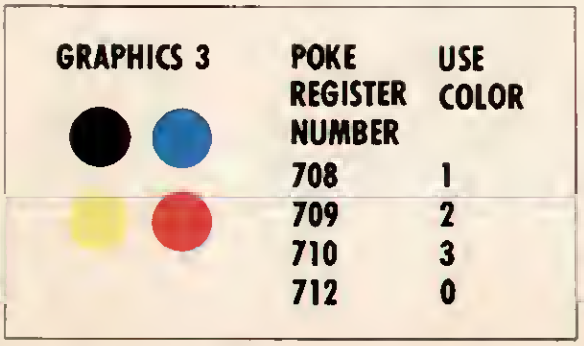
say. Have no fear, Dr. Wacko's here with POKE to the rescue!

A POKE statement has two numbers. Study the illustration below:



You use the POKE statement to stuff information directly into a special location inside your Atari Computer's memory. This location is called a *register*. The first number in the POKE statement tells the computer which register you want to stuff information into. The second number is the stuff that you're "POKEing."

Listed below are the four POKEs used in Graphics Mode 3 to control each of the four COLORs available:



By adding a number between 0 and 255, you can change the color of each PLOTted COLOR pixel. The added POKE color number changes both the pixel's color and its *luminance*. To get different shades of color, you simply add or subtract 2 from the color number. To see what all this POKEing around does, change line 10 of the COLOR & PLOT program to read:

```
10 GR. 3:POKE 708,99
```

Now, when you RUN your new program you'll see that little dot has changed to *pugnacious purple*!

Go wacko with this short program. By POKEing 708 with any number between 0 and 255, you can change the color of the pixel that you've PLOTted using COLOR 1. Experiment with other COLOR statements and registers. Replace the COLOR statement in line 20. Make it "COLOR 2," and change line 10 to read:

```
10 GR. 3:POKE 709,195
```

Play with these concepts and refer to my colorful chart until you've got a good grasp of how the different-colored POKEs control each PLOTted COLOR.

DRAWTO the Stage

Now that you're all wacked-out on color, I'll show you how to use the DRAWTO command to draw lines on the screen. But, before you use DRAWTO, you'll have to PLOT a beginning coordinate. DRAWTO is *always* used with PLOT like this:

PLOT X,Y:DRAWTO X,Y

Put the *COLOR & PLOT* program back together again and add this line:

```
40 DRAWTO 39,19
```

Again, fool around a little and then we'll mess it up some more. OK?

Now, using the same program, add this line 15:

```
15 POKE 708,99
```

And change line 40 to read:

```
40 DRAWTO 0,19:DRAWTO 39,19:DRAWTO
   39,0:DRAWTO 1,0
```

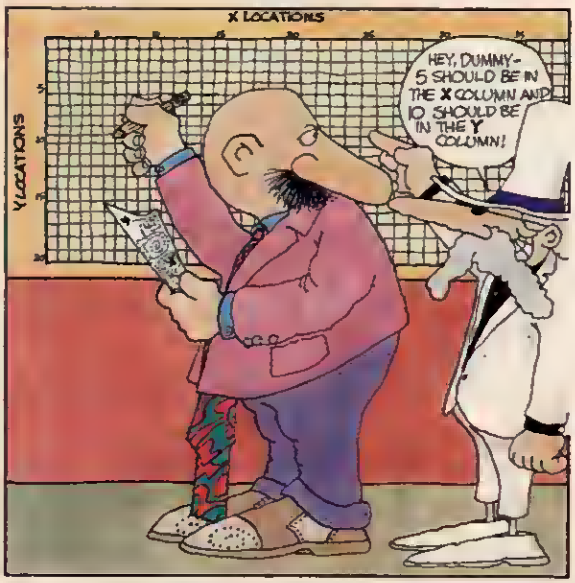
Now, when you RUN your modified program, a strip of purple borders the edge of the screen—it's that pugnacious purple again because we've POKEd COLOR 1's register with 99. Remember?

Line 40 shows that you can "chain" DRAWTO commands one after another. You can draw any arcade playing field imaginable by combining these PLOT, DRAWTO and COLOR techniques!

Stepping Through the "Amazing Maze"

Now, armed with your knowledge of Graphics Mode 3 commands, you're ready to encounter the *Amazing Maze* program. But, first, let's take a closer look at the program so you'll know what you're doing.

In line 10, after selecting "windowless" Graphics Mode 3+16, I've POKEd color registers 1, 2 *and* 0 (POKE 708, 709 and 712) with exciting and vibrant colors.



To speed up the drawing process, I often use the handy FOR/NEXT loop. Lines 20 to 90 of the *Maze* program quickly draw five COLOR 1 lines of decreasing lengths from the top to the bottom of the screen.

Here are lines 10 through 90 of the *Amazing Maze*. I've slowed down the drawing process by adding a pause FOR/NEXT loop in line 85 so you can watch the screen being drawn. RUN this short program and you'll get the picture—or a headache!

```
10 GRAPHICS 3+16:POKE 708,74
15 .
20 COLOR 1
30 FOR A=0 TO 4
40 PLOT A*2,A*2
50 DRAWTO 39-A*2,A*2
60 DRAWTO 39-A*2,23-A*2
70 DRAWTO A*2,23-A*2
80 DRAWTO A*2,A*2
85 FOR PAUSE=0 TO 200:NEXT PAUSE
90 NEXT A
95 .
100 GOTO 100
```

In the rest of the *Amazing Maze* program, lines 100 through 250, is broken into sections headed by a COLOR statement and followed by a bunch of PLOT and DRAWTO statements. COLOR 0, the background color, is used to "cut away" sections of the lines that were first drawn. COLOR 2 and 3 are used to draw the design in the center of the screen.

Now, let's take a closer look at some of the wacko things that are happening in line 1000.

```
1000 POKE 710,PEEK(20):GOTO 1000
```

The PEEK(20) command generates a constant stream of ever-changing numbers between 0 and 255. By continually changing the number that follows "POKE 710," COLOR 3 at the center of the screen continually changes colors! Here's a short demo program that shows how PEEK(20) works:

```
10 PRINT PEEK(20):GOTO 10
```

Here's another example:

```
10 GRAPHICS 3
20 POKE 712,PEEK(20):GOTO 20
```

Now that you know what you are doing (I wish I could say the same . . .), use your new skills to modify the *Amazing Maze* program's playing field. You can change all the colors by POKEing the appropriate COLOR registers; modify the PLOT and DRAWTO statements to make your own design—go completely wacko! Really mess it up!

Wacko's Amazing Maze

```
5 . WACKO'S AMAZING MAZE
10 GRAPHICS 3+16:POKE 708,74:
   POKE 709,60:POKE 712,128
15 .
20 COLOR 1
30 FOR A=0 TO 4
40 PLOT A*2,A*2
50 DRAWTO 39-A*2,A*2
60 DRAWTO 39-A*2,23-A*2
70 DRAWTO A*2,23-A*2
80 DRAWTO A*2,A*2
90 NEXT A
95 .
100 COLOR 0
110 PLOT 19,2:DRAWTO 19,21
120 PLOT 20,2:DRAWTO 20,21
130 PLOT 2,11:DRAWTO 37,11
140 PLOT 2,12:DRAWTO 37,12
145 .
150 COLOR 2
160 PLOT 10,10
170 DRAWTO 29,10
180 DRAWTO 29,13
190 DRAWTO 10,13
200 DRAWTO 10,10
205 .
210 COLOR 3
220 PLOT 11,11
230 DRAWTO 28,11
240 PLOT 11,12
250 DRAWTO 28,12
255 .
1000 POKE 710,PEEK(20):GOTO 1000
```

LOCATE and Collisions

Chances are, you're going to want lots of *action* in your games. When a missile strikes its target, you'll want to see a flash and hear an ear-splitting explosion—*BAAROOOOOM*!! When a ball smacks the side of a wall, you want to know it—*PING*!! One way to let the weirdos and other elements in your program know that a collision has occurred is by using the LOCATE statement. LOCATE is used in this format:

LOCATE X,Y,Z

"X" is the column location of the collision, "Y" is the row location, and "Z" is the value that's encountered at location "X,Y." (You don't have to use "Z"—any variable other than X or Y will work just fine.) In Graphics Mode 3, the value returned in "Z" is the number that follows the COLOR statement, either 0, 1, 2 or 3. Here's a short program that shows this basic LOCATE concept. There's a big bonus program at the end of this article that uses LOCATE in an action arcade situation!

```
10 GRAPHICS 3
20 COLOR 1
30 PLOT 0,0
40 COLOR 2
50 PLOT 1,1
60 COLOR 3
70 PLOT 2,2
75 .
80 LOCATE 0,0,A
90 LOCATE 1,1,B
100 LOCATE 2,2,C
110 LOCATE 3,3,D
115 .
120 PRINT A,B,C,D
```

After you RUN this program you'll see three colored squares displayed at the upper left of your screen. Their corresponding COLOR statement numbers (1,2,3, and 0) are LOCATED then printed in the text window.

COLOR 1 (Orange) is plotted at 0,0
COLOR 2 (Green) is plotted at 1,1
COLOR 3 (Blue) is plotted at 2,2
COLOR 0 (Black) is the background color

Amazing! I shall return next issue with my joysticks to show you how to move, bop, dive, blast and sneak your way into your own action-packed arcade game!

Wacko's BONG Program

And now, here's the grand finale bonus-program I promised earlier. I've titled it *Bong*. It's a version of my famous "sing-along with the bouncing ball" program that wowed 'em in Peoria. This super program uses many of the elements that you have learned in this article. It's also a great example of how to use the LOCATE command in an arcade game—so, without any further padew, here's BONG:

Bong

```
10 GRAPHICS 19:POKE 710,15
20 X=1:XB=X:Y=1:YB=Y:DX=1:DY=1
30 COLOR 1:PLOT 14,11:DRAWTO 26,11
40 COLOR 1:PLOT 0,0:DRAWTO 38,0
50 PLOT 0,23:DRAWTO 39,23
60 COLOR 2:PLOT 0,1:DRAWTO 0,23
70 PLOT 39,0:DRAWTO 39,22
80 COLOR 0
90 PLOT XB,YB
100 XB=X:YB=Y
110 X=X+DX
120 Y=Y+DY
130 LOCATE X,Y,Z
140 IF Z<>0 THEN GOTO 190
150 COLOR 3
160 PLOT X,Y
170 POKE 77,0
180 GOTO 80
190 SOUND 0,100,10,10
200 IF Z=1 THEN DY=-DY
210 IF Z=2 THEN DX=-DX
220 X=XB:Y=YB
230 POKE 707+Z,PEEK(20)
240 SOUND 0,0,0,0
250 GOTO 110
```

Here's how *Bong* works:

Line 10: This line first selects Graphics Mode 3+16, then POKEs COLOR 3 with the number 15. This command (POKE 710,15) paints the bouncing ball white.

Line 20: These commands assign the ball's movement parameters. (More on movement in upcoming articles!)

Line 30: This line draws the horizontal bar at the center of the screen. I've selected COLOR 1 (organic orange) as the color of my "drawing ink."

Lines 40 and 50 draw lines across the top and bottom of the screen using COLOR 1.

Lines 60 and 70 draw lines down the left and right sides of the screen using COLOR 2.

Lines 130 and 140 make the *Bong* "action." The "Z" in the LOCATE statement in line 130 senses which "wall" COLOR the ball has hit. In line 140, if the value of "Z" is *not* 0, the ball has collided with one of the walls (COLOR 1 or 2) and the program jumps to line 190—BONG!

Lines 200–210 make the ball bounce away from the wall it hits: Line 200 for a horizontal wall (COLOR 1); line 210 for a vertical wall (COLOR 2).

Line 230 changes the color of the wall when the ball hits it. The trick to this feat is pulled off by PEEKing into memory location 20 [PEEK(20)] and POKEing its value into the proper COLOR register. PEEK(20) is a source of ever-changing numbers between 0 and 255, which means lots of colors! So, every time the ball hits the wall, a different color is POKEd into the "wall color." ■

Dave Heller is the Technical Editor of ATARI CONNECTION *magazine and the author of two other books for the Atari Home Computer:* Space Knights, *Reston Publishing, and* Free Software for Your Atari Computer, *Enrich/Ohaus.*

Ten Tips from the Programming Pro

by Jim Inscore

FANTASY IS THE KEY that unlocks the door to game design artistry—at least that's how Chris Crawford sees it. And, as Atari's game artist in residence, Chris definitely has his finger on the fire button of effective game design.

More gnomelike than nerdish, Chris can often be found deep within the bowels of Atari's R&D facility, toiling away over a line of code or a scrolling map, or discussing a tough design trade-off with a colleague, to distill the penultimate essence of perfect game art.

It's no small task, but Chris has consistently delivered the goods. His informative nuclear plant simulation game, *Scram*, was followed by World War II-based *Eastern Front (1941)* from Atari Program Exchange (APX), and by *Centurion*, a game of strategy that is played out among ancient Roman legions.

This summer, Chris is back with a quadruple whammy: *Eastern Front (1941)*, a new version of his earlier hit, the new games *Excalibur* and *Gossip*, and a book of his insights into effective game programming, *The Art of Computer Game Design*. All are certain to appeal to a wide audience of game buffs. "Technical issues are not the dominant concerns of the book," says Crawford. "Artistic considerations are. So anyone interested in and concerned with the issues surrounding computer games should be interested in it."

While appealing to such a broad audience, the book also addresses the issues of how to program a good game, with some specific pointers on how to get the ball rolling. With those in mind, Chris ticked off for us his ten tips on successful computer game programming.

PHOTOGRAPHED BY STANLEY GREENE

Chris Crawford's secrets for effective (and artistic) game design . . .

1. WORK IN ASSEMBLY LANGUAGE.

Nothing else provides the speed and utility that assembly provides. BASIC is an adequate first step to learning to program—thousands of people are writing games in BASIC. But BASIC is not a production language. In order to write really effective programs, you have to work in assembly language.

2. BE ORIGINAL.

Everybody in the game business seems to spend a great deal of time copying one another. This is where amateurs have it all over the professionals. They're not wedded to anything; they're not being asked by their boss to crank out anything in particular. If you're going to copy something, recognize it as a copy and use the experience to learn—then go from there and design something that's your own.

3. HAVE A GOAL.

Know what you're trying to accomplish. Are you working for yourself? For someone else? For your friends? For money? Beyond that motivating goal, decide what the goal of the game is. Don't just design an effect. Develop a game play strategy, a goal for the game player.

4. STORE LESS AND PROCESS MORE.

Data is static. Process is dynamic. The more data statements in a program, the more beautiful the graphics may be, but the less involving the game will be. Lots of conditional statements and branches in a program are indicative of good, active game design. There's an analogy in writing—good writing doesn't use dead, dry verbs but active, dynamic verbs. If you're storing lots of data, using lots of tables and numbers, then your game will be static.

5. DESIGN AROUND INPUT/OUTPUT LIMITATIONS.

Don't ignore graphics—just be aware of their limitations. It's important that graphic and sound interaction is the best representation you can possibly create. But graphics and sound should flow from the logic behind the game.

6. CONCENTRATE ON GAME MECHANICS.

What goes on? What do the players actually do? Run and hide? Duck for cover? Persuade an opponent? The capabilities of the player are essential to define.

7. DESIGN A PROCESS.

Don't be graphics-driven. Don't just make a pretty picture and try to put a game around it. You'll end up with a game that may be technically flawless but incredibly dull to play.

8. POLISH, POLISH, POLISH.

When you get to the point where the game is working and is fun to play, you're about three months from being finished. From there on in, it's the tiny touches you add that make the difference. Play the game hundreds, thousands of times. Polish it to the limits of your patience—because ultimately, it's the limits of your patience that separate the adequate design from the truly great game. For example, *Eastern Front (1941)* went through six weeks of polishing before it was introduced as an APX game. With another four months of polishing, the new *Eastern Front (1941)*, which is now available on cartridge as a main-line Atari product, is immensely superior—more levels of play, the opportunity for longer game play, more of the parameters of the actual battle brought into play.

9. WRITE THE MANUAL YOURSELF.

There are two good reasons for this. Number one is that it may help you avoid bad game play. If you have a feature in your game that you think is just the greatest thing, but you find out it takes five pages to describe it, you should probably scrap that feature. Second, by putting it all down on paper, you provide a good starting place for a professional writer to come in and do the job more accurately. When game designers just hand their games over to writers and say, "Here, document it," the results can be mutually unsatisfactory. By giving a writer a written version of your concept of how the game needs to be described, you're ensuring a greater chance for the writer to succeed at doing his job.

10. DON'T EXPECT FAME AND FORTUNE.

Success in the game business is like lightning striking. Just as the record business has a few superstars and a huge mass of people trying to make it, so it is in the game programming biz. *Eastern Front* is a hit, but I didn't design it to be. I designed it for myself, to see what I could do. If you design games because you enjoy the process and the end results—the actual game—then you're ultimately going to end up being happy about what you're doing.

■

Jim Inscore is the Writing Manager for the Atari Products Company.

HARDWARE REVIEW

New Products for Atari Home Computers

REMOTE CONTROL JOYSTICK

Look Ma, No Cords

EVER BEEN INVOLVED in a heavy session of *PAC-MAN* or *Star Raiders* and gotten all tangled up in the joystick cord? Or perhaps knocked over a cup of coffee or strangled the cat with that same cord? Well, relief is on the way with the Atari Remote Control Joystick.

Simply plug the small receiver into the controller ports on your Atari Home Computer and you're ready to use the Remote Control Joystick anywhere in the same room. Using a low-power transmitter operated on batteries, the joystick sends your commands to the receiver through its whip antenna.

The Atari Remote Control Joystick is available in a complete package which includes two joystick/transmitters and a receiver. Suggested retail price is around $60.

Available now at your Atari Computer retailer.

ATARI LIGHT PEN

Artistic Control at Your Fingertips

ATARI WILL SOON introduce a light pen that lets you create graphics simply by drawing on a TV screen. Just plug the pen into the controller port and insert the graphics program cartridge. A black screen appears with a menu at the bottom.

You select the color you want by touching the light pen to that colored box on the menu. Then, when you touch the light pen to the screen, the computer draws any pattern you trace. Other boxes let you cycle the colors available, and create lines and rectangles automatically by touching the screen at the ends or corner points of the figure you want drawn. You can also draw in mirror or quad modes.

The Atari Light Pen will be available this winter, with a suggested retail price of under $100.—**Gary Paul Fox**

EXPERIMENTAL DESIGNS

Science at Home with Your Atari Home Computer

REMEMBER YOUR OLD science classes? It took awhile to assemble all the beakers, bottles, thermometers and mysterious potions you needed to do one simple experiment. You spent hours mixing and measuring, charting and plotting. Then, finally, when you had finished meticulously writing out your science report, you spilled a giant beaker of goop all over it, smearing the letters and graphs till they were illegible. For someone who had no desire to be the next Einstein anyway, it was quite a frustrating experience.

Now, like so many other things, even learning science can be made easier with the help of your home computer. The newest educational series by Atari allows you to use your computer to collect, analyze, and display scientific data at home or in the classroom. Virtually the only science application kit available on the market at such a low cost, the *AtariLab Science Series* can add a new dimension to a world of scientific discovery. And it makes it easy for people of all ages to learn science by doing science.

ILLUSTRATED BY LANCE JACKSON

The "AtariLab Starter Set" includes the AtariLab Interface, a special device which plugs into a joystick controller port and allows interaction between the physical environment and the computer, the Temperature Module Cartridge, a Temperature Sensor, and a Thermometer. The special interface connects to your computer. Also included in the starter set is a complete instruction manual with suggested activities and experiments for novices and experienced science buffs.

All you need for most of the experiments are typical household items. For example, say you want to try the experiment that graphs the heat reaction that occurs when an acid and a base substance are mixed together. The experiment is explained in the manual with detailed step-by-step instructions. First, plug in your AtariLab Interface, your Temperature Module Cartridge and Temperature Sensor. Next, measure equal amounts of vinegar (an acid) and baking soda (a base) into separate containers, and place the temperature sensor first in the vinegar, then in the baking soda. Your computer graphically displays the temperature of each on your TV screen. When both substances are mixed, you can see the change in temperature in either Centigrade or Fahrenheit, and the program will chart the rise and fall of the temperature for a period of 10 seconds to 24 hours. All data collected can be saved on either cassette or diskette and printed on an Atari 1020 Color Printer/Plotter.

The AtariLab Science Series requires no previous knowledge of programming, but, for users with programming skills, the manual includes examples of how to set up and program your own experiments in BASIC.

The AtariLab Starter Set with the Temperature Module will be available shortly, and other modules are under development for release in the near future. For example:

Light Module—Measures light intensity, reflectivity, and absorption using a light sensor, a small light, a special light stand and other accessories.

Biofeedback Module—Helps you learn to relax while your computer measures your skin moisture, temperature and pulse rate. Choose from a large array of sounds and color video displays to personalize your relaxation program. Literature on stress management is included with the module.

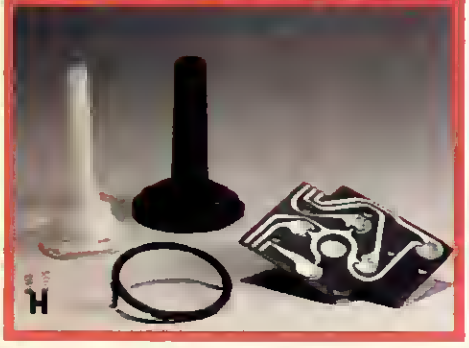
Timekeeper Module—Turns your computer into a stopwatch, a lap timer, or a repeating timer. A special metronome feature allows a musician to play under the direction of a "conductor" by watching a wand beat time on the screen.

Lie Detector Module—Gives F. Lee Bailey a run for the money! Discover who is telling the truth by using the sensors included in the module. Literature on lie detection also accompanies the set.

Mechanics Module—Measures projectile and pendulum movement. Do your surveying projects with the special angle sensor.—Teddi Converse

Requires 16K RAM. Cassette recorder or disk drive optional. Starter Set suggested retail price: $89.95. Light Module available in November, Timekeeper Module available in December, and Biofeedback Module available in January. Mechanics Module available in February.

Teddi Converse is a Writer in the Creative Services Department of the Atari Products Company.



THE
ATARI
GIFT
CATALOG*

ATARI®

GIFTS FOR THE ENTIRE FAMILY

ATARI gifts and computer accessories are as unique as people themselves and this selection offers something for everyone.

A

ATARI Back-Jack Chair. Finally, a leisure seat for playing games. As suited to the beach as it is to the bedroom, it's easily adjustable to several positions. The sturdy fabric and finely crafted frame make this a gift for anyone of any age. (Red, brown, blue, green)
#AC106 $19.95

B

ATARI DISKBANK. End your diskette-storage dilemma with this compact storage *and* filing system that grows as you grow with its modular design. Makes every diskette instantly accessible via sturdy interlocking cases that each hold ten 5¼" diskettes in slide-out drawers. Can be stacked vertically or horizontally depending on your space requirements. Comes with pressure-sensitive identification labels. Space-age storage for a space-age hobby. #AC105 $5.95

C

ATARI 810 Diskettes. Prevent cries of "Oh, no, no more diskettes" with this convenient package of five blank diskettes for use with the ATARI Disk Drive for recording programs and data. Not only a necessity, but a thoughtful gift as well. #AC104 $25.00

ATARI Sport Shirts For Sure. Whether your sport is golf, polo, rugby, rainbow-watching or computing, ATARI has just the right shirt for you. A great gift for any occasion.

D **CHILD'S POLO SHIRT** (dark blue, red, green, sky blue) S-M-L. # AC118 $9.95

E **GOLF SHIRT** (maroon, dark blue, sky blue, white) S-M-L-XL. # AC117 $14.95

F **RAINBOW SHIRT** (blue, cream) S-M-L-XL. #AC107 $14.95

G **RUGBY SHIRT** (dark blue, maroon) S-M-L-XL. #AC119 $14.95

H

Joystick Repair Kit. If those hot little hands have made your joystick jiggle, this kit lets you fix it yourself in just minutes. Comes with plastic insert, black outer cover for handle, printed circuit board, trigger springs and easy-to-follow instructions. No need to delay fun. #AC101 $4.50

I

ATARI Belt Buckle. Let everyone know you've joined the Computer Age. Cast in solid brass

to last a lifetime. Goes great with any outfit. Fits belts 1" to 1½" (belt not included). #AC109 $7.95

J

ATARI 3-Ring Binder. This sleek binder lets you know where to go when you need the info. Great for storing computer notes, manuals. Super gift for kids, Mom's kitchen, Dad's workbench. #AC102 $4.95

K

INSIDE ATARI BASIC Illustrated Book. No need for anyone in the family to break into a cold sweat when starting to compute. This easy-to-follow, witty and fun book guides you through key computer concepts at just the right pace and exposes you to programming, graphics, color and sound in a painless and playful way. A must. #AC103 $12.95

L

ATARI Button Clock. Keep real time while you're home computing. Convenient LCD clock sticks anywhere with self-adhesive back. Color matches ATARI Home Computer's keyboard.
#AC110 $6.50

M

ATARI "Captain's Cap." You'll be in charge with this official ATARI cap that's as suited to sailing as it is to softball. Adjustable headband, one size fits all. (Blue, green, red)
#AC108 $3.95

TO ORDER
use the bound-in order card

OR
SEND ORDER WITH YOUR CHECK TO:

Atari Gift Catalog
603 Baltic Ave
Sunnyvale, CA 94086

Include $2.00 for handling
add 5% of total order for shipping
California residents add
6½% sales tax

ATARI Game T-Shirts Plus Free Posters. Now you can get a free poster when you order your favorite full-color silk-screened Centipede or Cavern of Mars T-shirt. Cool 100% preshrunk cotton. Please specify game and size (children's S-M-L, women's S-M-L, men's S-M-L-XL). Children's: $6.95. Adult's: $7.95

N

CAVERNS	
Child	#AC111
Women	#AC112
Men	#AC113

O

CENTIPEDE	
Child	#AC114
Women	#AC115
Men	#AC116

SOFTWARE REVIEW

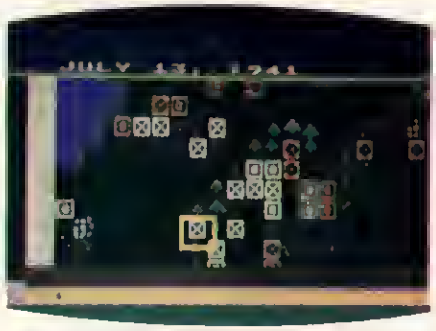
New & notable programs for your Home Computer

EASTERN FRONT (1941) REVISITED

Now with More Levels, More Strategy, More History

IF CHRIS CRAWFORD had been born forty years earlier, either General Patton or General Rommel —depending on where Chris was born —would have had a run for his money when it came to battle strategy. As it happens, however, Chris was born in the age of computers, so he created an Atari Program Exchange (APX) game, *Eastern Front (1941)*.

Chris's innovative game featured a detailed scrolling map of the Eastern Front, that vast expanse of Russia



which confronted the German Army as it tried to conquer the Soviets in the middle years of World War II. As you may recall, you played the Germans, and your Atari Home Computer played the Russians. Both sides were bound by the actual rules of combat, and had to plan their tactics accordingly. In addition, there was the weather to consider —each battle brought closer the onset of winter, when movement would slow to a crawl. A brain-teasing strategic exercise rather than a simple shoot-em-up, *Eastern Front (1941)* became a big hit for Chris and APX in the early 1980s.

The game was so popular that Atari hired Chris to design games full-time, and asked him to convert the APX game to a mainline product. "Convert!" Chris exploded. "I've had three years to plan new and better strategies. Convert, indeed!" He produced, instead, a new, improved *Eastern Front (1941)*.

Whereas the APX game consisted of only one campaign, the mainline Atari game consists of six, from a Learner's Level to *two* Expert Levels, and more closely simulates the logistical realities

of war. You learn to establish Zones of Control. You learn to worry about your supply lines, and to use your air support to best advantage. You can fight on through the winter into the spring, and mount a fresh campaign in a second year. You learn what it's like to face an unyielding foe on his home ground.

One of the highlights of the APX game was its detailed manual, explaining not only the game, but also the historical forces that shaped the actual invasion of the Eastern Front. The user's guide for the new *Eastern Front (1941)*, of course, covers much more of the campaign, and in even more detail. But in addition, it presents both game play and history in a highly entertaining documentary fashion. Letters from the front, fragments of speeches, and combat reports are interspersed with authentic photographs from the Eastern Front, so that the player truly feels himself the commander on the spot. A detailed map of the entire field of game play provides an overview of the campaign.

Thousands of people found the APX *Eastern Front (1941)* to be the best video game they'd ever encountered. The new *Eastern Front (1941)* from Atari is even better. What more can we say?—**Stephen Englehart**

Eastern Front (1941) *is available in cartridge format. Requires 16K RAM; joystick. Suggested retail price:* $44.95.

Stephen Englehart is a Senior Game Designer for the Atari Coin-Operated Games Division.

FUN AND GAMES—SPELLING IT OUT

Play Wordgo and Bingo! You're Learning to Spell While Building Your Vocabulary

WHEN I FIRST SAT DOWN to give *Wordgo* a try, I thought, "Uh-oh, a spelling game. Just like liver. Good for you, but no fun." I knew the idea behind *Wordgo* was to make a word by combining blends and digraphs at the beginning of a word (like "ch" or "sch") with phonograms at the end of a word (like "asm" or "isin"). But I was in for a surprise as I began playing and found the game challenging not only my word recognition skills, but my game strategy skills as well.

After loading the disk into your Atari Home Computer, choose one of two difficulty levels ("Greenhorn" or "Strategist") and a one- or two-player game. Press the START key and a grid of 16 boxes, each filled with a two- or three-letter word ending (like "unk"), appears on the screen. A two- or three-letter word beginning (like "sk") is displayed at the top of the screen, and a timer in the top left corner begins ticking. You've got 30 seconds to make a

word by combining the word beginning with one of the 16 word endings. Using your joystick to move around the grid, position the cursor over the word ending ("unk") that you think will form a word when combined with the word beginning ("sk"). Press the FIRE button and *bingo!* I mean *Wordgo*. "Skunk" flashes at the top of the screen. A new word beginning appears, and the box containing the word ending turns blue. Your score—based on how long all of this takes you—appears at the top right corner of the screen. If you try to make a word that doesn't exist (like "skank"), you hear a low beep. The clock continues to tick away while you move on in search of a word. If you don't make a word before your 30 seconds are up, again you hear the beep, and a new word beginning appears at the top of the screen. You also lose 10% of your total score.

Pretty easy, right? But now the game gets really interesting. While using word endings to create those blue-outlined boxes, you're not only scoring points, you're also playing a kind of tic-tac-toe game. By lining up four blue boxes in a row—horizontally, vertically, or diagonally—you finish the game and score a *Wordgo*, which doubles your final score. You quadruple your final score by outlining two intersecting rows and double it again by outlining three intersecting rows (this is particularly tricky and requires a *lot* of patience).

If you think that no word can be made by combining the word beginning with any of the word endings on the grid, you can press the SELECT key on your computer. If you're right, you will be given points equal to the number of seconds left on the timer, and a new word beginning will appear at the top of the screen. If you're wrong, you'll lose 20% of your total score and the timer will continue to tick away until you make a word. This can be a costly error, so you might want to try for a word, even if you aren't sure if it's correct. There's no penalty for guessing, and you might surprise yourself—which brings us to an important feature of the game. If you do make a word you've never heard before, or if you just want to brush up on your vocabulary, continue to press your FIRE button after you've made a word, and a short definition will appear above the grid.

So if you're curious about the spelling and meaning of words, *Wordgo* is surprisingly fast, fun, *and* good for you. It manages to bridge the vast chasm (or is it schism?) between Scrabble and Centipede.—**Margaret Harrison**

Wordgo *is available now from Atari Program Exchange (APX). Diskette requires 40K RAM and ATARI BASIC Language Cartridge. Suggested retail price:* $24.95.

Margaret Harrison is an Assistant Product Marketing Manager for the Atari Products Company.

START YOUR ENGINES!

Introducing the Home Version of the Hit Arcade Game Pole Position

STEP INTO YOUR finely tuned, turbo-charged Formula 1 racer and get ready to test your road skill like never before!

First the qualifying run: You capture the pole position—the spot every driver dreams of—with an incredible lap time of 57.4 seconds. The other racers line up, engines roar, and you're off! You start out in low gear, pick up speed, then go for full-throttle acceleration

and blur past another car. Hugging the curves, you maneuver your car around the course. Use your joystick controller to shift gears, steer and brake. Test your skill and command of professional race car driving!—Teddi Converse

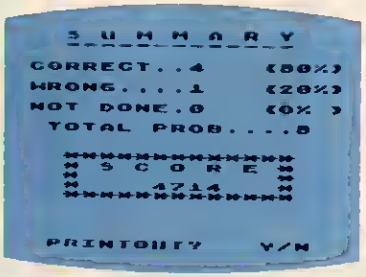
Cartridge format available now. Requires 16K RAM. Suggested retail price: $49.95.

THREE R MATH HOME SYSTEM

Bringing Home the Electronic Classroom

FOR GENERATIONS, young students have squirmed and sweated their way through an inescapable rite of elementary education: Arithmetic. It seemed there was no way to escape the necessary drills and rote learning that were an inevitable part of the educational process. And, with the introduction of "new" marh, few parents could really help their children at home. The different techniques used to solve the prohlems only served to confuse the children.

But, with the introduction of the *Three R Math Home System* tutorial program, adults and children can once again spend time together developing math skills.



The *Three R Math Home System* program offers two kinds of exercises, each accessed hy entering a code corresponding to your child's skill level. First comes a flashy routine called a "sprint" —a timed lesson designed to huild speed and confidence. You can select the numher of problems, the type of operation (addition, suhtraction, division, multiplication), and the number of digits (up to four) of each problem to he calculated.

The problems appear on the screen one at a time and the timer begins counting down. By answering the prohlems quickly and correctly, a score is generated, with happy sounds and congratulatory messages. The score and other information is printed out for your use. Children are encouraged to beat their old scores.

They can practice for the sprints by working on similar prohlems that can be printed out in worksheet form. When you print out a worksheet for a child, you can also print out an answer sheet for yourself. This makes it very simple to check rhe child's work.

The program's author, Dan Rohr, says that in order for the system to work, children need to learn math applications and reasoning. This is a well-written program with accurate and easy-to-read documentation. Adults and children will be ahle to increase computational skills through a shared learning experience.—Cassie Stahl

Three R Math Home System *is available now from Atari Program Exchange (APX). Diskette version requires 40K RAM and a* BASIC *cartridge. Suggested retail price: $24.95. This program is not compatible with the* Gradebook Program *in the* Three R Math Kit *also from APX.*

Cassie Stahl is a Customer Support Technician in the Customer Service Group.

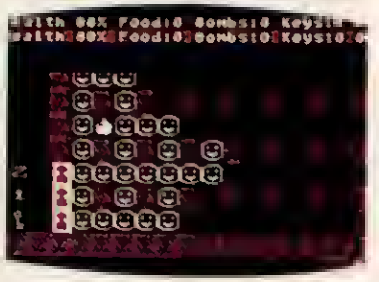
FINE AND DANDY

The Game That Even a Non-Gamer Learns to Love

I WANT TO LET YOU know right away that I'm not much of a game huff. To me, games are a source of amusement when there's nothing else better to do. Sort of like reading hillboards while riding cross country. When I was a kid, I enjoyed reading the Burma Shave signs that used to adorn the highways in the countryside— the ones that made funny little verses a line at a time, five signs to a set. Iambic dimeter or something. The last one always said "Burma Shave." We tried to guess the next phrase before we could read it. When there weren't any signs to read, we made up Burma Shave verses, the sillier the better.

You might wonder what that has to do with computer games. Well, I have the idea that a game is a game, whether played out a car window or through a TV set. I'm not wild about most games because I have a short attention span; I get bored easily. The ideal game for me is one that is immediate enough to keep my attention, bigger than I can see at a glance, and direct enough that I don't have to read a lot of instructions to play.

Dandy, by John Palevich, passes on all counts. It's a scrolling maze/adventure game TWENTY-SIX dungeon levels deep that you play with joysticks. The combination of "skill and action" play with a dungeon game structure is a winner! Wandering around a dungeon nine times as large as the TV screen keeps you on your toes. Also, *Dandy* can he played as a team sport—an unusual feature in video games. Playing with partners adds a lot to the quest, and having all those levels to play in keeps the game very interesting. Best of all, if you finally get bored with all those dungeons, you can make your own! *Dandy* includes instructions for making and saving your own dungeons.



The game play itself has a lot of subtleties. They are more fun to discover for yourself, hut I can't resist giving clues.

First, the monsrers can see through the walls and will follow you.

The second one is a little verse I made up all by myself:

WHEN WANDERING
THE DEPTHS OF DANDY
AN EXTRA BOMB
CAN COME IN HANDY
BURMA SHAVE

—Earl Rice

Dandy *is available in diskette format from Atari Program Exchange (APX). Requires 40K RAM; one to four joysticks. Retail price: $29.95.*

Earl Rice is the Manager of Atari Users' Group Support.

LET YOUR FINGERS DO THE TALKING

Fingerspelling Offers New Ways to Communicate

NOWADAYS, SIGNING—THAT IS, using sign language to communicate —is a standard practice for the deaf and hard of hearing, and is becom-

ing increasingly popular at public functions and on television. People who work with the hearing-impaired find sign language indispensable. It can also be very—pardon the pun—handy when you're at a rock concert, riding with your motorcycle group, or scuba diving.

Now Atari Program Exchange (APX) is offering a unique new educational program that teaches you fingerspelling, the first step in learning sign language.

Fingerspelling is a self-prompting program. It opens by graphically displaying how each of the alphabet letters is formed, accompanied by the "Alphabet Song." You first choose the letters you would like to see, and begin to memorize them. They you type in whole words, and the computer fingerspells the word back to you. Your words and letters can be fingerspelled back to you at varying speeds, represented by animals—as slow as a snail or as fast as a cheetah—an appealing feature for younger users. The program's testing option lets you know how well you're doing, by ranking your proficiency with either letter or words.

It takes a little time to learn to "hand signal" the alphabet, but once you learn, it's fairly easy to build up speed with practice. What delighted me most about this program was being able to practice reading the hands. Doing it yourself is one thing, but trying to recognize the patterns on somebody else's hand is a neat trick. Fortunately, the computer is a vey patient teacher and lets you learn at your own speed.

APX invited two instructors from Stanford University, who are themselves deaf, to evaluate the *Fingerspelling* program, and both were impressed with its accuracy and practicality. Signing actually involves gestures that represent entire words or expressions, so just knowing how to fingerspell won't entirely break the communication barrier between you and a hearing-impaired person. But it's a useful skill presented in an entertaining way.

APX is also now offering an *Advanced Fingerspelling* program. This program teaches you to fingerspell entire sentences, and lets you input various sentences and save them, to be played and practiced with whenever you want.

—Janet Lee Littlefield and Teddi Converse

Fingerspelling is available now from APX on diskette and cassette. Requires 24K RAM (diskette) and 16K RAM (cassette); joystick. Suggested retail price: $24.95.

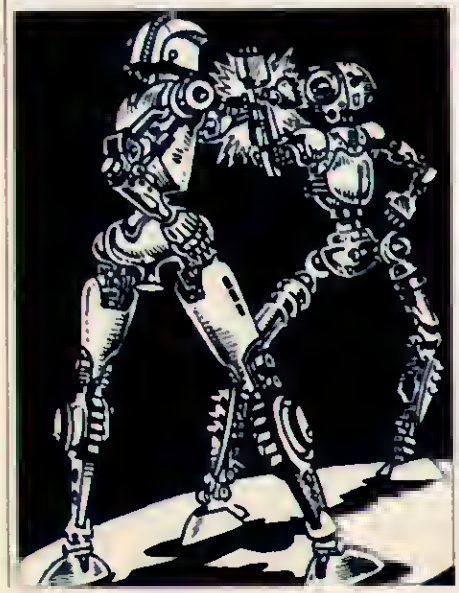
Janet Lee Littlefield is a Sales/Marketing Assistant and Teddi Converse is a Writer in the Creative Services Department of Atari Products Company.

ROBOT WARS!

In Robotron: 2084, the Human Race Depends on YOUR Fire-Button Thumb

RED ALERT!! The Robotrons—a highly intelligent species of robots—are on a mission to destroy humanity! Recenrly intercepted Robotron communiqués reveal that only a handful of humans remain alive on earth! And only you, as a result of a genetic error, are immune to their reprogramming. The fate of human life rests with you! Can you save earth's last survivors?

Use your joystick controllet to fire your anti-robot laser gun in eight directions. Dodge oncoming Grunts, and maneuver yourself around Hulks, Spheroids, Enforcers and Quarks. Race to save the defenseless humans before grotesque Brains transform them into freakish mutant Progs. The action is fast and the game play challenging. Choose from four different levels, and use with one or two joysticks for one- or two-player games.—Teddi Converse



Cartridge program available now. Requires 16K RAM. Suggested retail price: $44.95.

THE ATARI MUSIC LEARNING SERIES

How to Master Music Theory Without Missing a Beat

LEARNING MUSIC can require a cumbersome collection of instruments, instruction books, and even bodies. For starters, there's the well-tuned 900-pound piano—or tiny $700 violin to worry about losing between home and class. Then, you need music sheets, lesson primers, staff-lined notebooks, and a well-sharpened pencil. Ideally, you have a music instructor. And finally, there's Mom and Dad—the people who write the checks to the music teacher—constantly encouraging you to practice, so they can get *their* money's worth.

Well, don't sell the piano yet. There's still no replacing a fine musical instrument, or a good teacher. But with a new tutorial program, you can make the most of your lessons, with a minimum of fuss. The program, called *The Atari Music Learning Series*, runs on your Atari Home Computer and uses —naturally—your joystick.

The Atari Music Learning Series makes the perfect music teacher for several reasons: It gives you immediate feedback on all your answers, it utilizes the full musical capabilities of your Atari Home Computer, including the four programmable "voices" and full range of pitches, and it lets you learn at your own pace and on your own rime. And, when you're ready for it, *The Atari Music Learning Series* tests you more enticingly than your friendly music teacher probably could—with outer-space video games.

Right now you can purchase two separate program levels of *The Atari Music*

CHEMICAL WOULD LIKE TO OPEN A BRANCH IN YOUR LIVING ROOM.

At Chemical, we've figured out a way to provide banking with all the comforts of home.

ANNOUNCING PRONTO. THE HOME BANKING SYSTEM.

Simply stated, Pronto offers money management and banking services 24 hours a day at home. At your convenience. Not ours.

In fact, with the help of your personal computer, TV and telephone, you'll be able to do most of your banking without ever entering the bank.

For example, with Pronto, you can check your balance. Or check which checks or deposits have cleared. And automatically reconcile your checkbook.

Pay bills to over 350 different merchants and services. Without writing a check or buying a stamp.

Or transfer money from one account to another.

There's also a feature called electronic mail. It can answer all your banking questions by putting you in direct contact with our Customer Service staff.

What's more, Pronto's budgeting system allows you to plan and control your finances while keeping track of all your expenses.

And no matter what kind of business you conduct on Pronto, it's strictly your business. You'll be the only one with access to your financial information.

CALL US OR COME IN FOR A DEMONSTRATION.

To see the new Pronto system in action, come to one of the Chemical Bank locations listed at the bottom of this ad. Or call us toll free at 1-800-782-1100.

Just think, this might be the last time we'll ask you to leave home to go to the bank.

PRONTO

THE HOME BANKING SYSTEM FROM CHEMICAL BANK.

MANHATTAN: 100 World Trade Center, 756 Broadway, 2 Penn Plaza, 277 Park Ave., 1350 Ave. of the Americas, 377 Madison Ave., 11 W. 51st St., 622 W. 168th St.
LONG ISLAND: 410 Northern Blvd., Great Neck, 350 Main St., Huntington, 10 North Village Ave., Rockville Centre
BROOKLYN: 50 Court St. QUEENS: 107-36 71st Ave., Forest Hills WESTCHESTER: 676 White Plains Rd., Eastchester, 222 Mamaroneck Ave., White Plains

Learning Series: AtariMusic I and *AtariMusic II*. Both products include two-diskette programs with easy-to-follow menus and detailed descriptive text.

AtariMusic I:

The main menu of *AtariMusic I* is divided into lessons on "Note Reading" and "Whole and Half Steps." In the former, you learn about the positions of all notes (including sharps and flats) on both the treble and bass clefs. By using the joystick, you can manipulate the position of notes on the staves while simultaneously *hearing* them.

In the "Whole and Half Steps" mode, you learn to fine-tune your sight reading; the computer displays (and plays) two notes, and you determine whether what you saw is a whole or half step.

It takes awhile to develop the skills to master the two space games in *AtariMusic I*. In "Note Attack," musical note-missiles follow the lines and spaces of the staff heading toward your rocket. Indicate the correct note with your joystick to blow them up before they attack. In the "Stepwise Transporter Game" you distinguish between whole and half steps to move cargo up and down the scale between home base and your space shuttle. Wrong answers wear away your shuttle's protective shields.

AtariMusic II:

By the time you've mastered *AtariMusic I*, you're ready to learn about the major scales and key signatures —and that's exactly what you'll do in *AtariMusic II*.

The first mode, "Major Scales," teaches you how to name and write key signatures in all the major scales.

In "Hearing Scalewise Melodies in Major Keys," the second mode of *AtariMusic II*, you start training your ear to hear melodies in the major keys. To successfully complete the drills in this lesson, you learn to name the notes in a melody using the letter names of notes, scale degree numbers, the keyboard, or solfeggio syllables—the DO, RE, MI of the musical alphabet.

These lessons are recommended for ages eight to adult and provide a great deal of learner interaction, allowing for both recall and exploration. But because the programs are heavily text-oriented in places, they may not move fast enough for very young children.

The Atari Music Learning Series was developed by Dr. Fred T. Hofstetter, Professor of Music and Educational Studies at the University of Delaware, a pioneer in the application of computers to music instruction. Teachers will appreciate the program's sound pedagogical presentation. And, music theory students will be glad to leave those violin cases at home once in awhile, and instead pack their featherweight diskette holders.—**Teddi Converse and Paula Polley**

Available on diskette. Requires 24K RAM; joystick. Suggested retail price: $39.95 each.

Paula Polley is a Writer for the Creative Services Department of the Atari Products Company.

MORSECODE MASTER

Learn Your SOS's and Be at Ease with Your Teacher

THERE'S A SCENE in a vintage World War II movie in which a Leatherneck captain, tough as a crocodile but with a heart as big as Parris Island, is putting his men through their paces. Forty recruits, hunched over radio receivers, are trying to keep up with the instructor as he taps out Morse Code. One man falters and the captain snaps, "There's no room in the Marines for quitters! Your life may depend on this!"

Morse Code hasn't changed much since Samuel F.B. Morse—the last person to use the telegraph without wiring for money—thundered, "What hath God wrought?" 140 years ago. Mr. Morse's code is still an important tool, and it still must be learned through hands-on experience, by hearing and deciphering those nonstop dots and dashes. But today, you can master the first form of electronic communication by using that marvel of 20th Century technology, the computer.

Morsecode Master is a tutorial program designed to simulate code signals, transmit them in increasingly complex forms, and inform students of their progress. Aspiring radio and telegraph operators are no longer at the mercy of a drill instructor, and can learn at their own pace. The various lessons start with single-character code recognition, and finish with entire sentence-code recognition.

The program begins by asking your name, and addresses you personally with each prompt. Very friendly.

The entire program is menu driven —any option can be selected by pressing a few keys as indicated on the screen. There are five exercises, from beginning to advanced. Lesson A allows you to hit any of the character keys on the keyboard and hear the Morse Code for that character through the TV or monitor speaker. The next step —Lesson B—lets you listen and respond to small groups of characters from the Morse alphabet: vowels, common consonants, other consonants, numbers, and punctuation. After choosing a category, you are asked to press the key corresponding to each character code heard. A correct answer turns the screen green and repeats the code you've identified. An incorrect answer gives you a red screen and a "Try Again" prompt. You may choose to have the code repeated, get the correct answer, or exit back to the Main Menu. The bottom of the screen displays the number of correct answers, the number of tries, and the number of wrong answers.

Lesson C is a drill for reviewing what you've learned, without being able to choose the category from which the code will come.

When you're ready to practice whole words, Lesson D lets you hear the codes for each letter of a word in sequence. When the code is finished, type the word into the computer. You get the now-familiar green and red responses for correct and incorrect answers.

Finally, when you've mastered single-word codes, Lesson E sends you whole sentences. Again, the computer tells you how well you did.

As you get better at recognizing the codes, you can turn up the speed, increasing the number of words sent per minute. Another option allows you to change the pitch of the tones to suit your taste.

That's something you won't find in the Marine Corps.—**Kent Smith**

Morsecode Master *is available now from Atari Program Exchange (APX). Requires 24K RAM (cassette), 32K RAM (diskette); ATARI BASIC Language Cartridge. Suggested retail price: $29.95.*

Kent Smith is a Technical Representative for the Customer Service Department.

COMPUTER CLASSROOM

by Bill Bartlett

Unravelling the Mysteries of Atari Screen Graphics

THE ATARI COMPUTER'S SCREEN is where all the action takes place. The more you know about your Atari Computer's display system and how to use the ATARI BASIC language to make it do what you want it to do, the more excitement you'll experience.

First, I'll take you on a short guided tour through your Atari Computer's display system, then show you three fun-filled programs that create images on your screen using *Character Set Graphics, Player/Missile Graphics*, and *Map Mode Graphics*.

Your TV's Display

When you connect an Atari Computer to a television set, it takes control of the display. But, before you can truly understand how your computer takes control, you should know how a television tube works.

If you live in North America, your television creates a picture by tracing 262 horizontal "scan lines" from the upper left-hand corner to the lower right-hand corner of tbe picture tube, 60 times per second. Each scan line contains 228 individual "color clocks" that can each display a single dot of color.

European televisions trace 312 scan lines every 1/60 of a second and have a higher resolution than sets here in the United States. It would be distracting, to say the least, if these scan lines and color clocks were visible, so the screen is deliberately overscanned to eliminate this unsightly activity.

ANTIC and GTIA

Your Atari Computer uses two custom-designed microprocessor chips, *ANTIC* and *GTIA*. These limit its display to 192 scan lines and 160 color clocks so that its displays fit neatly onto most television screens. But ANTIC and GTIA do much more. ANTIC defines wbat appears on your screen. ANTIC's definition of the screen's display is coded in its "Display List" (a list of instructions that tell your Atari Computer how to combine scan lines and color clocks to form picture elements called "pixels").

There are 17 built-in Display Lists that you can "call up" in ATARI BASIC by using the GRAPHICS command. Each of these 17 Display Lists uses a different-sized pixel to fill up a screen. GRAPHICS 0, for example, is made up of 24 rows by 40 columns of pixels, which equals 960 pixels.

You can program ANTIC's Display List to create your own custom display. This technique is often used to display more than one GRAPHICS mode on the screen at a time. By mixing *Character* Graphics Mode 1 with *Pixel* Graphics Mode 3, for example, you can simultaneously display both characters and drawings on your screen.

GTIA determines the color and luminance of the pixels that are "bit-mapped" to the screen. In bit-mapped graphics, each screen pixel is related to a number of bits in RAM that assign it a color and luminance.

GTIA cannot assign each of the 256 available colors to its own special screen pixel. The memory required to do this would be an extra 30,720 bytes (30K) of RAM. Instead, the bits in RAM define the color of four separate "color registers." This method reduces the memory requirements to a maximum of 7,680 bytes but only allows BASIC to display four colors on the screen at a time.

Character Set Graphics

Your Atari Computer comes equipped with a Character Set built into ROM (Read Only Memory). The Character Set is a bit-map table that determines how pixels are shaped into the letters, numbers and symbols that appear on your screen. Each character's shape requires 8 bytes of memory, and because there are 128 distinct character shapes, the size of the built-in Character Set is 1024 bytes.

You can "look" at the bytes that make up your Atari Computer Character Set by ENTERing then RUNning the "PEEKER" program listed in the "Bits and Pieces" section. The Character Set begins at location 53744.

To display the built-in Character Set from BASIC you'll use the COLOR, SETCOLOR, PLOT, DRAWTO, POSITION and PRINT commands.

Character Redefinition and Animation

You can make a copy of the Atari Computer Character Set, then shape and redefine any character to resemble almost anything you'd like. You could use this feature to make your own custom-designed alphabet (called a "font") or, if you like action, you can redefine a few characters and "flip" between them to create animation. The *Talking Boxes* program listing below uses a partially redefined Character Set to create an "animated box."

Line 40 sets aside protected memory for the redefined Character Set.

Line 210 makes a copy of the uppercase portion of the Atari Computer's Character Set (512 bytes) and puts it into a protected memory area directly below Screen Graphics memory. After a copy of the Character Set is placed in RAM, the shape of the characters A, B, C and D are redefined as different sized-boxes.

Line 220: POKE 756,RAM/256, "tells" your computer where your redefined Character Set is. Then, in lines 100 through 180, the redefined characters are PRINTed repeatedly on the screen to give the illusion of animation.

TALKING BOXES

```
10 . Character Set Graphics
20 . Animation
30 GRAPHICS 2:POKE 752,1:
   POSITION 7,0:PRINT #6;"ABCO":
   ? "ABCOEFGHIJKLMNOPQRSTUVWXYZ"
32 .
36 . Set aside protected memory
   for redefined Character Set.
38 .
40 MEMTOP=PEEK(741)+256*PEEK(742)-1
50 RAM=INT((MEMTOP-1024)/1024)*1024
60 ADJTOP=RAM
70 POKE 742,INT(ADJTOP/256):
   POKE 741,ADJTOP-256*PEEK(742)
75 .
80 ? "MOVING CHARACTER SET TO
   RAM...":GOSUB 190
90 ? "REDEFINING LETTERS
   A,B,C,D...":GOSUB 230
100 POSITION 9,5:? #6;"A":
    FOR DELAY=1 TO 20:NEXT DELAY
110 POSITION 9,5:? #6;"B":
    FOR DELAY=1 TO 20:NEXT DELAY
120 POSITION 9,5:? #6;"C":
    FOR DELAY=1 TO 20:NEXT DELAY
130 POSITION 9,5:? #6;"D":
    FOR DELAY=1 TO 20:NEXT DELAY
140 POSITION 9,5:? #6;"D":
    FOR DELAY=1 TO 20:NEXT DELAY
150 POSITION 9,5:? #6;"C":
    FOR DELAY=1 TO 20:NEXT DELAY
160 POSITION 9,5:? #6;"B":
    FOR DELAY=1 TO 20:NEXT DELAY
170 POSITION 9,5:? #6;"A":
    FOR DELAY=1 TO 20:NEXT DELAY
180 GOTO 100
185 .
190 . Copy Character Set from ROM
    to RAM.
195 .
200 ROM=57344
210 FOR I=0 TO 511:POKE
    RAM+I,PEEK(ROM+I):NEXT I
220 POKE 756,RAM/256:
    RETURN :. Point to redefined
    character's location.
225 .
230 . Redefine the shapes of
    A,B,C and D.
235 .
240 . A = Character Set position
    33. (ATASCII 65-32)
250 ASHAPE=RAM+33*8:FOR
    I=ASHAPE TO ASHAPE+7:READ
    J:POKE I,J:NEXT I
260 DATA 255,129,129,129,129,129,
    129,255
```

```
270 . B = Character Set
    position 34. (ATASCII 66-32)
280 BSHAPE=RAM+34*8:FOR I=BSHAPE TO
    BSHAPE+7:READ J:POKE I,J:NEXT I
290 DATA 0,126,66,66,66,66,126,0
300 . C = Character Set
    position 35. (ATASCII 67-32)
310 CSHAPE=RAM+35*8:FOR I=CSHAPE TO
    CSHAPE+7:READ J:POKE I,J:NEXT I
320 DATA 0,0,60,36,36,60,0,0
330 . D = Character Set
    Position 36. (ATASCII 68-32)
340 DSHAPE=RAM+36*8:FOR I=DSHAPE TO
    DSHAPE+7:READ J:POKE I,J:NEXT I
350 DATA 0,0,0,24,24,0,0,0
360 RETURN
```

Player/Missile Graphics

Player/missile graphics provide another exciting method for creating screen graphic images. Each Atari Home Computer has the provision for four *players* and *missiles* that can be thought of as vertical bars of colors extending from the top to the bottom of the screen.

The GTIA chip provides many embellishments for players/missiles that allow you to create realistic and enchanting animation. These embellishments are accessed by PEEKing and POKEing the appropriate registers.

Each player has its own color register. This lets you display four unique colors plus the four playfield (character and map mode) colors—a total of eight vibrant colors on your screen at one time! Missiles are the same color as their corresponding player. (Missile 1's color is the same as player 1's color, and so forth.)

Each player/missile has its own horizontal position register. This allows horizontal movement with a single POKE command. Each player also has its own *size register* that lets you display it in normal, double or quadruple resolution. There is also a priority register that can be set to display players in front or in back of playfield graphics. This register is used to achieve three-dimensional screen displays. The final player/missile embellishment is the easy detection of collisions between players, missiles and playfield objects by PEEKing into the collision registers.

Each Player has a "graphics" register that defines its vertical shape. Normally, the player/missile graphics registers are loaded from RAM automatically by ANTIC using a process called *Direct Memory Access* (DMA). Using DMA to define a player's vertical shape requires setting aside up to 2K of player/missile RAM, POKEing in the player's vertical shape definition of up to 256 bytes, and POKEing the correct values into the appropriate control registers. (Refer to "Player/Missile Graphics, Right Side Up!" in *Bits and Pieces* for a complete explanation of this technique.)

Player/Missiles Without DMA

Player/missile graphics may be used without DMA if a vertical bar of color is all that is desired. Each bit that is set in a player's graphics register (POKE) will appear as a vertical line on the screen. Assigning a value of 255 creates a wide vertical line. Other patterns can be chosen by POKEing the player graphics register with any value from 0 to 255. The *Player/Missiles Without DMA* program below demonstrates this simple method of using player/missile graphics.

This colorful program first displays the players in quadruple, double and normal size. It then moves each vertical player across the screen. To demonstrate three-dimensional player/missile effects, the *Prioriy Register* (623) is set to 4 in line 20 during the program's first cycle to give tbe *Playfield* priority over all players. When the first cycle is completed, the Priority Register is reset to 0 (line 270), and in each subsequent cycle it is reset to 1, 2, 3, 4 then back again to 0.

PLAYER/MISSILES WITHOUT DMA

```
10 . Player-Missiles without DMA
20 PRIORITY=4: . Set Priority
   register to 4
30 GRAPHICS 2:POKE 710,0:POKE 752,1
40 FOR X=0 TO 19:FOR Y=0 TO 9:
   POSITION X,Y:PRINT #6;CHR$(35):
   NEXT Y:NEXT X
50 GOSUB 290:GOSUB 370
60 POKE 623,PRIORITY
62 .
64 . Display each Player's size.
66 .
70 POKE P0SIZE,3:POKE P1SIZE,3:
   POKE P2SIZE,3:POKE P3SIZE,3
80 ? "QUADRUPLE PLAYER SIZES...":
   FOR DELAY=0 TO 1500:NEXT DELAY:
   ? CHR$(125)
90 POKE P0SIZE,1:POKE P1SIZE,1:
   POKE P2SIZE,1:POKE P3SIZE,1
100 ? "DOUBLE PLAYER SIZES...":
    FOR DELAY=0 TO 1500:NEXT DELAY:
    ? CHR$(125)
110 POKE P0SIZE,0:POKE P1SIZE,0:
    POKE P2SIZE,0:POKE P3SIZE,0
120 ? "NORMAL PLAYER SIZES...":FOR
    DELAY=0 TO 1500:NEXT DELAY:
    ? CHR$(125)
122 .
124 . Move each Player.
126 .
130 ? "PLAYER 0 POSITION"
140 FOR I=50 TO 255:POKE P0POS,I:
    ? I;CHR$(28):NEXT I:? CHR$(125)
150 FOR I=255 TO 0 STEP -1:
    POKE P0POS,I:NEXT I
160 ? "PLAYER 1 POSITION"
170 FOR I=90 TO 255:POKE P1POS,I:
    PRINT I;CHR$(28):NEXT I:
    ? CHR$(125)
180 FOR I=255 TO 0 STEP -1:
    POKE P1POS,I:NEXT I
190 ? "PLAYER 2 POSITION"
200 FOR I=130 TO 255:POKE P2POS,I:
    PRINT I;CHR$(28):NEXT I:
    ? CHR$(125)
```

```
210 FOR I=255 TO 0 STEP -1:
    POKE P2POS,I:NEXT I
220 ? "PLAYER 3 POSITION"
230 FOR I=170 TO 255:POKE P3POS,I:
    ? I;CHR$(28):NEXT I:? CHR$(125)
240 FOR I=255 TO 0 STEP -1:
    POKE P3POS,I:NEXT I
250 POKE P0POS,50:POKE P1POS,90:
    POKE P2POS,130:POKE P3POS,170:
    FOR DELAY=1 TO 500:NEXT DELAY
252 .
254 . Cycle Priority Register.
256 .
260 PRIORITY=PRIORITY+1
270 IF PRIORITY>4 THEN PRIORITY=0
280 GOTO 30: . Begin again.
285 .
290 . Equate Player variables.
295 .
300 RESTORE 310:READ P0COLOR,P0POS,
    P0SIZE,P0SHAPE
310 DATA 704,53248,53256,53261
320 P1COLOR=P0COLOR+1:P2COLOR=
    P0COLOR+2:P3COLOR=P0COLOR+3
330 P1POS=P0POS+1:P2POS=P0POS+2:
    P3POS=P0POS+3
340 P1SIZE=P0SIZE+1:P2SIZE=
    P0SIZE+2:P3SIZE=P0SIZE+3
350 P1SHAPE=P0SHAPE+1:P2SHAPE=
    P0SHAPE+2:P3SHAPE=P0SHAPE+3
360 RETURN
265 .
370 . Initialize Player values.
375 .
380 POKE P0COLOR,42:POKE P0POS,50:
    POKE P0SHAPE,255
390 POKE P1COLOR,200:POKE P1POS,90:
    POKE P1SHAPE,255
400 POKE P2COLOR,110:POKE
    P2POS,130:POKE P2SHAPE,255
410 POKE P3COLOR,152:POKE
    P3POS,170:POKE P3SHAPE,255
420 RETURN
```

Map Mode Graphics

In addition to the six character modes, ANTIC has eight "map" modes ($8-$F). Six of these map modes are available from ATARI BASIC (GRAPHICS 3– 8). The Display Lists for all eight map modes are built into the Atari XL series of Atari Home Computers. There are also three GTIA modes that are variations of ANTIC mode $F (GRAPHICS 9–11).

A map mode displays a screen full of tiny colored pixels. Each map mode has a certain resolution (pixel size) and allows either two or four colors. All map modes have their pixel's color mapped directly from the screen RAM. The amount of screen RAM required increases as the resolution (smaller pixels) and number of colors increases.

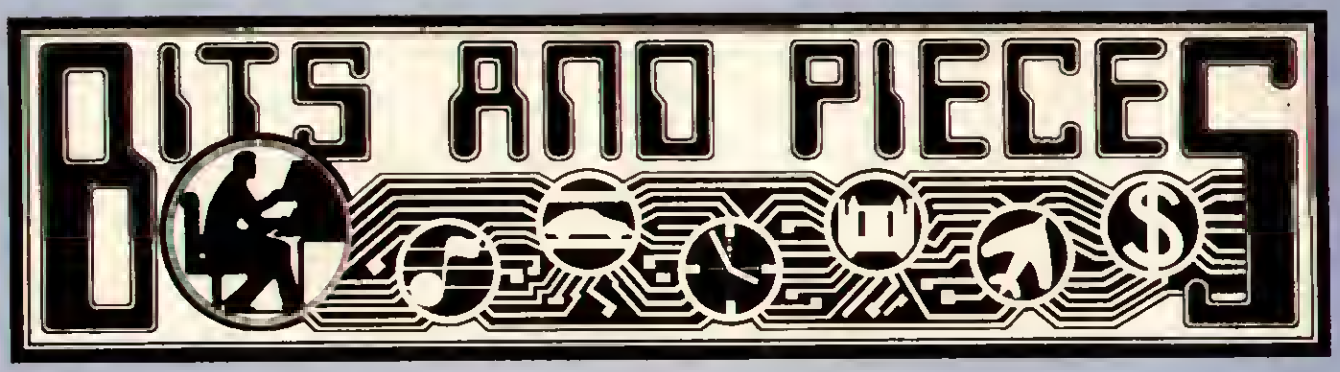
To create map mode graphics from BASIC, the following commands are used: COLOR, SETCOLOR, PLOT, DRAWTO, and PRINT. The *And They're Off!* program below is a racing game that uses map mode graphics. Type it in, RUN it and enjoy a day at the races!

"AND THEY'RE OFF!"

```
10 . Map mode graphics.
20 GRAPHICS 2+16:. NO TEXT WINDOW
30 POSITION 0,5:? #6;
   "SELECT # OF PLAYERS"
40 OPEN #1,4,0,"K:":GET #1,PLYRS:
   CLOSE #1:. READ A KEYSTROKE
50 PLYRS=PLYRS-48:IF PLYRS<1
   OR PLYRS>9 THEN 40
60 GRAPHICS 2+16:POSITION 0,4:
   ? #6;"PAY YOUR DUES":POSITION
   0,6:? #6;"start TO CONTINUE"
70 IF PEEK(53279)<>6 THEN GOTO 70
80 POSITION 0,8:? #6;"AND THEY'RE
   OFF..":FOR DELAY=0 TO 250:
   NEXT DELAY
82 .
84 . Map mode
86 .
90 GRAPHICS 3+16:POKE 712,195:
   . FULL SCREEN MAP MODE
100 DIM DISTANCE(9):FOR I=0 TO 9:
    DISTANCE(I)=0:NEXT I:
    . INITIALIZE
110 SETCOLOR 0,3,4:SETCOLOR 1,0,14:
    . RED, WHITE
120 . PERFORM RACE
130 RACER=INT(RND(10)*PLYRS)+1:
    DISTANCE(RACER)=DISTANCE(RACER)
    +1:. ADVANCE
140 HORZ=DISTANCE(RACER):VERT=
    RACER*2:. COORDINATES ON SCREEN
150 KOLOR=RACER:IF KOLOR>3 THEN
    KOLOR=KOLOR-3:IF KOLOR>3 THEN
    KOLOR=KOLOR-3
160 COLOR KOLOR:. ASSIGN COLOR
170 PLOT HORZ,VERT:SOUND
    KOLOR,40-HORZ,10,6
180 IF HORZ<39 THEN GOTO 120
190 . END OF RACE
200 FOR DELAY=0 TO 255:NEXT DELAY
205 .
210 GRAPHICS 2+16:POSITION 0,5:
    ? #6;"THE WINNER IS #";RACER
220 FOR DELAY=0 TO 500:NEXT DELAY:
    RUN
```

I hope you have enjoyed our short tour through Atari's Screen Graphics. Put the techniques you've learned here into your programs and utilize the full power of your Atari Home Computer. Here's a supplementary reading list that will help you explore your Atari Computer and its screen graphics capabilities.

Technical Users' Notes, Atari, Inc.
De Re Atari, Atari Program Exchange
Mapping the Atari, COMPUTE! Books



IN THE PUBLIC DOMAIN

by David L. Heller

YOUR ATARI COMPUTER IS the best home computer on the market today, and I'm not the only one who thinks so. Thousands of enthusiastic Atari Computer users agree. They've spent hours of tedious thought and time to develop programs that help you get more from your Atari computing experience. After these programs are finalized and tested, they're donated to the growing program-banks of the more than 250 Atari Users' Groups throughout the wotld. Now, these programs are in the "Public Domain" —another users' gift to you!

The Atari Computer Enthusiasts (A.C.E.) of Eugene, Oregon, have compiled an extensive library of useful programs (called "Utilities") that put you in touch with the power of your Atari Computer system by easing your programming chores and organizing your computer files. I've drawn upon the resources of the A.C.E. program-bank ro write this issue's "Bits and Pieces." You can put these programs ro practical use immediately. They're guaranteed to enrich your computing experience!

HELP
Electronic Documentation

by Stacy Goff

Almost every public domain software diskette that the A.C.E. Users' Group mails to fellow Arari Computer owners is fully documented. But, you don't get pieces of paper—it's electronic! The documentation appears on your monitor's screen, because it's written on the disk itself!

Stacy Goff has developed "HELP," a complementary set of programs to make electronic documentation possible. If you have lots of programs on a disk to share with others, you can use Stacy's HELP programs to make the recipient's (and your) life a lot easier. I'm sure you'll find many applications for these two programs. Bur, before you can rurn on your creative juices, you've got to see this pair in action.

The first program in this set is called "HELP.WRT." It lets you write documentation for each of the files on your disk. When you're finished, it automatically creates a data file on your disk that is read and then displayed on your screen by the "HELP" program. Enter the HELP.WRT program, then SAVE it to your disk by typing: SAVE "D:HELP.WRT" [RETURN].

HELP.WRT

```
0 . An A.C.E. Program.
5 DIM F$(12),A$(40)
10 ? "NAME YOUR FILE":? "(ENTER
   AS D:PGMNAME.HLP)"
20 INPUT F$
30 OPEN #1,4,0,"E:":OPEN #2,8,0,F$
40 ? " Press RETURN to end each
   line. ------ Press START +
   RETURN to end file.":?
50 FOR X=1 TO 37:GET #1,A:IF
   PEEK(53279)=6 THEN GOTO 100
60 IF A=155 THEN POP :GOTO 80
70 A$(X)=CHR$(A):NEXT X:
   A$(X)=CHR$(155):FOR X=40 TO 1
   STEP -1:SOUND 0,20,4,INT(X/4):
   NEXT X:SOUND 0,0,0,0:GOTO 90
80 FOR X=40 TO 1 STEP -1:
   SOUND 0,20,14,INT(X/4):NEXT X:
   SOUND 0,0,0,0
90 PRINT #2;A$:FOR X=1 TO 38:
   A$(X)=" ":GOTO 50
100 ? #2;A$:CLOSE #1:CLOSE #2:END
110 ? :? :? "END OF FILE":CLOSE #1:
    END
```

Before you RUN "HELP.WRT," type in and SAVE its complementary program *HELP.* I'll then show you how to use both programs in concert to electronically document all the programs on your disk. Enter the *HELP* program, then SAVE it to your disk.

HELP

```
0 . An A.C.E. Program.
10 DIM A$(40)
20 TRAP 100:OPEN #1,4,0,A$:TRAP 200
30 ? CHR$(125):FOR X=1 TO 20:INPUT
   #1;A$:PRINT A$:NEXT X:? :
   ? " PRESS RETURN ";:INPUT
   A$:GOTO 30
40 CLOSE #1:? "WHAT HELP FILE":?
   "(ENTER AS D:PGMNAME.HLP) ":
   INPUT A$:GOTO 20
50 ? :? :? "END OF FILE":CLOSE #1:
   END
```

```
100 CLOSE #1:? "WHAT HELP FILE":
    ? "(ENTER AS D:PGMNAME.HLP) ":
    INPUT A$:GOTO 20
200 ? :? :? "END OF FILE":CLOSE #1:
    END
```

Now that you have both programs tucked safely away on your diskette, it's time to have some fun!

I'll walk you through a sample application of the *HELP* combo. Then use your creativity and this great utility to write your own electronic documentation.

Writing Documentation

1. RUN the *HELP.WRT* program.

2. You'll be asked to "NAME YOUR FILE." Just to humor me, answer this prompt with: D:TEST.HLP [RETURN].

3. Now you can start writing your documentation. Hit RETURN at the end of each line. Enter as much text as you'd like, the more the merrier. When you think you've said enough, press the [START] key and [RETURN] at the same time to end your file.

That's all there is to it! You've just created a data file (filled with great prose) named "TEST.HLP" on your diskette. Now it's time to look at the great documentation you've created.

Reading Your Creation

1. RUN the *HELP* program.

2. When you are asked, "WHAT HELP FILE?" answer with: D:TEST.HLP [RETURN].

3. Sit back, relax and warch your carefully structured prose flow onro the screen. If you're prolific and have written more rhan one screen-full, you'll be asked to "PRESS RETURN" to continue to the next page.

AUTORUN.BLD
Create an AUTORUN Program

Provided by

The Jacksonville Atari Group

Now that you've documented your programs, here is a program to create a file that will automatically RUN your documentation (or any other program) when your Atari Computer is turned on! Enter this program, then SAVE it to your disk as: "AUTORUN.BLD."

AUTORUN.BLD

```
 0 . PROVIDED TO A.C.E BY THE
   JACKSONVILLE ATARI GROUP
 5 . AUTORUN.SYS FILE BUILDER
10 DIM A$(30),B$(20)
20 A$="RUN":A$(4)=CHR$(34)
30 B$="D:FILENAME.EXT"
35 A$(LEN(A$)+1)=B$
40 OPEN #1,8,0,"D:AUTORUN.SYS"
50 PUT #1,255
60 PUT #1,255
70 PUT #1,0
80 PUT #1,6
90 L=61+LEN(A$)-1
100 PUT #1,L
110 PUT #1,6
120 FOR I=1 TO 61
130 READ D
```

Submitting a Program or Subroutine to *BITS AND PIECES*

To prepare a program or subroutine for submission to *Bits and Pieces* follow these guidelines:

1. Give yout program or subroutine a name of eight or less characters.

2. Write a brief description of the task performed by your program or subroutine and describe any special features.

3. Write a few lines about yourself and how you came to develop your program or subroutine.

4. If you are a more experienced programmer submitting a subroutine, then compile the following important information:

- List the variables for which you need values when calling the subroutine (entry variables).
- List all DIMensioned variables with their sizes.
- List any IOCBs used in OPEN statements.
- List the variables which may be generated by the subroutine and passed back to the main program (exit variables).

5. Send a copy of your program on cassette or diskette, attach a printed listing of your program or subroutine, and mail to:

Bits and Pieces
c/o ATARI CONNECTION
P.O. Box 50047
San Jose, CA 95150

Even if you don't consider yourself an "expert" or "advanced" programmer, don't be discouraged! If you have created something neat or interesting, submit your program and explain its operation to the best of your ability. "Bits and Pieces" is for experts and beginners alike. So don't be shy. Even novices develop programs and routines that are of interest to others!

Editor's Note: *Last issue we asked you to submit your programs to "Bits and Pieces" and many of you responded with some very excellent material. We thank you. Unfortunately, everything has arrived here at our offices as we are producing this issue. But, don't despair. Next issue will feature the best of the many "Bits and Pieces" submissions we have received—needless to say, we'll be preparing a number of APX Gift Certificates! So, please continue to send your programs to "Bits and Pieces." We look forward to reviewing your programs and ideas.*

```
140 IF I=12 THEN PUT #1,LEN(A$):
    GOTO 160
150 PUT #1,D
160 NEXT I
170 FOR I=LEN(A$) TO 1 STEP -1
180 PUT #1,ASC(A$(I,I))
190 NEXT I
200 PUT #1,252
210 PUT #1,1
220 PUT #1,254
230 PUT #1,1
240 PUT #1,255
250 PUT #1,5
260 CLOSE #1
270 END
320 DATA 169,18,141,33,3,169,6,
    141,34,3
330 DATA 169,10,141,128,6,76,105,
    243,251,243
340 DATA 51,246,33,6,163,246,51,
    246,60,246
350 DATA 76,228,243,0,172,128,6,
    240,9,185
360 DATA 60,6,206,128,6,160,1,96,
    140,33
370 DATA 3,169,228,141,34,3,169,
    155,160,1
380 DATA 96
```

The best way to appreciate the power of this program is to use it in a real application. You've just written and created a documentation file named "TEST.HLP." So, let's use AUTORUN.BLD to create an AUTORUN.SYS file on your disk that will automatically display your documentation when you turn on your Atari Computers' power.

The AUTORUN.BLD program only works with DOS 2—so, before you proceed, make sure you have this Disk Operating system on your disk. All set? Here goes!

1. LOAD the AUTORUN.BLD program then LIST it nn your screen.

2. Change line 30 to read: 30 B$="HELP"

3. RUN the program. It will create a file named "AUTORUN.SYS" on your disk.

4. Replace "A$" in line 20 of your *HELP* program with "D:TEST.HLP" and SAVE the modified program to your diskette. Replacing "A$" makes the *HELP* program instantly RUN your documentation without waiting for your input.

Here's what your new line 20 will look like:

20 TRAP 100:OPEN #1,4,0,"D:TEST.HLP":TRAP 200

5. Turn off your computer. Be sure that your disk drive is on and the diskette you've just prepared is inserted and ready to go.

The moment of truth is here! Turn on your Atari Computer and (if you've carefully followed the instructions) watch your well written documentation flnw across your screen! Use this valuable public domain software to add that professional touch to your programs by creating an AUTORUN.SYS file that will automatically RUN any prn-gram of your choice.

LPDIR
Miniature Disk Directory
by
"Those Folks In Texas"

The following program was donated to the public domain by "Those Folks In Texas." I'm not sure if they're from Austin, Houston, Dallas, or possibly Armadillo, but wherever they're from, they sure did a fine job. LPDIR organizes all your disks by printing an information-loaded mini-label that you paste onto your diskette's jacket. This label, printed in extra-small type, shows at a glance the date the directory was made, each file's name and sector length, whether or not the file is protected and how many free sectors remain on the disk.

You can print your labels with the Atari 825 or Atari 1025 printers or on an EPSON MX-80. You have to enter a different line 10 depending on the type of printer you are using. Tn use LPDIR with an Atari 825 or Atari 1025 printer:

10 LPRINT "ESC/ESC, CTRL+T"

After "LPRINT," type a quotation mark ", press the [ESC] key twice, then press the [CTRL] and "T" keys simultaneously. End the statement with a final quotation mark.

To use LPDIR with an EPSON MX-80 printer:

10 LPRINT "ESC/ESC, 0, CTRL+O"

After "LPRINT," type a quotation mark ", press the [ESC] key twice, type in a zero (0), then press the [CTRL] and "O" keys simultaneously. End the statement with a final quotatinn mark.

Correct spacing is critical tn the printout. I've included numbers within brackets in this listing to show you how many blank spaces to enter. For example, "[8]" in the program means to enter 8 blank spaces.

LPDIR

```
10  LPRINT "[Refer to above
    instructins]"
20 LPRINT "  [8]   ***DISK
   DIRECTORY***"
30 LPRINT " "
40 DIM A$(40),MSG$(2),DATE$(18),
   DISK$(14):GRAPHICS 0
50 ? :? "DATE(DD-MM-YY)";:
   INPUT DATE$
60 LPRINT "  [6] ";DATE$
70 ? :? "DISK NUMBER ";:INPUT DISK$
80 LPRINT "  [6] DISK NUMBER [3]";
   DISK$
90 OPEN #1,6,0,"D:*.*"
100 LPRINT
110 LPRINT " [6]  FILENAME.EXT [2]
    LENGTH [2] ACCESS":LPRINT " [6]
    ------------[2]------[2]------"
120 INPUT #1;A$
130 MSG$=",W"
140 IF A$(1,1)="*" THEN MSG$="[2]"
```

```
150 IF LEN(A$)<17 THEN 180
160 LPRINT "  [6] ";A$(3,10);".";
    A$(11,14);"[3]";A$(15,17);
    " [5] R";MSG$
170 GOTO 120
180 LPRINT :LPRINT "   [11]    ";
    A$;""
190 A=VAL(A$):LPRINT "  [12]
    (";A*128;" BYTES)"
200 CLOSE #1
```

PEEKER
Look Into Your Atari's Memory

Many utility programs are designed to help you learn more about your Atari Computer. PEEKER is one such program. It lets you "look" at the binary contents of any memory location of your choice.

RUN this short tutorial program, follow the prompts, and gain an insight into the inner secrets of your Atari Home Computer.

PEEKER

```
1 ? "STARTING ADDRESS =":INPUT A
10 GRAPHICS 0:SETCOLOR 2,0,0:W=0
20 FOR W=A TO 65535
21 IF PEEK(84)=23 THEN GOSUB 1000
22 SETCOLOR 2,0,0
30 ? "ADDRESS ";W;"   ";PEEK(W)
40 NEXT W:END
1000 OPEN #1,4,0,"K:":GET #1,CH:
     CLOSE #1:IF CH=155 THEN
     GRAPHICS 0
1001 RETURN
```

THE AMAZING ELECTRONIC RING

Protecting Your Programs

Programs by
Lee Sherman and Kent Smith

Many, many years ago, I remember clipping a coupon from the back of a cereal box, scrounging up twenty-five cents, and mailing both to the cereal manufacturer for a plastic ring loaded with amazing features. Not only did this ring (which fit all sizes) glow in the dark and have a siren built into its top, but (and this is what really got me excited) it had a built-in message coder/decoder and secret message compartment!

Now, with the help of Lee Shenman and Kent Smith, I've discovered electronic replacements for the amazing ring of my youth. Ways to protect valuable, and not so valuable, programs from the prying eyes of others —specifically my thirteen-year-old son and his inquisitive gang of computer freaks.

You don't have to clip coupons or send in twenty-five cents (today it would cost lots more anyway) because we're going to share these foolproof (famous last words) and innovative program-protection methods with you. First, we'll write a one-line program that makes your screen flash a hunch of colors when it's RUN, then I'll show you how to protect it. Here's the program:

FLASHING COLORS

```
10 FOR X=1 TO 50:POKE 710,X:NEXT X:
   GOTO 10
```

1. Disable the BREAK key

The first thing you can do to protect *Flashing Colors* is to add another line of programming that will disable the [BREAK] key. This prevents the program from being broken into and listed while it's RUNning. Also, if you've designed a program that requires keyboard entry, disabling the [BREAK] key protects against "finger-slip," that dreaded mishap when a klutzy finger hits the [BREAK] key and brings your program to a screeching halt.

Delete "GOTO 10" from FLASHING COLORS and add this line to the program:

```
20 POKE 16,64:POKE 53774,64:GOTO 10
```

Now, RUN your new program and try to stop it by pressing the [BREAK] key. You can't get into it!

To be effective, this "BREAK Disable" routine must be inserted in your program after each Graphics Mode command.

2. The System Reset Foiler POKE

Disabling the [BREAK] key has its limirations. Some smart computer freak is going to figure out, real fast, that he or she can break into your program and list it by simply pressing the System Reset key. Let's foil this culprit. Add this line to your program:

5 POKE 580,1

Now, when the inquisitive smarty presses SYSTEM RESET, the *Flashing Colors* program is purged from the computer's memory—no program, no listing! This "System Reset Foiler POKE" is always the first line of your program.

3. Create a Run Only file

If the combination of the above two techniques isn't enough to dismay the curious and mischievous, here's a third method that's sure to foil even the most determined program hacker-cracker. Creating a *RUN Only* file prevents your program from being listed-out—after it has been loaded. Even you won't be able to list your program after you've followed the next diabolical procedures. So, just to play it safe, SAVE an unadulterated back-up copy of your program to diskette or cassette before continuing, then:

1. Add this line to your program: (It must be the *last* line of code in your program!)

```
32767 POKE PEEK(138)+256*PEEK(139)
      +2,0:SAVE"D:FILENAME":NEW
```

(If you are using an Atari 410 or Atari 1010 Program recorder, use SAVE "C: as the command to save your program.)

2. After you've typed in the program, replace the word "Filename" with your program's name. Here's how the *Flashing Color* program looks now:

```
5 POKE 580,1
10 FOR X=1 TO 50:POKE 710,X:NEXT X
20 POKE 16,64:POKE 53774,64:GOTO 10
32767 POKE PEEK(138)+256*PEEK(139)
      +2,0:SAVE"D:FLASHING.CLR":NEW
```

3. Now, in the immediate mode, type GOTO 32767 and your new, super-protected and uncrackable program will be SAVEd to diskette or cassette.
4. To see this program in action, type in: RUN"D:FLASHING.CLR" (or RUN"C: for cassette). Try to LOAD, ENTER or RUN the program. It won't work!

After you have *Flashing Color* up and RUNning there's no way to break into it—your computer will freeze up if you try.

We've *almost* created the perfect electronic secret ring. But, one element is still missing.

The Secret Password

Even though your program's listing is protecred from curious eyes, it still can be used by unauthorized computer freaks. By adding this short routine to the beginning of *Flashing Colors* (or any other program), only those who know the secret password will be able to gain access.

First, entet your program after line 140 as I've done (right) in lines 200 and 210. Then, create a RUN Only file and SAVE the entire humongous, el grando, protected mess to either disk or cassette. Now, no one will be able to list or use your valuable program—unless, of course, they know the secret password.

Assign Your Own Weird Password

The numbers following "K<>" in lines 50 through 90 are the ATASCII codes of the five letters or symbols you've selected as your password. Refer ro rhe "ATASCII Charactet Set" chart in your Atari Operators Manual to choose your secret password. I've used "ZAP!!" in this example. You can change the number of characters that make up your password by adding or deleting lines that contain "GET #1" statements. Don't forget to modify the PRINT statement in line 30 to tell the user how many characters to enter.

Change rhe Number of Tries

In line 110, the PASSWORD program allows the user three tries (IF TRYS>3) to guess the password before clearing memory with a NEW command later in the same line. You can easily change this to ">2" for two tries or any other number you'd like—don't give 'em to many tries!

Program Pilferers

These short routines don't "glow in the dark" or have "sirens." But rhey do fit all sizes (of programs, rhat is) and come equipped with electronic coding and a "secret compartment" to keep your programs safe from those marauding hordes of program pilferers.

Password

```
10 POKE 580,1
20 TRYS=0:OPEN #1,4,0,"K:"
30 PRINT "ENTER THE 5 CHARACTER
   PASSWORD."
40 PRINT :PRINT "PASSWORD_____";
50 GET #1,K:IF K<>90 THEN WRONG=1
60 GET #1,K:IF K<>65 THEN WRONG=1
70 GET #1,K:IF K<>80 THEN WRONG=1
80 GET #1,K:IF K<>33 THEN WRONG=1
90 GET #1,K:IF K<>33 THEN WRONG=1
100 IF WRONG=0 THEN GOTO 140
110 TRYS=TRYS+1:IF TRYS>3 THEN
    PRINT "SEE YOU LATTER...BYE!":
    NEW
120 IF WRONG=1 THEN PRINT "THAT'S
    NOT CORRECT...TRY AGAIN."
130 WRONG=0:GOTO 40
140 PRINT "YOU GOT IT!!"
150 .
160 . Your program follows:
170 .
200 FOR X=1 TO 50:POKE 710,X:NEXT X
210 POKE 16,64:POKE 53774,64:GOTO
    200
```

PLAYER/MISSILE GRAPHICS:

RIGHT SIDE UP!!

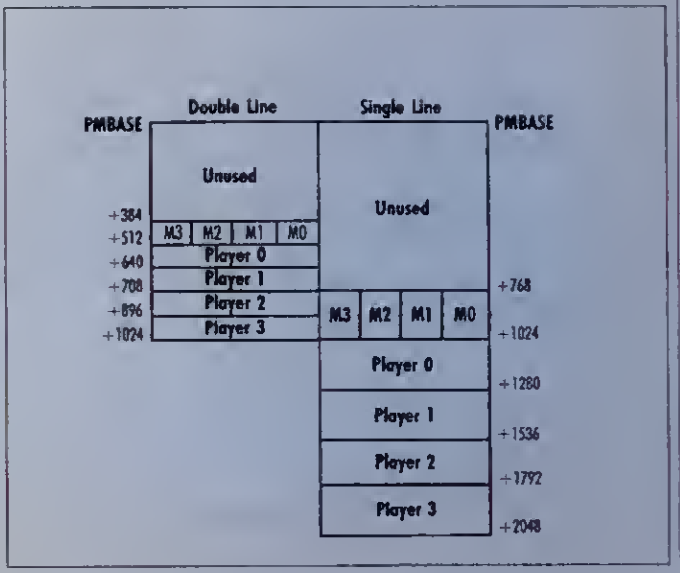
by David L. Heller

TAKE A LOOK AT FIGURE 1. See it? Have you seen it before? If you have, and you are familiar with player/missile graphics, then read on. It's the same player/missile memory allocation chart that has appeared in every computer publication since Atari introduced its computers. But it's wrong—it's upside down!!

While researching the player/missile chapter for the Dr. C. Wacko book, I read everything I could lay my hands on to gain an in-depth understanding of the subject. When I thought I knew player/missiles backwards and forwards, I sat down in front of my Atari Computer and started writing. That's when my problems began.

Problem 1

All those books said that "the pointer to the beginning of the player/missile area is labeled PMBASE." But PMBASE



LEARN TO PROGRAM IN ATARI® BASIC

ESI's TRICKY TUTORIAL™ Series shows you how to write GRAPHICS, SOUND EFFECTS, even complete GAMES! Each comes with a friendly, comprehensive manual and a tape or disk full of examples and utilities for you to use in your own programs!

TRICKY TUTORIALS™

Tricky Tutorial #1 DISPLAY LISTS - Learn how to make changes to the Atari's program that controls the format of the screen. For example, when you say "Graphics 8", the machine responds with a large graphics 8 area at the top of the screen and a small text area at the bottom. But what if you want some other graphics modes AT THE SAME TIME? Now, by modifying the Display List in the computer, you will be able to create screens similar to arcade games. The program does all of the difficult things (like counting scan lines). You will quickly be able to use the subroutines included in your own programs. 16K Tape or 32K disk. **$19.95**

TT#2 SCROLLING - Learn to move either text or graphics in any direction you wish. Many adventure games use this technique to allow a player to move over a complicated map that extends off the visible TV screen. We provide the basic methods and leave the rest up to your skill and imagination. Includes 18 examples to get you started, with several using a small machine-language subroutine for smooth scrolling. 16K tape or 32K disk. **$19.95**

TT#3 PAGE FLIPPING - Now you don't have to redraw the screen every time you change the picture or text. You will learn how to have the computer draw the next screen you want to see while you are still looking at the previous screen, then flip to it instantly. You won't see it being drawn, so a complicated picture can seem to just appear. Depending on your machine's memory size and the complexity of the picture, you can flip between many pages, thus allowing animation and other special effects. 16K tape or 32K disk. **$19.95**

TT#4 BASICS OF ANIMATION - This program shows you how to animate simple shapes (with some sound) using the PRINT and PLOT commands, and it also has a nice little PLAYER/MISSILE GRAPHICS game you can learn from. The P/M example is explained and will get you started on this complicated subject (more fully explained in TT#5). This would be an excellent way to start making your programs come alive on the screen with movement! Recommended for beginning users. 16K tape or 32K disk. **$19.95**

TT#5 PLAYER/MISSILE GRAPHICS - Learn to write your own games and other animated applications! The Tutorial begins with many small examples that complement the extensive manual, then gradually builds up to a complete game. Also included are two machine-language utilities that you can use to animate Players from BASIC. We also include tools for editing your background and your Player shapes. Requires 16K tape or 32K disk (the tape editor needs 32K, but is not required). **$29.95**

TT#6 SOUND AND MUSIC - Adding music to your programs can be easy! We show you in a unique way how to program notes, chords, and entire songs using a simple array method. You will even learn how to add graphics while the songs are playing! Also included is PLAYER PIANO. This program allows you to play the computer like a piano. It will even remember the notes you play and save them to tape or disk. 16K tape or32K disk. **$29.95**

TT#7 DOS UTILITIES - Included are seven utilities with explanations on how they are written. First is MENU, to help you select and run the program of your choice. Next comes an AUTORUN.SYS creator to make your disks run themselves. Also you will find a FORMATTING TOOL, an INSPECTOR to look at anything written on your disk, a lesson on writing and reading DISK FILES, and a program to check your drive's SPEED. Finally, if you have a printer, there is a small tool to create lists of all the titles on each disk. 32K disk only. **$29.95**

TT#8 CHARACTER GRAPHICS - By changing the shapes of the letters and numbers built into your keyboard, you can create almost any shape you can think of. Included is a powerful editor that will make changing the shapes and even animating them easy! The explanations in the manual start simply, then gradually build up to all that's needed to understand the COMPLETE GAME that is included as an example. 16K tape or32K disk. **$29.95**

TT#9 GTIA GRAPHICS - Most ATARI computers now enable BASIC access to the three new Graphics Modes -- 9,10, & 11 -- allowing nine colors or 16 shades of a single color. The many examples of graphics programming will show you three-dimensional shapes, digitized pictures, animation and more. The Tutorial even includes an editor to allow you to create and save your own nine-color pictures. 16K tape or32K disk. **$29.95**

TT#10 SOUND EFFECTS - Now you will be able to master the use of system timers and FORNEXT loops to make a wide variety of sound effects for all the programs you write. Included are examples of synchronizing your sound effects with graphics, two menus with over 30 special sounds, a sound editor, and full instructions for making it all work in YOUR programs. 16K tape or32K disk. **$29.95**

TT#11 MEMORY MAP TUTORIAL - Over 30 locations from our famous MASTER MEMORY MAP™ are fully explained and demonstrated by interactive examples. A few of the subjects include: how to control the cursor, text windows, user keys, tabs, joystick, paddles, inverse video, upside down lettering, break key disabling system timers...etc. 16K tape or32K disk. **$29.95**

TT#12 THE S.A.M. TUTORIAL - If you own the Software Automated Mouth™ by DON'T ASK™ Software, then we can teach you a few new tricks. Learn to make S.A.M. sing, change the sound of his voice, and even combine graphics with the voice! Of course we will show you the basics like phonemes and stress, with editors for each subject. 32K disk only. Requires S.A.M. **$29.95**

TT#13 BASIC PROGRAMMING TOOLS - The Ultimate Renumbering Program allows even partial blocks of code to be renumbered. The same goes for DELETE. Next we offer TRACE to go through your program step by step while "de-bugging" it. If you have ever tried to read someone's code that was written with many statements per line, we offer EXPAND to make them all readable again. For disk users we have QUICKREF. It tells you where all of your BASIC program's variables and numeric constants are used. It also tells you how many bytes of memory your program uses. Printer owners get a special bonus: LISTER I for Epson® printers and LISTER II for all others. These allow you to print out program listings exactly as they apear on the screen, with all inverse video and "special" characters shown! 16K tape or 32K disk. **$29.95**

TT#14 ADVANCED PROGRAMMING TOOLS - Full explanations of new tricks for old techniques using 7 of our popular machine language routines, designed to be used in BASIC, and 2 of our advanced programming utilities. Learn new memory efficient and more effective ways of manipulating player missiles, time delays, page flipping, high resolution graphics, and special character fonts. Do "timed" keyboard inputs. Save graphics screens to disk or tape with our I/O routine. A full package of tricks the pros use that takes the work out of sophisticated programming. 32K disk only. **$39.95**

TT#15 FANCY FONTS - The font styles of past ages, as well as future ones, now become available on your computer. Learn to create custom lettering styles, special displays and more. Includes the powerful font editor from our Character Graphics Tutorial, "YE OLDE TYPEFONT" to generate fancy displays like you have never seen before. Many sample styles of lettering that are ready for you to use, and for those with a printer, a few utilities to make your custom fonts come alive in your work! 16K tape or 32K disk. **$29.95**

OUR FAMOUS REFERENCE GUIDE

MASTER MEMORY MAP - A remarkable, 32-page guide to the built-in capabilities of your ATARI computer. It starts out by explaining how to PEEK and POKE values into memory, so that even new computer owners can use many of these "Tricks." Then, you are given hundreds of the most useful memory locations, along with hints and sample programs. The Appendix discusses problems with BASIC and the Operating System, the new GTIA Graphics Modes 9, 10, and 11, and much more! **$6.95**

SPECIAL PACKAGES!

THE FIRST SIX TUTORIALS
in a 3-ring binder for only **$119.95**
a$20.00 savings!

GRAPHIC AND SOUND TUTORIALS
TT's #8, #9, #10, #11, and #15
in a 3-ring binder for only **$119.95**
a$30.00 savings!

See your local dealer; if not available, order direct-

For C.O.D/MASTERCARD/VISA orders only:
(800) 692-9520 or (408) 476-4901

Please add $3.00 for postage and handling.
If ordering only a Memory Map add $1.50.
Outside U.S. and Canada add 15%.
California residents add 6.5% sales tax.

EDUCATIONAL SOFTWARE INC. 4565 Cherryvale Ave. Soquel, CA 95073

(short for Player-Missile BASE) was ar the TOP of the chart! Right off the bat, something smelled fishy. A "base" should be at the bottom—shouldn't ir?

Problem 2

Next, I read that "PMBASE must start on a 1K boundary for *Single Line* resolution, and a 2K boundary for *Double Line* resolution." What are they talking about, I wondered: "A '2K boundary' from what?"

Problem 3

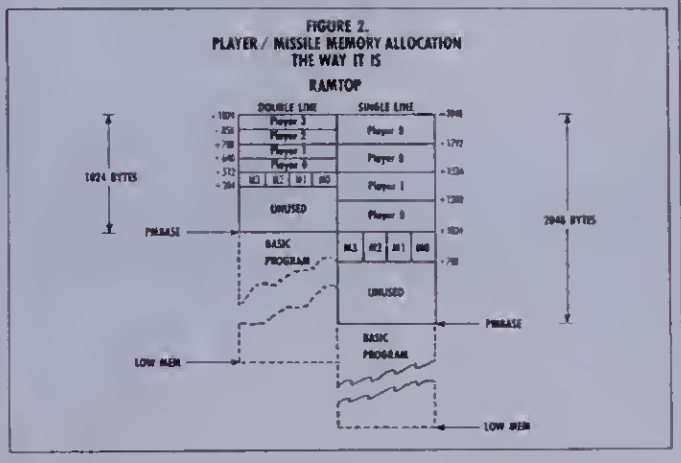
The third thing that confused me were articles that said: "You can use the portion of the player/missile area marked 'unused' for anything you like." But, I asked myself, where does this "unused" memory area reside?

Problem 4

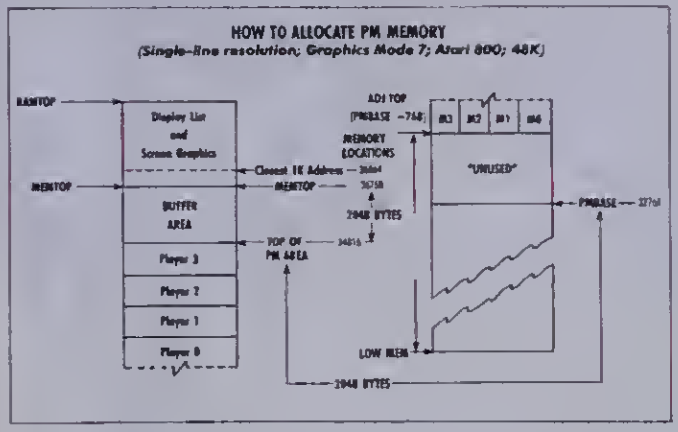
The final bit of exasperation came as I tried to move RAM below the top of RAM by assigning a value returned by PEEK(106), then multiplying this value by 256 to arrive at PMBASE. Confused? I sure was. And to make matters worse, I discovered that this nifty method didn't work in all *Graphics Modes*. There was no set formula, no method to any of this madness!

The Ultimate Solution

After much work, and with more than a little help from my friends, I finally discovered the ultimare solution to *all* these problems. And here it is!



Now it's logical! RAMTOP is at the *top* of the chart and PMBASE is at the *bottom* where it belongs! This is the way it really is!



Look carefully at Example 1. It shows the actual memory locarions present when memory is ser aside for *Single Resolution* player/missiles in Graphics Mode 7 using an Atari 800 with 48K.

Right below RAMTOP (the top of your computer's Random Access Memory) is the *Display List* and *Graphics Screen Memory* allocation. Below this is a "Buffer Area," followed by the player/missile area, then the "unused" area and finally, your BASIC program—what a relief!

Here's a short program rhat shows how to put this new knowledge to use:

P.M. Not A.M.

```
10 GRAPHICS 7+. Any Graphics Mode
   works. Check it out!
25 .
20 MEMTOP=PEEK(741)+256*PEEK(742)-1
30 PMBASE=INT((MEMTOP-2048)/2048)
   *2048
40 ADJTOP=PMBASE+768
50 POKE 742,INT(ADJTOP/256):
   POKE 741,ADJTOP-256*PEEK(742)
55 .
60 POKE 54279,PMBASE/256
70 POKE 53277,2
80 POKE 559,34+8+16
90 P0=PMBASE+1024
100 FOR A=P0 TO P0+256:POKE A,0:
    NEXT A
110 FOR A=P0+100 TO P0+107:
    READ B:POKE A,B:NEXT A
120 POKE 53256,2
130 POKE 704,PEEK(20)
140 POKE 53248,PEEK(20):GOTO 130
150 DATA 60,126,129,153,255,36,66,
    129
```

All the right stuff is in lines 20 rhrough 50.

Line 20 finds out where the Free Memory High Address is. This memory location, called MEMTOP, is right below the Graphics Screen area you've selected. MEMTOP is location 36768 in Example 1.

Line 30 allocates room for player/missiles. When you set up *Single Line* player/missile graphics, the top of the player/missile area will be 2K below the *closest memory location to* MEMTOP *that is evenly divisible by* 1024. Here's how this works our in Example 1:

> MEMTOP is 36768. If you divide 36768 by 1024 you'll get 35.90625. To find the closest location to MEMTOP that is *evenly divisible* by 1024, multiply 36 hy 1024—it's 36864! Get it?

Next, you find the top of the player/missile area (location 34816) by subtracting 2048 (2K) from 36864.

You finally subtract 2K (2048 bytes) from 34816 to arrive at PMBASE. In Example 1, PMBASE is at location 32768.

Line 30 does all the work. I just want to show you the logic of this method. The code in line 30 does all this work for you!

To make room for *double line* player/missiles, replace all the "2048s" with "1024" in this line.

Lines 40 and 50 use that "unused" area. Two very important things are accomplished by making *ADJTOP* equal to PMBASE+768:

1. You get to use rhe "unused" area and pick up an additional 768 bytes of usable memory! The "unused" area can store a redefined character set, machine language routines and orher data!

2. You protect the player/missile area, because you've rold your computer that it can't go past ADJTOP—the new Free Memory High Address you've set.

*Note: To set aside room for *double line* player/missiles, change the statemenr in line 40 to PMBASE+384.

Now you can breathe easier—I certainly am. You know the right way to allocate player/missile memory! Put this algorithm in your programs and display your players and missiles in *any* Graphics Mode while protecting the player/missile area and increasing the usable programming area.

Footnote: Many thanks to Bill Wilkinson of Optimized Systems Software (the people who designed and produced ATARI BASIC) for helping me solve this mystery.

COMPUTER COMFORT

Some Questions on Atari Graphics

A Pixel of a Different Color

by Jane Sokolow

WHETHER YOU DISCOVERED the fact after you brought your Atari Compurer home or purchased it for this reason, you must know by now that you have a computer with some powerful graphics capabilities.

OK, you understand that there's all that graphics power at your fingertips, but never mind player/missile graphics, shadow registers, vertical blank inrerrupts, ANTIC and GTIA—ler's get down ro rhe simple stuff. We will be answering some rather obvious "but afraid ro ask" type of questions about graphics, using ATARI BASIC.

Q: How come, when I type GRAPHICS 2+16, there's a flash and then the GRAPHICS 0 screen comes up with the READY prompt at the top? Is my machine broken?

A: No. In all cases but Graphics Mode 0 there will be a text window at the bottom where the READY prompt will be. You have eliminated the window correctly hy adding 16 to the Graphics Mode number. In direct mode, BASIC always displays a "READY" prompt when it has executed your command. The READY prompt is always at the top of a Graphics Mode 0 screen. If you eliminate the text window, BASIC will briefly execute your command of GRAPHICS 2+16 and then default to Graphics Mode 0 in order to display "READY" at its usual spot at the top of the screen. The instantaneous default causes the "flash" you see.

Q: How can I center my text in the text window?

A: Use formars such as:

```
10 PRINT (Gives a blank line)
20 PRINT "          HI"
```

A POSITION statement won't work. The format ahove will display a blank line followed by a HI in the middle of the rext window.

Q: When I try to draw in a fourth color using SETCOLOR, one of the other drawings changes color, too. Why does this happen?

A: Aha! There are ways to get more than three foreground colors on the screen, hy reaching deep within the ANTIC chip—but they are fancy *and* tricky. We are staying simple, remember? You used SETCOLOR to change a color you originally assigned to a register. When you tried for a fourth color, ynu changed the first drawing's color by changing the color in that register. Using conventional ATARI BASIC, you can't get more rhan three foreground colors in the Graphics Modes 3, 5 or 7.

This answer begs for a brief discussion of Graphics Modes 4, 6 and 8. *Resolution* means how many pixels (units) there are. Graphics Modes 3 and 4 are the same resolution (40 across and 24 down). Graphics Modes 5 and 6 are the same as well (80 across and 48 down—twice as many pixels as 3 and 4). Graphics Modes 4 and 6 only have one foreground color. The computer uses memory to store information about each pixel—and the higher the resolution, the more "expensive the overhead." Adding color adds one more item of overbead. Graphics Modes 3, 5 and 7 have to use two bits per pixel, whereas modes 2, 4 and 8 use only one. Multiply these numbers hy the number of pixels in a line, divide hy 8 (8 bits in a hyte) and you will get the number of bytes of memory each line will take. Graphics Mode 5 will take 2×80÷8=20 bytes of memory, whereas Mode 6 takes 1×80÷8=10 hytes, for example. Graphics Modes 4 and 6 were developed for times when memory is a consideration.

Q: When I plot a vertical line in any Graphics Mode it is a different color from a horizontal line, and in Graphics Mode 8 even columns are different from odd columns. Shall I take my machine in for repair?

A: No, but when there's a cure for this problem, I'll be the first in line at the repair window! The problem is caused by the way the TV picture tube projects the color signal on the screen. *Horizontal lines* are often more intense and clearly delineated than *vertical lines*. Graphics Mode 8 has such fine resolution thar each pixel can only express the signal from half of the *color clock*—the unit of time it takes the television receiver to light up rwo pixels. Thus odd columns are a little different from even columns, which creates a phenomenon known as "artifacting." This phenomenon can be used to create stunning moire patterns and other interesting effects in Graphics Mode 8. COMPUTE!'s *First Book of Atari Graphics* has an entire chapter dedicated to the possibilities of color combinations in Graphics Mode 8.

Listed below are some excellent books on the subject of Atari graphics that will help you undersrand and learn more about computer graphics and animation.

Jane Sokolow is a Product Specialist in the Customer Relations Department at Atari, Inc.

Bibliography

Inside Atari Basic by Bill Carris, $12, 183 pages, Reston Publishing Co. (800) 336-0338.

Your Atari Computer by Lon Poole, et al, $16, 458 pages, Osborne/McGraw-Hill (415) 548-2805.

Atari Games & Recreations by Kohl, Kahn & Lindsay, $15, Reston Publishing Co. (800) 336-0338.

Atari Sound and Graphics by Moore, Lower & Albrecht, $10, John Wiley & Sons (212) 850-6000.

COMPUTE!'s First Book of Atari Graphics, $13, 248 pages, COMPUTE! Books (800) 334-0868.

The Creative Atari, $16, 243 pages, Creative Computing Magazine (800) 631-8112 or in NJ (201) 540-0445.

Computer Animation Primer by David Fox and Mitchell Waite, $19, Byte Books (609) 426-5454.


You
Don't Just Play
DONKEY KONG*
You Feel
It!
Now, the Thrill of the Arcade Game at Home
What's barrels of fun for ATARI Home Computer owners? It's DONKEY KONG* by Nintendo*. The familiar music, appealing cartoons, and four screens of play that have made the arcade game so popular are now in a cartridge for ATARI Home Computers.
Kongratulations ATARI!
ATARI®
A Warner Communications Company
*Trademarks and © Nintendo 1981, 1983
© 1983 Atari, Inc. All rights reserved.
Made in U.S.A.
Atari, Inc. Home Computer Division
P.O. Box 61657
Sunnyvale, CA 94086
Games People Feel

Educational Programming

Your Home Computer Is a Tool for Learning

by Earl Rice

AFTER YOU'VE HAD your computer awhile, you may notice that playing games gets less interesting. Maybe you've got an uneasy feeling that your computer is getting a little tired of the same old stuff. That's the computer version of the "Call of the Wild"; the urge to see what this machine can *really* do; the curiosity that leads you to enter a programming example and see if it runs, or to try to Find the Bug.

When first learning about the world of computing, most people concentrate on the computer itself. Learning about bits, bytes, and BASIC—that's natural enough; after all, the computer nomenclature is something you have to master. And it takes a lot of attention. Using a computer can be easy, but programming takes a little more time. It takes a willingness to learn, to try, and to try again. Users' Groups are a way to make that learning easier. They are full of people who program, and most are eager to help you learn.

I found that learning to write programs was only the first step. Once I learned to program fairly well, I realized I hadn't really thought about why I had learned in the first place. I hadn't any idea of what to do with my newly developed skill.

I learned to write the same way. Back in grade school, my teachers taught me to put words on paper. I remember wondering why anybody would want to do that. It seemed to be a lot of trouble. I enjoyed learning a new skill, but I hadn't yet learned what to do with it.

It seems to me that most of us in the users' group community are in a similar situation: We've learned to make our machines do things, but we're not quite sure what we ought to be using them for. Certainly, many groups are using their computers to increase and improve communication. Many are involved in the development of educational and artistic programs. But most of us are still playing with new skills and applying them to our points of view—like passing notes in class or scrawling graffiti on clean walls. Skill without purpose.

That's why this issue's education theme is so important. We always apply new skills in support of our personal values. Seeing if we can, somehow, design and program our own computer game is one expression of those values. Improving communication is another. Using what we know to serve a worthy cause is yet another. Education, at its best, helps us to acquire positive values and enables us to improve our lives.

There are many educators in the users' group community, but there are few educators' users' groups. As a result, there are many young programmers with the skill to produce games that teach. But few are being taught how to apply good, sound educational theory to a learning environment. And so the primary value being transmitted by most amateur and professional games is *conflict*. Educators who program can make a tremendous impact on our future values.

Elsewhere in ATARI CONNECTION, you will find information about an educators' network—*the Atari Teachers' Network*. Check it out. Even if you're not an educator, but you've begun to master the art of programming and want to know what to do with your skill, look into the field of education. The combination of computer games and education is one of the most powerful forces to become available to us in a long time. More powerful than the printing press, perhaps, because while a book presents you with ideas, a computer lets you interact with ideas. No other resource can do that.

Except a good teacher. ■

READER SERVICES

Change of Address? Please give us 6 weeks notice for a change of address. Attach your mailing label and print your new address below.

New Subscription? Check the box below and enter your name and address. You can receive the next four issues of ATARI CONNECTION for just $10.

A Question? We will help you with any subscription problem you have. Remember to enclose your mailing label in order that we can give you prompt service.

- ☐ **New Subscription**
- ☐ **Change of Address**
- ☐ **Payment Enclosed**
- ☐ **Bill Me Later**

NAME

ADDRESS

CITY STATE ZIP

Subscription Dept.
ATARI CONNECTION
P.O. Box 50047
San Jose, CA 95150

"If Red Were Blue Then I Could Not Love You."

IF/THEN: A Conditional Statement

by Herb Kohl

RED IS NOT and can never be blue —but think about what the sentence says: "If red were blue then I could not love you." Does it mean that if red were *not* blue then I *could* love you?

"IF/THEN" statements are wonderfully simple and amazingly complicated. Even very simple programs usually have places where more than one thing might happen; that is, where different conditions lead to different results.

For example, a game program might ask your name and then give you the choice of using a joystick or the keyboard to control the game. If you choose to use the keyboard, the computer will display the keys used. If you choose the joystick, the computer will tell you which port to plug your joystick into. At that point the program is using a conditional mode.

Most likely, if the game is interesting there will be many other choices built into the program as the play progresses. At the end of the game the program will offer another conditional when you are asked if you want to play again or not.

An argument I heard the other day made me think about the use of conditionals in computer languages. Two young boys were arguing about which was the better language, *BASIC* or *PILOT*. It seems one had been using BASIC at his school and the other PILOT at his school. The argument seemed more about whose school was better than about the languages. Yet their dispute raised a question that is often heard among adults. Is there a "best" computer language? Can they be ranked like golfers or tennis players?

I don't have the answers to those questions but believe that a study of how conditional statements are used may reveal a more sensible attitude toward the differences in computer language.

Computer languages handle conditionals in different ways because they were created for different purposes. For example, BASIC (Beginner's All-purpose Symbolic Instruction Code) was created in the mid-1960s at Dartmouth College by John Kemeny and Thomas Kurtz to give students who were not math or science majors the opportunity to program computers. It was meant to be a simple nontechnical language and sacrificed some of the mathematical complexity of *FORTRAN* and *Pascal* in order to come closer to the logical structure of English. It is a language of *logic*, not of mathematics. *PILOT* (Programmed Inquiry Learning Or Teaching) was developed by John Starkweather for a different purpose. Starkweather created a language which teachers could use to design their own compurer-aided instructional material. The ability to easily handle "conversation" and text is a characteristic of PILOT and central to its structure.

It is important to understand that though PILOT and BASIC are different, it is irrelevant to ask which is the better language. They were designed to do different things and the critical question is how well they perform the tasks they were designed to do.

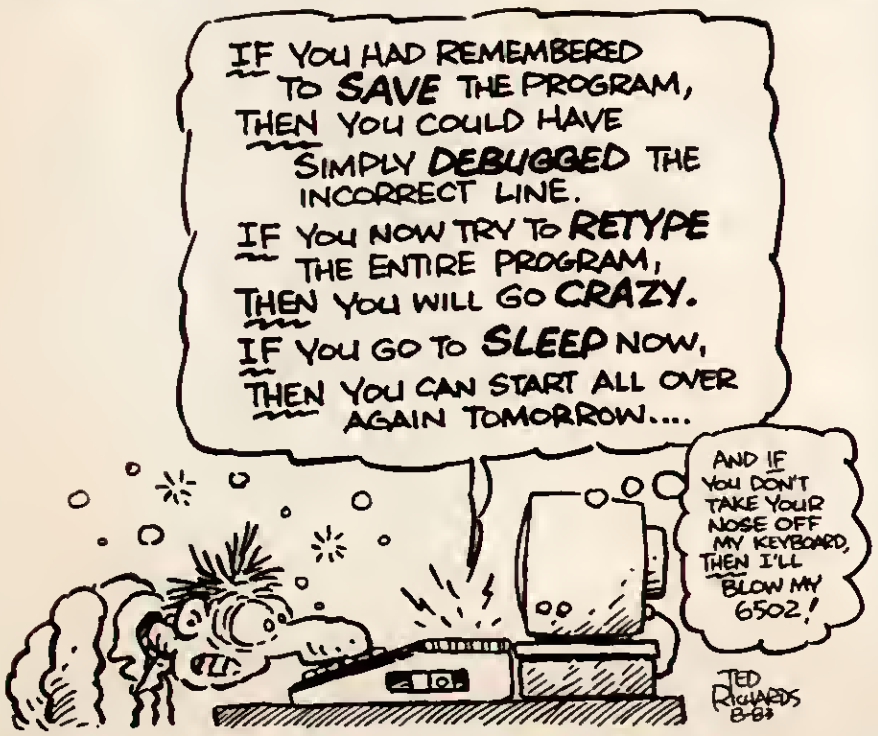
We can actually witness the struc-

BASIC I

```
5 ? CHR$(125)
10 PRINT "Is red blue?"
20 DIM A$(3)
30 INPUT A$:? CHR$(125)
40 IF A$="YES" THEN PRINT
   "I love you":END
50 PRINT "Sorry but I
   love another"
```

PILOT I

```
10 T: Is red blue?
15 T:
20 A:$ANSWER
30 M:YES
35 T:
40 JM: *LOVE
50 *SORRY
60 T:Sorry, but I love
   another.
70 E:
80 *LOVE
90 T: I love you.
100 E:
```



tural differences between PILOT and BASIC by programming a simple introduction to a game in both languages. When you run these programs they will do exactly the same thing on the computer screen. Their differences will appear in the second example and require a bit more reading (an old-fashioned skill worth developing).

You'll notice the BASIC program is *half* as long as the PILOT program. It uses what could be called a *direct conditional*, the "IF/THEN" statement. The PILOT program uses the two modules *LOVE and *SORRY and a match(M) statement to do the same thing. Superficially, it would seem that since BASIC uses half as many lines as PILOT, it handles conditionals better. But let's suppose that our programmer was a fan of the poet Andrew Marvell and wanted to introduce some poetic elegance into the program. That could be done quite easily simply by adding the poem to the *LOVE module as in this program:

BASIC II

```
5 ? CHR$(125)
10 PRINT "Is red blue?"
20 DIM A$(3)
30 INPUT A$:? CHR$(125)
40 IF A$="YES" THEN GOTO 60
50 PRINT "Sorry but I love
   another...":END
60 PRINT "My love is of
   birth as rare"
70 PRINT "As 'tis for object
   strange and high:"
80 PRINT "It was begotten by
   despair"
90 PRINT "Upon Impossibility."
```

PILOT II

```
10 T: Is red blue?
15 T:
20 A:$ANSWER
30 M:YES
35 T:
40 JM: *LOVE
50 *SORRY
60 T:Sorry, but I love
   another.
70 E:
80 *LOVE
90 T:My love is of a birth
   as rare
100 T:As 'tis for object
    strange and high:
110 T:I was begotten by
    despair
120 T:Upon impossibility.
130 E:
```

BASIC now finds this increased complexity (a poem) harder to deal with. Its line 40 simply cannot accommodate the verse, especially since each line in it has to have its own separate line on the screen when RUN. The program has to jump using a GOTO statement which smells suspiciously like a *module*. Notice that the difference in length between the PILOT and BASIC programs has decreased considerably and the amount of typing you have ro do hardly varies.

The simplicity of BASIC compared to PILOT was deceptive in the first set of programs. As you explore a language, you discover strengths and weaknesses that are not apparent in the simple examples above. Making comparative judgments when first learning computer languages doesn't lead to a better understanding of them. It is smart to take a respectful attitude toward different computer languages and, instead, discover the ways in which they are most useful to you.

A good exercise that illustrates this idea is to try several different programming problems and applications in the languages you are learning. Outlined below are several problems in "comparative computer linguistics" that will reveal some rather startling differences between PILOT and BASIC. Try to write your own programs—first in PILOT, then in BASIC—that perform the following tasks:

1. Create a *form letter* that allows you to fill in the blanks. Here is an example you can use:

Dear _____ ,

Thank you very much for the _____ . I always wanted _____ .

I hope ro see you _____ _____ .

Your _____ .

2. Write in a program that allows you to perform long division.

3. Create a graphic program that draws this pattern on the screen:

For those who feel they have created interesting solutions to the above programming problems, you're invited to send your programs, along with a short lerter, to:

Herb Kohl
c/o ATARI CONNECTION

Herb Kohl is an author of numerous books on computers and education and is the Director of the Coastal Ridge Research and Education Center.

Build Confidence Byte By Byte

High-Tech Info for Women

by Elizabeth Metzger

COMPUTER CONFIDENCE: A WOMAN'S GUIDE
by Dorothy Heller and June Bower
illustrations by John Johnson
Acropolis Books Ltd., 260 pages
$16.95 hardcover, $9.95 paperback

WHEN TWO RECOVERED "computerphobics" sit down to write a book together, chances are they'll come up with something for computerphobics. But *Computer Confidence: A Woman's Guide*, by Dorothy Heller and June Bower, is a different kind of remedy for those who suffer from a fear of computers. Written specifically for women, it is, according to the authors, a "personalized travel guide to the computer revolution" which has had "a special impact on women, both as an opportunity and a challenge."

Indeed, the *opportunity* and *challenge* of computers lie at the heart of this book: the opportunity for women to leave traditionally female, low-paying, dead-end jobs in nontechnical fields; and the challenges of entering the traditionally male computer field with its high technology and competitive salaries.

With listings of high-tech "buzz words" and guides to shopping for software and hardware, Heller and Bower set out to help readers learn the basics of computing and thus overcome their fear of technology. Interspersed with the more practical "how to" sections are doses of inspiration, including a short history of women in computing and several stories of women who have carved successful niches for themselves in computers. Artist John Johnson's cartoons add a humorous lightness that makes readers stop and laugh at their own preconceptions about computers.

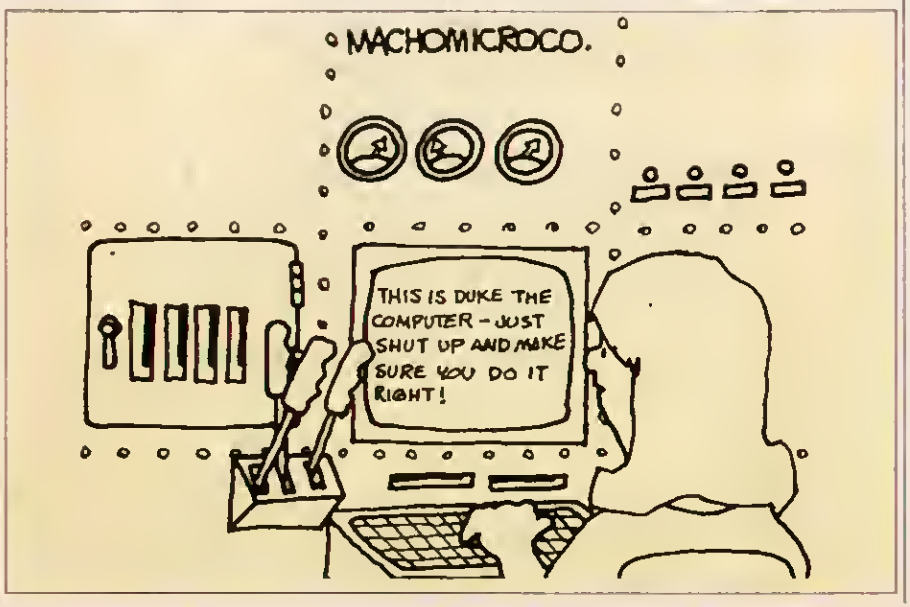
The authors encourage readers to see themselves as technologically competent people. For starters, they ask you to list all the tools and technology you already use. "Don't forget your hobbies that involve machines—your camera, dark-room equipment, your sewing machine." They assure you that you don't have to be a high-level mathematician or technician to learn how to use computers.

The book is rich with human interest stories, both past and present. Ada, the 19th-century Countess of Lovelace, nicknamed "the Princess of Parallelograms," invents the concept of "loops," "subroutines," and the "conditional jump"—all basic components of programming. Captain Grace Murray Hopper, a World War II-era mathematician and naval midshipman, is the first to use the term "debug" when she discovers that a moth inside one of the computers she's using is the source of a malfunction. Emily Mims, a present-day Texas chemistry teacher, sells her first romance novel to Dell Publishers after doing major revisions in one week using an Atari 800 Home Computer.

The best parts of the book are those that deal with the nitty-gritty of entering the computer age: chapters on how to buy a computer, how to know what software you really need, how to find a computer dealer who won't try to sell you a bookkeeping system when all you want is a word processing program.

Although *Computer Confidence: A Woman's Guide* has been written for women, its message is of value to men as well. And herein lies the dilemma of this book. On the one hand, the authors will reach an audience that might not ordinarily buy a book about computers—women whose kids already know BASIC and whose husbands use VisiCalc at work, but who've been left behind by the computer revolution. On the other hand, the very title of the book seems to ignore a potential audience of men who may not know a byte from a disk drive. Given the alternatives, however, Heller and Bower have written an important book. One only hopes that as more and more women start to work with computers, there will be less need for a guide to the field that addresses itself exclusively to women.

For further reading on women and computers, see "Careers in Computing" by Teddi Converse in the Spring 1983 issue of the ATARI CONNECTION. ■



ILLUSTRATED BY JOHN JOHNSON

Elizabeth Metzger is a writer in the Creative Services Department of the Atari Products Company, and a Contributing Editor to ATARI CONNECTION.

Errata, Errata

Our series, *Cartoon Computer Animation*, by David Fox and Mitchell Waite, will continue in the Christmas/Winrer, 1984 issue of ATARI CONNECTION.

Meanwhile, we'd like to correct a few typos and mistakes that appeared in last issue's article:

1. On page 57, you were told to change a program line. You really didn't have to "change" a line; all you had to do was type the line POKE 756,200 to allow you to peek into the Atari Computer Operating System—you don't need a "program" to do this.

2. On the same page 57, under Case of the Reserved Character Set," you were told there were three "building blocks" within the *memory location*. This is not true. Please forget about the "third block."

3. On page 58, we told you to see "Fig. 7" to understand how a *character frame* is arranged in a 2 x 3 array. We meant Fig. 2 at the top of rhe page.

4. On page 58, in rhe last paragraph of the middle column, line 0020 should read line 8020. In the last paragraph of the third column, line 230 should read 330.

5. In the "Special Listing Instructions, on page 61, the example in Instruction 1, should have the letters C, E and F underlined as follows: S$="ABCDEFGHI"

Miscellaneous Errata:

Our article on "Networking with AtariWriter" had a minor snafu: In the last column on page 37, under instructions "4" and "5," the instructions for LOAD DEVICE:FILENAME were wrong. You are told to type R—R1: in both instructions. The correct format is simply R1:

A typo appeared in Jane Sokolow's *Computer Comfort*, which provided corrections for program listings published in early editions of some Atari instruction manuals. The program line corrections all belong to *The Basic Reference Manual*; however, there was a missing correction for *The ATARI 800 Owner's Guide* in *Program 1* that appeared under the section titled "Enjoying Your ATARI 800 Computer":

```
150 PRINT "Your Guest";:INPUT G
```

In *Computer Classroom*, the program listing for "Program 2" is missing 4 lines. Add lines 640, 650, 660 and 670 from "Program 1" to "Program 2" beginning at line 630.

Logo vs. BASIC

(continued from page 29)

the powers of the machine.

Logo

Logo is ideal for the novice computer user who just wants to get his or her feet wet—it's an excellent introduction to the world of computers and programming. Logo was especially constructured to help individuals develop their problem-solving skills and to encourage logical thinking. It is ideal in this environment, even for advanced programmers.

Logo also provides a very good environment for studying various concepts in the domain of computer science, but these concepts (Artificial Intelligence, data structures, list processing) are all quite complex and not for the beginner.

The Logo language was designed so a programmer would not have to encounter the internal architecture or operating system of the computer. With Logo, people can be creative without having to worry about technical details. But Logo does share BASIC's capability to store and retrieve data directly from the computer's memory (RAM). In BASIC, the commands are PEEK and POKE; in Logo you use .EXAMINE and .DEPOSIT. Also, both have the ability to call upon a *machine language* routine. In Logo this command is a simple CALL; in BASIC, the USR command.

The machine language features of both these languages are indispensable for programs that need fast processing speeds and need to "talk" directly to the computer's hardware. But with Logo's limited "access commands" you really can't "tinker" with your computer. In this regard, Logo is probably not the kind of language that would satisfy the computer enthusiast.

BASIC

BASIC is a good language to start with when first using computers, but many people develop poor habits when BASIC is their first programming language. As programming languages go, BASIC doesn't encourage systematic approaches to problem solving.

As far as learning about computers, BASIC provides an adequate environment for developing an understanding of computer sciece, hut cannot be recommanded as a language that can teach human interaction with computers—one of the more challenging and sophisticated aspects of computer science.

But, for the computer enthusiasts —you hacker types ready to jump in and get your hands dirty — BASIC provides a plethora of programming features that allow easy access to the internal operating system of computers. In this respect, BASIC is clearly the choice for this type of user environment.

Which Language is Best?

You could say BASIC is best if you want to write programs for simple database managers, arcade-style computer games and system utilities—it's a true *workhorse language*, in other words.

Then again, you could say Logo is best if you simply wish to learn a language to know more about computers or if you want to teach your child about computers. Logo is also the perfect language for writing short, easy-to-program learning games that are fun to play, because of the turtle graphics and list processing capabilities. After you start learning Logo you will find that it is an excellent language for challenging your mind. It is like a very complex and exciting video game that never ends—and doesn't use up your quarters.

In answer to the question *Which language is best?*, you should ask yourself, "What do I want to do with my computer?" The chances are you'll answer "Logo *and* BASIC . . ."

by Dave Menconi and Ted Richards, with portions excerpted from Language Comparisons *by Dr. Wayne Harvey of Atari Special Projects.*

Have You Read The Reviews On Our New AtariWriter™?

Atari's New Word Processor.

Try it, be your own critic. Use new AtariWriter for all your family correspondence, school papers, committee reports, mailings.

Our advanced ROM-based cartridge technology means you can use it on any ATARI Home Computer (even 16K). It also lets you choose between cassette and disk storage systems.

Easy to learn and to use. "Yet," says ANALOG Magazine, "as your uses for word processing grow, (you'll employ) its more advanced features."

Menu driven, with helpful prompts. One special feature: you can correct as you write, instantly, without switching back and forth between Create and Edit modes.

You can move entire paragraphs. Search and replace. Even restore text you've just deleted, with our memory buffer.

You can change format instantly, and preview on-screen at full printer width. Print extra copies for your records.

As ANTIC Magazine concluded, new AtariWriter is "clearly superior on price and performance to Text Wizard and the ATARI Word Processor." At Atari dealers.

ATARI® HOME COMPUTERS

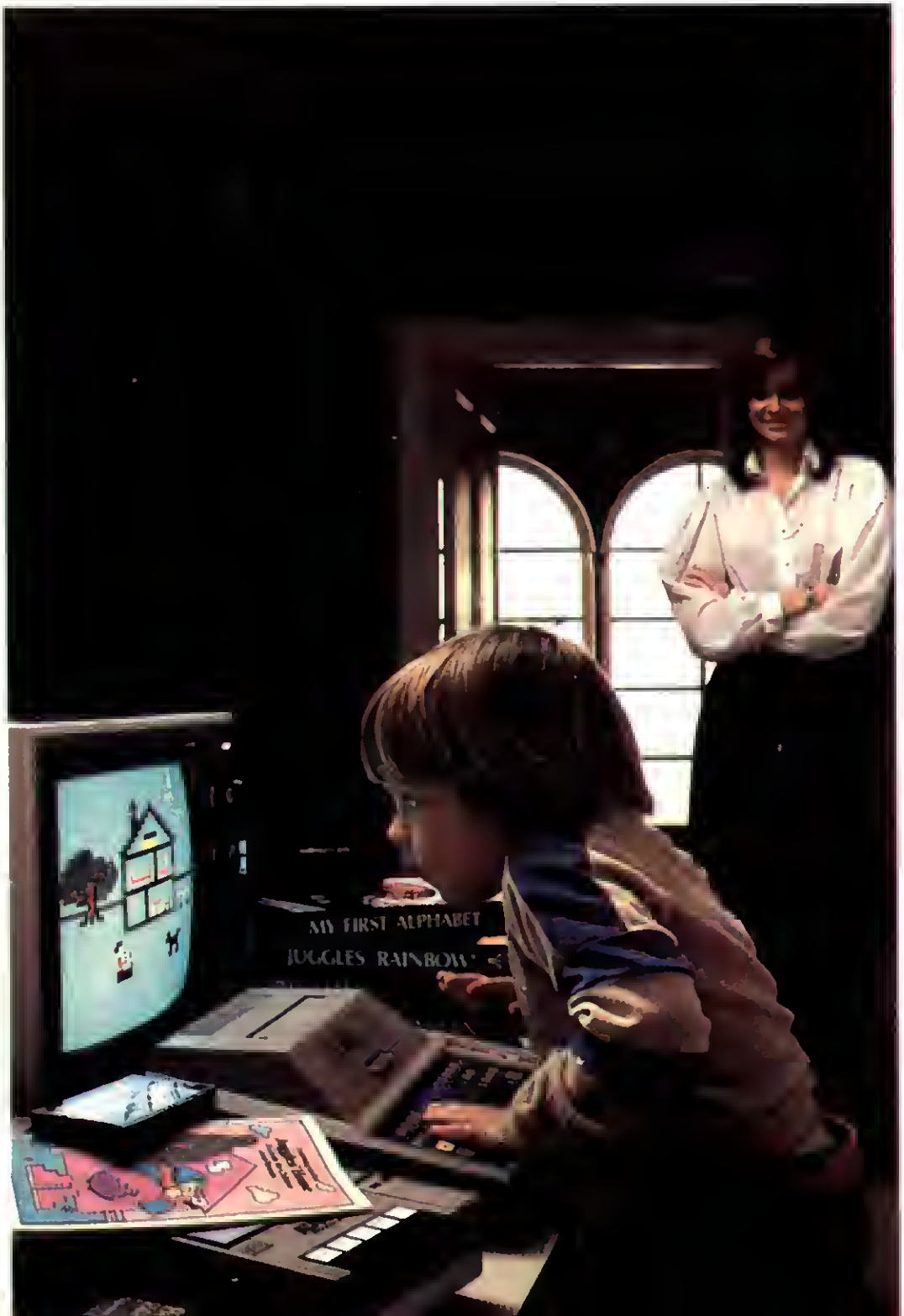
A Warner Communications Company.

ATARI CONNECTION
60 E. Plumeria Drive
P.O. Box 50047
San Jose, CA 95150

Address Correction Requested

Bulk Rate
U.S. Postage
PAID
Permit No. 219
Pontiac, IL

Gavin thinks he's playing. Gavin's mother knows he's learning.



With colorful graphics, lively melodies and captivating computer animation, Juggles' House* teaches *upper* and *lower*, *inside* and *outside*. Understanding these concepts helps your child learn to recognize shapes and letters – to begin reading.

Games with windows, toy shelves and a clown's house help children understand basic writing and counting concepts. And naturally, Juggles' House comfortably establishes computer awareness.

Give your child Juggles' House for its early learning fundamentals. Or, give your child Juggles' House for the fun of it.

ATARI®

C017668-11