SPECIAL REPORT: COMPUTER CAREERS FOR WOMEN

NEW Atari XL Family of Computers

SUMMER 1983
$3.00

ATARI CONNECTION™

THE HOME COMPUTER MAGAZINE

Listening for Extra-Terrestrials with a Home Computer

0 74470 12432 32

WHY USER-WRITTEN SOFTWARE REALLY STACKS UP TO THE BEST

There's a lot of ATARI Computer users out there. And many of them write software. They really know what users like—and what users want. Because honestly, who knows more about users than other users?

The ATARI Program Exchange—APX—publishes user-written software for ATARI Home Computers. Which means all APX software is written *by* and *for* people just like you.

Their Home Management programs are written by people who manage their *own* homes.

Their Home Office programs come from people whose offices *are* at home!

The APX Personal Development line is written by those with a natural love for the subjects they choose.

APX games are written by game-players with one single-minded objective: FUN.

So next time you're ready to invest in some new software, see how APX programs stack up against the rest. You'll see why ATARI Home Computer users make such great software writers. In fact, you could get so carried away that you might end up writing your own software. If you do, send it to APX!

ATARI

A Warner Communications Company

LEADING APX TITLES

Entertainment

Attack!	Galahad and the Holy Grail
Getaway!	Outlaw**/Howitzer
Yachtman	Eastern Front (1941)
Quarxon	747 Landing Simulator
Downhill	Seven Card Stud
Salmon Run	

Learning

Counter	Mapware	Monkey Up a Tree
Math*UFO	Typo Attack	
Frogmaster	Cubbyholes	Wordmaker
Letterman	Number Blast	The Magic Melody Box
Starware	Spelling Genie	

Personal Development

Advanced MusicSystem	Astrology
Personal Fitness Program**	

Home Management

Family Budget	Weekly Planner
Family Cash Flow	Real Estate Cash Flow Analysis
Atspeller	

Systems/Telecommunications

Mapmaker	Chameleon
Disk Fixer	Extended fig-FORTH
DDT	ATARI Pascal Language
Supersort	BASIC/XA
Instedit	Deep Blue C Compiler

. . .and more than 100 other titles.

**Indicates trademark of ATARI, Inc.

APX—PROGRAMS FOR USERS BY USERS

SUMMER 1983 VOL. 3, NO. 2

CONTENTS

FEATURES

DEPARTMENTS

Cover Photograph by Mark Tuschman. Software Screen by Ted Richards.

ATARI CONNECTION

The Home Computer Magazine

EDITOR
Ted Richards

CIRCULATION MARKETING MANAGER
Julie Riegel

ART DIRECTOR
Charles Rush Denson

MANAGING EDITOR
Markene Kruse-Smith

PRODUCTION MANAGER
Thomas Clark

CONTRIBUTING EDITORS
Liz Metzger, Dr. Steven Young

SENIOR TECHNICAL EDITOR
Tay Vaughan

STAFF WRITERS
Jim Carr, Teddi Converse, Patti Donovan, Carlos Greth, Stephen Englehart, Gary Paul Fox, Myrna Rae Johnson, Paula Polley

CONTRIBUTORS
Bill Bartlett, Phillip Chapnick, David Fox, Jason Gervich, Herb Kohl, Lee Sherman, Kent Smith, Marianne Smith, Jane Sokolow, Cassie Stahl, Mitchell Waite

DIRECTOR MARKETING SERVICES
Rich Moyer

MANAGER, WRITING STAFF
Jim Inscore

EDITORIAL ASSISTANT
Myrna Rae Johnson

ADMINISTRATIVE ASSISTANT
Paula Surian

CIRCULATION CLERK
Hydee Decker

TYPOGRAPHY
CBM Type

PRINTING
W.A. Krueger Co.

Please submit articles to Editor, ATARI CONNECTION, Atari, Inc. Home Computer Division, 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 Home Computer Division, P.O. Box 50047, San Jose, CA 95150.

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

Letters From Our Readers

Dear ATARI CONNECTION,

I am 12 years old and I'm a sixth grader at Valley View Elementary Center. Some kids work with the Apple Computers in school, but I like the Atari better. We have the Atari 800, the 410 Recorder, the 810 Disk Drive, BASIC, PAC-MAN, Star Raiders, Missile Command, Galaxian, Choplifter, and we're going to add a modem, and a printer.

When I get older I'll get an Atari 1200XL Home Computer (it's great!). I really enjoy programming our Atari 800, and I'm going to attend a summer computer course (1 week) that my school is holding.

Keep up the good work on computers!

Cordially,
Jenn Hutchey
Archbald, Pennsylvania

■

Dear ATARI CONNECTION:

I read and enjoy your magazine very much. I have an Atari 800 computer, 810 disk drive and a cassette recorder.

While going through some back issues of your magazine (Summer 1982, Volume 2, Number 2) I ran across some graphics that were very interesting (Farrah and Cola). Since the programs for these graphics were not included, I was hoping that I might obtain these from you.

Sincerely,
Joseph M. Coory, Jr.
Harbor City, California

You're in luck! Our new Bits and Pieces section in this issue has the program listing for 'Farrah' by Bill Carris. Your second choice, Cola, may be featured in a future Bits and Pieces. Stay tuned!

Dear ATARI CONNECTION:

I have enclosed my payment for renewal of my subscription to Atari Connection based entirely on the quality of the last issue—prior to that, they have been almost trivial and juvenile in content. You might consider your audience of Atari computer owners: some novice, some experienced, some juvenile, some adult.

What I would like to see are:

- Novice level and more advanced tutorials.
- Useful Utilities
- News articles on new uses of Atari computers, new hardware, new useful software (Atari and outside).
- Less kid stuff—make a juvenile section or page, but keep it separate.
- Less blatent "party line" style. I believe Atari is the best computer for the money—I bought one! But quit trying to convince me on every page!

Thanks,
Tom Pillichody
Parkersburg, West Virginia

You have several valid points, about which we felt the same way. So, in this issue, you'll see our brand new Bits and Pieces section for both advanced and novice programmers. There is also a section featuring the latest Atari Home Computer product line-up. The Kidbits section is designed for children, but please note that the Find The Bug contest is open to all. Matter of fact, our winner in this issue is an Army chaplain!

Now, for our convincing "party line"—we have given the magazine content a shot in the arm by beefing up our technical editing support and giving a bigger focus to the technical side of the magazine.

■

Dear ATARI CONNECTION,

I've just started programming on my Atari 400 computer this year but I'm already having as much fun programming as I did playing games on my home computer. I find it hard to remember how I spent my idle hours before I purchased my system. I am at present awaiting the arrival of my Communicator II Kit and hope to have as much fun with it as I have with the Programmer Kit.

I've sent along an alarm clock program I've written. It's useful if you've misplaced or broken your real one.

Sincerely,
Pat Jones
Colby, Kansas

```
10 ? "What time is it now?"
20 ? "Hour=";:INPUT H
30 ? "Minutes";:INPUT M
40 ? "Alarm time?"
50 ? "Hour=";:INPUT A
60 ? "Minute=";:INPUT L
70 ? CHR$(125)
80 IF M=60 THEN H=1
100 ? "                Time"
110 ? "                ";H;":";
120 IF M<10 THEN ? 0;
130 ? M:X=0
140 IF H=A AND M=L THEN GOTO 200
150 FOR S=0 TO 320:NEXT S
160 X=X+1
170 IF X<60 THEN GOTO 150
180 IF M<60 THEN M=M+1
190 GOTO 70
200 FOR Y=1 TO 50
210 ? CHR$(253);:NEXT Y
```

Not bad for a young programmer! This alarm clock depends upon the execution speed of BASIC and hardware operations within your computer. The "timer" which Pat has used is located in line 150, and is a FOR/NEXT loop. The FOR/NEXT loop is commonly-used to create a delay in program execution, and Pat has carefully measured his delay in building his clock. The problem is that all computers do not execute at the same speed. A warm computer, for example, will execute more slowly than a cold computer. And if you remove the REM statement in line 70, the program runs faster because BASIC doesn't have to read these characters (you gain about two seconds every minute). The best way to build an accurate clock is explained in the Bits and Pieces section of this issue where you will find a sample program.

Is Programming Fun?

THE *LARRY KING SHOW* came over the radio. The all-night, national radio talk show marked the hour of 12:00 p.m., West Coast time. My concentration waned to a temporary lull as I noted Larry's guest for the evening. Hopefully, I'd finish debugging my "Typespec" program before having to listen to a taped replay of the interview—at 2:00 a.m.

The "Typespec" program was my first attempt at creating a little BASIC database to keep track of typesetting specifications for magazine articles. The idea was to enter each article by its title, then the number of lines and characters per line, then its typeset column width. I wanted all this information calculated to give me a line count and column length for the finished article. I could then save this information on diskette, and have it printed out. I was in way over my head. And not having fun.

The fun part had been thinking up the idea, sketching out the flow chart, and imagining how competent and intelligent I'd seem to our designer when I presented the magazine's manuscript with an attached computer print-out of all the article specifications neatly lined up in columns or whatever. My ambitious imagination had led me astray—taken me to the very end of my knowledge and beyond.

A baseball fan was on the line with Larry King. He talked about rookies in spring training. How they always hit home runs, made great plays, then were sent down to the minors for another year. The caller wanted to know if the rookies were so talented, why couldn't they play in the major leagues? I knew why. They had to learn the fundamentals—the secrets of the game— the little tricks of the trade that marked the difference between the amateur and the pro. They would be exposed to key game situations and their lack of polish revealed on a certain play. They would pick up that uncanny ability to hit the ball hard—put it in play despite being in the midst of a slump, or facing a hot pitcher.

I sent myself back down to the programming minor leagues. The foggy cloud of frustration lifted. Quickly, I began jettisoning features of my program. I really didn't need a print-out of the specifications. I'd let the computer do the calculations, then just note them in by hand alongside the article manuscript. Simple addition from that point could be done manually to get the total amount of typesetting.

I was finally having some fun—a release from the aggravation and tedium. I began writing funny instructions for the PRINT statements: "Just how long do you think this article's going to be? You're the editor. Try and write the headline in 15 characters or less—that's all the space this string array has. Have fun!" I ended up with a simple BASIC program that calculated the typeset line counts of each article. Our designer would be happy just to get a complete manuscript of all the articles with the captions, headlines, etc, period.

Yes, I think programming is fun—even in the "minor leagues." It's also a lot of hard work, tedious at times, and often frustrating. But the fun is there, hidden amongst the syntax errors, the bugs, the hardware snafus, and the lapses in "memory." I clicked off the radio seconds before Larry King's taped interview with an author. It was 2 a.m. I wasn't going to have much fun the next day, but I had fun that night. And I think you'll have fun with all the programming features in this special programming issue of ATARI CONNECTION.

—Ted Richards

HARDWARE REVIEW

The new ATARI XL family of Home Computers

THE LITTLE COMPUTERS THAT CAN

New Atari 600XL and Atari 800XL Home Computers: Full-Featured Computers at the Lowest Prices Ever.

THE NEW ATARI 600XL Home Computer looks simple enough. It's bantam weight, about half the size of its hardware sibling, the Atari 1200XL, and sells for under $200.

But looks and price can be deceiving. The low-priced Atari 600XL is a "true" Atari Computer that offers a ground-floor opportunity to participate in the exciting world of computers. New owners can gradually work their way up, building a complete computer system as their budget and needs allow.

With its standard 16K RAM and built-in Atari BASIC, the Atari 600XL has plenty of memory for the classic Atari Computer games and useful computing tasks, including word processing with AtariWriter. However, advanced software such as VisiCalc or the Atari Bookkeeper program requires at least 32K RAM.

But this new low-cost home computer doesn't stop here—the Atari 600XL has been designed to expand as the owner discovers more complex and useful applications. When it comes time to try programs requiring a relatively large amount of memory, there'll be a *48K Memory Expansion Module*, available from Atari Retailers by early fall. This module can expand the Atari 600XL Computer's capacity to a powerful 64K RAM!


Atari 800XL

The Memory Expansion Module plugs into the *parallel interface port*, conveniently placed on the Atari 600XL Computer's back panel. A new feature on all Atari XL Computers, the parallel interface port, allows users to connect a wider variety of advanced peripherals to their systems for special applications.

The low-profile, full-stroke keyboard is easy to master, with a layout similar to that of the Atari 800. A major improvement is that the control keys—OPTION, SELECT, RESET, and START—and the new HELP key are set flush in an appealing metallic band on the right side of the keyboard. The single cartridge slot is centrally placed above the keyboard so program cartridges can be inserted or removed while the computer is on without disrupting its operation.

Because it's compatible with most Atari hardware, the new Atari 600XL owner will have access to the large library of Atari software, not to mention the responsive Atari Customer Support and the nationwide network of factory-backed service centers.

Atari 800XL

The Atari 800XL brings you all the essential features of the Atari 600XL: built-in BASIC, a full-stroke keyboard, a HELP key, and an international character set. And, thanks to the memory (64K RAM), the Atari 800XL can operate a full-featured home computer system.

So, if you're looking for a fully supported home computer with the most memory for your money, the Atari 800XL is worth investigating. ■


Atari 600XL

THE ATARI 1400XL AND 1450XLD HOME COMPUTERS

The Best Just Got Better


Atari 1450XLD

Atari has just introduced two new top-of-the-line home computers: the Atari 1400XL and the Atari 1450XLD. Both models come with a built-in speech synthesizer and a direct-connect modem in addition to the advanced features found in other Atari XL series computers—a powerful 64K RAM, built-in Atari BASIC programming language, an external processor bus for future expansion, full-stroke keyboard, advanced sound and graphics capability, HELP Key, and programmable function keys.

The built-in direct-connect modem comes complete with ModemLink software which allows the convenience of dialing from the keyboard, and a memory buffer to save telecommunications data for later storage on diskette or cassette. With the huge array of information and services available, such as stock market quotes, electronic mail, databases, bulletin boards, personal communications and more, the telecommunications capabilities of the Atari 1400XL series is a welcome feature.

An additional bonus with the Atari 1400XL series is the on-board speech synthesizer, which can be programmed to generate phonemes (the individual sounds of spoken language) directly, or to "translate" from text.

The Atari 1450XLD Computer incorporates a dual-density, double-sided disk drive in the same compact package, eliminating the need for a separate disk drive and interconnecting cables. The dual-density format allows up to 127K bytes (about 100 typed pages) of information to be stored on each side of the diskette.

These latest two additions to the Atari XL family of computers join a product line that is already thoroughly supported with more than 2,000 available software programs, a well-established network of user groups, and readily available customer service. **by Gary Paul Fox**

NEW LINE OF PERIPHERALS CREATES ATARI XL SYSTEM

The recently introduced Atari XL family of computers has been joined by two new 80-Column Printers, a Disk Drive, Program Recorder, Direct Connect Modem, Touch Tablet, and Trak-Ball to form the newest generation of Atari Home Computer systems.

Styled to match the XL series, this new group of peripherals offers improved standards of performance and features. Let's take a closer look at what's in store.

THE ATARI 1025 80-COLUMN PRINTER

A Dot-Matrix Printer for Word Processing and Program Listings

If you need to print letters, manuscripts or other information quickly and economically, the inexpensive Atari 1025 80-Column Printer is ideal. It uses a dot-matrix print head to generate standard English alpha-numerics *and* European-language characters in a variety of formats.

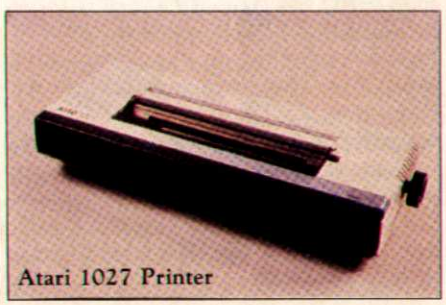
You can choose regular-width (80-column), condensed (132-column) or extra-wide (40-column) characters to suit your requirements. In addition, the Atari 1025 can accommodate standard typing paper in single sheets, fan-fold computer paper, and even roll paper with an optional holder. An easy-to-use side knob lets you adjust the paper just like a typewriter.

The engineers really did their homework when they designed the Atari 1025. Since it works directly from the Serial Input/Output port on all Atari Computers, there is no longer any need to purchase a separate interface module. The Atari 1025 is much quieter than most comparable 80-column printers (though a bit slower than some), and it also uses a typewriter-type spool ribbon that is a lot easier (and cleaner) to replace. In short, this new printer is a worthy addition to the Atari Computer Product line, and with a suggested retail price of $549.00, one that is bound to please many home computer users.

THE ATARI 1027 PRINTER

Letter-Quality for Under $300

The new Atari 1027 Letter-Quality Printer represents a low-cost technological breakthrough for letter-quality printing. Inside, it contains a five-wheel printhead that creates fully-formed letters like a daisy wheel, but at a fraction of the cost. With a respectable rate of 20 characters per second the Atari 1027 is


Atari 1027 Printer

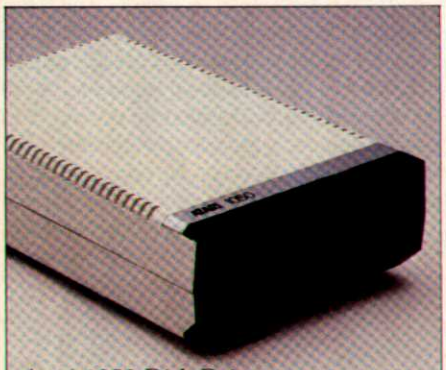
no speed demon, but it still operates much faster than most people can type. Best of all, it makes letter-quality printing greatly affordable at under $300 suggested retail price!

The Atari 1027 is ideal for use with a word processor like AtariWriter, and you'll find the compact Printer design is a great feature that helps fit a complete home computer system into cramped quarters such as a living room corner, home office, or a student desk in a dormitory.

THE ATARI 1050 DISK DRIVE

Fast, Efficient and Now Dual-Density

Disk drives have proven themselves as the fastest and most convenient way to store and retrieve information in a home computer system. Since they work as a Random Access Storage Device using file names, they are extremely quick, with a typical access time being a fraction of a second. The new Atari 1050 Disk Drive is no exception, and it also offers a dual-density format for greater storage capacity.

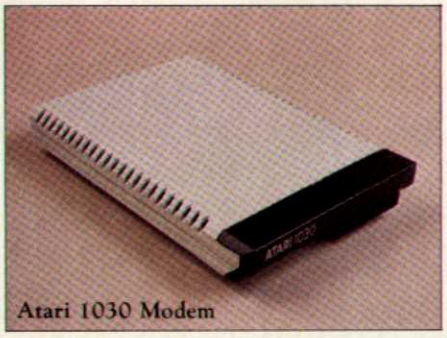

Atari 1050 Disk Drive

Atari has developed a new Disk Operating System III which allows the Atari 1050 to operate as a dual-density drive. That means even greater storage capacity (about 127K bytes per diskette). The new DOS III also allows you to convert any information stored on a single-density diskette to the new format, using the Atari 1050.

The Atari 1050 is a well-thought-out model, employing an improved loading mechanism that has a more positive feel than the old "door" type. All these features come in an attractively designed package about half the height of the previous Atari 810 Disk Drive. At a suggested retail price of $449, the Atari 1050 should become a very affordable addition to many Atari systems.

THE ATARI 1030 MODEM

Direct-Connect Convenience with Built-in ModemLink Software


Atari 1030 Modem

The Atari 1030 Modem is an economical, all-in-one accessory that turns your home computer into a telecommunications terminal. It plugs directly into any modular telephone jack with the included cord, and connects to your computer's Serial I/O port. The built-in ModemLink telecommunications program allows you to dial directly, using the keyboard on your Atari XL series computer for fast, efficient operation.

Other features include selectable pulse or tone dialing, full or half duplex operation, and memory buffer on/off and printer on/off functions. These last two options allow you to save your communications to cassette or diskette, and to simultaneously printout information as you are receiving it.

THE ATARI 1010 PROGRAM RECORDER

An Inexpensive Way to Save and Load Programs

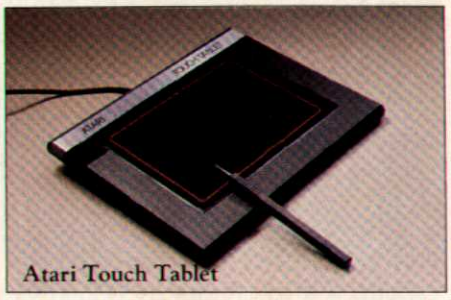
The new Atari 1010 Program Recorder is an intelligently engineered product. Styled to match the new Atari Home Computers, and compact enough to fit almost anywhere, it includes a host of features. With two Input/Output ports, you can plug the Atari 1010 anywhere in your system along with another printer, modem and other accessories. The 4-track, 2-channel format allows for computer programs or data on one channel, and pre-recorded sounds on the other.

The Atari 1010 has also been designed for convenient operation. An automatic volume control assures perfect record and playback levels, and the top-mounted control buttons have a positive touch. Its easy-to-read tape counter comes in handy when you're trying to find a particular program. The Atari 1010 includes a great many features for a suggested retail price of just $99.

THE ATARI TOUCH TABLET

Now, Graphics Are As Simple As Drawing with a Pen

The Atari Touch Tablet puts all the built in graphics power of your Atari XL series computer at your fingertips. You can use it to paint pictures, draw charts and diagrams, write in script, and even design computer graphics.


Atari Touch Tablet

It's straightforward and simple to operate and comes with a special diskette program for saving your graphics handiwork. Just load the diskette program into your computer, plug the Touch Tablet into controller port #1, and you're ready to start drawing. Put the special pen to the tablet and you create the designs and images you want to make . . . right on your own TV screen.

THE ATARI TRAK-BALL

Arcade-Style Action Comes Home!


Atari Tr

For you inveterate game players who've felt something was missing, get ready to roll into action with the new Atari Trak-Ball. It enables you to turn tighter corners, zero in faster on video villains and blast'em more accurately.

The Atari Trak-Ball works with any joystick game—Centipede, Missile Command, and Galaxian to name but a few—and lets you capture the ultimate in arcade action and excitement right at home. ■

SOFTWARE REVIEW

New & notable programs for your Home Computer

SPEED READING

Keeping Up With the Information Explosion

I must admit I was at first skeptical about the new Atari *Speed Reading* course as an effective learning tool. My experience with some educational software is that the graphics generally lack imagination and interesting visual appeal.

As I glanced through the workbook my doubts persisted. No pictures.

Then I began reading. What a surprise! The introduction gives a clear overview—from the need for Speed Reading to a detailed format of the course, including all information necessary to use the program. I began to take heart.

The directions for starting up the tapes, are thorough and clear. You are directed to the *Basic Reference Manual* if you get a loading error. If you don't have one, order a list of *Error Messages* and descriptions by calling Atari Customer Service.

I began the first unit with a "pretest" to determine my reading speed. The pretest also serves as an introduction to the business of using the joystick to start and stop the reading timer. When you have finished reading the piece, and have signaled with the joystick, a computation of your reading speed in words per minute is calculated. You then take a small quiz, enter your score (easily calculated), and these two scores are used to formulate your Reading Efficiency Index (REI), which will then be used to create a graph of your progress.

The potential difficulty of this approach is softened by the authors' encouraging comments. As I progressed through the lessons I became more and more impressed by the clarity of the directions, and it finally struck me that the visual severity that had dismayed me at first was a *planned feature*. How else to get the most out of a reading course than to force you to glean all the information by reading the material itself?

Course Outline

Four of the five tapes provided include the eight units or lessons. The fifth is a timing and pacing program to use with literature of your choice for future practice. The "pretest" mentioned earlier is only a preliminary exercise for the first time you do Lesson One, and it can be omitted thereafter.

1. *Warm-up Exercises.* These are designed to train your eyes to move quickly and consistently. You are given a highlighted word on your screen, followed by a list of words. As the highlighting window moves quickly to each word in the list, you signal the correct word with the joystick.

2. *Phrase-reading Exercise.* You are sped through the first 750 words of the first reading material. The window highlights each phrase at the pace you have previously determined (and which you may change during the exercise). This activity is accompanied by a pacing tone.

I had some trouble with this portion. It had the adrenaline-stimulating effect of an arcade game, which is excellent for heightening your motivation, but I found that it was easier to read the *un*highlighted phrases, and my eyes moved all over the screen instead of focusing on the window. This might change with more practice, but at least my peripheral comprehension was increasing.

3. *Paced and Timed Reading.* Here texts of increasing length are timed, and some of them paced (a tone-pacing metronome in the computer accompanies the reading.) The subject matter has been intelligently chosen for variety and interest.

4. *New Techniques.* This part is especially well done. A large variety of techniques—from simple, multiple-choice "finish the sentence" quizzes, to picking items out of a phone book page, to taking efficient notes—are all designed to focus the mind and eyes on determining key words and phrases. This is the essence of reading quickly: don't read what you don't have to. Read for "essence" as well as facts.

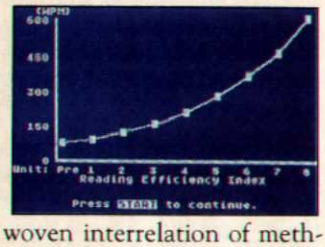
5. *Flexible Reading.* This means gearing your reading speed to the nature of the subject. The articles in this section are especially challenging for detail and unusual subjects (from an article about feet and foot care to a Chinese tract on animal reactions to earthquakes).

6. *Reading Progress Graph.* When you are finally through with your lesson and have logged in all your scores, you are presented with a graph of your progress. This serves as encouragement and visual reward for your efforts, which by this time have been considerable. This leads to an important question: Why a computer, rather than a simple timer and an audiocassette? Though I'm not certain the eye-pacing techniques worked

Illustrated by Lance Jackson

for me, they did get me interested in the project. The text's encouraging notes were clear and effective. In the end, I realized how *personal* this little machine is. It has made-to-fit tailoring of technology to people, at which the computer excels.

In addition, this course won my respect for its intelligent planning. What I first judged as tedious severity I later understood to be a richly



woven interrelation of methods. It claims to double your reading speed in a month, if two units are completed each week. I have only been using the course a few days, and judging from my progress I'd say it has set but a modest goal. **—Jane Sokolow**

Speed Reading *requires a minimum of 16K RAM, an Atari BASIC cartridge, an Atari 410 or Atari 1010 Program Recorder, and a joystick. The package includes five Program Cassettes and one Atari Speed Reading Workbook. Suggested retail price: $74.95.* ■

CAVERNS OF MARS

THE CAVERNS OF MARS war zone—Mars suffered heavy losses—indeed, the planet was totally destroyed several times—in recent conflict between Helicon VII starfighters and enemy forces deployed deep in the red Caverns of Mars.

"It was a joyride at first," said a proud Captain Dalf Halon after working his way steadily from a novice who cleaned zircon-encrusted outdrives to Commander of the Helicon VII, the most-feared starfighter in the Sol quadrant of the Milky Way Galaxy.

"I must admit I had some uneasy moments when I first flew into those blood-red Caverns," he said. "It took me a couple of seconds to get my bearings, and I almost crashed into the walls several times. I had a few close calls with enemy rockets still on their launching pads, but they were easy to bomb with my laser cannons.

"I blew up enough fuel silos to keep my ship's tanks nearly full, and putting the Martians' radio transmitters out of commission was like shooting Zackers on a sunspot.

"Getting to the bottom of the first Cavern wasn't exactly a snap, but I managed it without any great problems. That left me with a big job at hand: landing on the Cavern's fusion bomb and activating it," he hissed. "Naturally, that went off without a hitch.

"Then I got the bad news," Halon confided, his lips drawn tightly across his steel-tipped, pointed teeth. "My flight computer told me I had 20 seconds to get out of the Cavern before the fusion bomb went off, or I'd be blasted into orbit around the big galaxy in the sky.

"Now, I'm as good a Helicon VII pilot as there is this side of the Crab Nebula," he boasted, "But flying backward through this Cavern was a thriller.

"I had to keep the Helicon VII as low as possible to avoid hitting the walls of the Cavern," he added, giving less-experienced Helicon VII jockeys a look of contempt.

"I couldn't remember every little twist and turn," Halon crowed. "But the sight of the Cavern blowing its tubes was worth it.

"The action got a little wilder in the second Cavern," acknowledged the grizzled veteran of many ferocious space battles. "That's where I had to outmaneuver a seemingly endless onslaught of Greon and Pyxias rockets, all bent on committing harikari at my expense.

"But it didn't take me long to figure out where I wanted to be in this Cavern—there were fewer enemy rockets on my left than on my right. Once I caught on to this trick, it was like a Rullion turkey shoot," he laughed.

When Halon remembered the next obstacle, a hard, mean—almost cruel—look came to his deep-set, nova-red eyes.

"Then the Martians threw their laser gates at me," he sighed. "That's when I had to really get rough," Halon scowled as he exhaled a blast of air hotter than a jetwash. "It took me a few seconds—those dastardly Martians almost got me there—before I figured those gates out.

"I had to stop, then slip through when the first gate disappeared momentarily. Then, a series of three gates, each one as impossible to get past as a Grogon Natadile protecting its young—and no way to blast them to oblivion!

"But I'm nothing if I can't reach into my bag of tricks," added Halon, gazing longingly into the heavens, aching to return to the glory of battle. "And I needed every trick in the book to survive the next Cavern. Those space mines have a mind of their own. There's only one thing in the Galaxy that can move more erratically—that's me.

"It was touch and go for a while, I must admit. As soon as I blew one of them up, another'd take its place. They had me worried—not scared, mind you, I don't scare easily.

No, the wildest action was getting out of those Martian pits."

A look of near-panic crossed Halon's sweat-covered brow as he remembered his escapes from the Caverns of Mars.

"What'd I do to celebrate victory?" said Halon, a wicked smile returning to his face. "I challenged the Martians to another battle!" he cried, as he lowered the canopy on his Helicon VII spacefighter.

—Jim Carr

You too can take the plunge into the depths of Mars. Caverns of Mars, *originally an Atari Program Exchange (APX) game available only on diskette, is now available at your Atari retailer as a cartridge. Suggested retail price:* $39.95. ■

Jim Carr is a Senior Writer for Marketing Publications in the Atari Home Computer Division.

THE GREAT ESCAPE GAME FROM APX

NO SOONER HAD I gotten my copy of *Getaway* home to review than my brother and sister showed up at the front door looking for something to do.

"Gee, I got this new APX game, *Getaway*, you might like to try . . . "

"Sure," they chimed in unison, "we'll try anything once."

I handed my sister the diskette and went to the kitchen to make some coffee. Soon, the sound of wailing sirens and clinging cash registers was drifting in from the den, mixed with shrieks of excitement and moans of frustration.

On my return I found my brother huddled over the computer, joystick in hand, desperately trying to guide his escape vehicle out of the patch of an oncoming police car. With a flick of the wrist, he pulled a quick turn into the driveway of his hideout. But the cops had the place surrounded and cut him off before he made it to safety. As his last getaway car dissolved to an ominous tune, my sister said "my turn" and grabbed the joystick. Soon she was busy scooping up cash and diamonds off the street, the sound of sirens occasionally blaring in the distance. My brother sat in shock, offering her driving tips. "Great graphics," he said to me. "You

got any potato chips?"

I came back with the chips and started thumbing through the manual for *Getaway*. I found out that this game of cops and robbers had been designed by Mark Reid, a chemical engineer who's rapidly finding his niche among the leading Atari game programmers (Mark's other APX titles include *Solitaire* and *Downhill*). Piecing together 35 separate screens, Mark created this intricate, scrolling action game based on a toy car set he had as a child.

I watched my sister play for a while. As she raced down highways, past factories and an airport, across bridges, through neighborhoods lined with houses and parked cars, I had to agree that the graphics were superb. "You better get some gas," my brother pointed out. As she pulled into the gas station, the miniature pump flickered to life, ticking off gallons as her tank filled. Soon, she was on her way after an armored car.

Referring back to the manual, I found that *Getaway* has eight levels of play, with different prizes at each level. First come the diamonds, then the crosses, then hearts, and so on. After scooping up all three of the prizes on each level and returning them safely to the hideout (there's only one, and good players quickly learn how to find it blindfolded), you have to knock off the armored car to get to the next level. The only snag is the cops.

There are three police vehicles chasing your getaway car—two patrol cars plus the tenacious radar van. They tend to wander aimlessly about the town until they know you've pulled a heist. They're particularly eager to nab you if they know you've done a job on the armored car. Then they zero in from all directions and it takes some skillful driving, plus a sure knowledge of the roads and highways, to get back to your hideaway.

"Lookout!" exclaimed my brother, "that's a dead end!" Sure enough, my sister found herself cornered with enough loot to gag a gangster. As her last escape car faded, her score appeared at the top of the screen: "Hoodlum."

"I wasn't watching for the radar blips closely enough, I guess," she explained sullenly. "Any more of those hot dogs we had the other night?"

While warming up the hot dogs, I recalled reading that each of the patrol cars, as well as the armored car, has a distinctive blip that appears at the edge of the screen before it comes in on you. I guess that means these are high-tech crooks with radar of their own. At any rate, by keeping an eye on the edge of the screen, it's possible to stay out of the long arm of the law—for a while at least.

When I got back, my sister had just started her next game.

"How'd you do?" I asked my brother.

"Oh I only got to Petty Thief. I just wasn't paying any attention to the sirens." How he could do that, I couldn't understand. I'd been hearing sirens wailing since the start of the evening. Each of the police cars has its own distinctive siren. As they bear down on you, their wails get louder; as you move out of their reach, they fade away. It's a nice little touch.

At about one in the morning, I fell asleep on the couch, the wailing and moaning drifting in and out of my dreams. When I woke up and looked at my watch, it was 4:00 a.m.—and they were still at it. Gently but firmly, I asked them to leave.

"Thanks for the great evening," they said as they walked out into the dawn's early light. "We'll be back soon. Great game!"

I went into the den, picked up the joystick, and finally managed to get in a game of my own.

—Jim Inscore

Getaway—a high-speed chase game by Mark Reid. Available in cassette or diskette format. Requires 32K RAM. Suggested Retail Price: $29.95 ■

CLIMBING THE LADDER TO SUCCESS

How to Stay a Step Ahead of Donkey Kong

THERE'S NO DONKEY IN Donkey Kong*. According to our native Japanese sources, the game title is simply the name given to it by the Japanese.

The game begins with Donkey Kong holding (our hero) Mario's girlfriend hostage atop a structure of girders, refusing to give her up without stubborn resistance.

She stands trembling on the top of a building's towering girders, searching below; looks for Mario, her lover (don't ask how we knew *that*!), to rescue her. The gorilla, Donkey Kong, grunts and growls. He stomps around near her, daring anyone to get close. Spying the would-be rescuer, he hurls a huge barrel down the girders. Bap! Mario falls down on the job. Actually, he just lies there after being smacked.

You, as Mario, have just been annihilated. You have one more chance to redeem yourself, but first you need to know how to avoid the gorilla's crude shenanigans.

Timing is everything in this game—the time clock at the top of the screen ticks off points as you go. Go slowly if you're a beginner; learn the basic rhythms of the game first. Once you catch onto the tricks of the game, practice building up speed.

Getting up the ladders safely on the first level of girders is paramount to your overall success. Avoid going up the ladder when a barrel approaches, because it might turn quickly down the ladder to squash you. And you must be all the way up the ladder before turning left or right.

When you pass under a hammer (there is at least one on all levels of play), push the red firing button on the joystick to leap up and grab it. You can use the hammer to destroy barrels, vicious firefoxes, and the sand barrels. The hammer is only good for about 11 seconds. Barrels and firefoxes may slip under the hammer if Mario's arms are raised, so stand still when they approach.

Leaping objects takes fast fingers. Practice running and pushing the red button to jump. If two objects approach at once, get a running start, then press the red button—watch Mario do a flying broad jump with the greatest of ease. This skill will be especially important in the elevator level; Mario has to jump between elevator platforms to get up the structure.

When you get to the rivets level, grab as many of your sweetheart's belongings as possible—they're worth extra points. Make sure that you long-jump over the firefoxes, because they may turn around underneath you.

Elevators are the trickiest to master. There are a number of paths to take to get to the top, all of which depend on your ability to get across the moving elevators. Timing is crucial. Wait for the elevator platform to be just slightly below your jumping-off spot. Leap over to the platform, then jump immediately to the next beam. If you wait too long on the elevator going up, you'll make pizza ala Mario. The bouncing springs are dangerous and not worth the 100 points to jump them—best just to stay out of their way until you become a pro at this game.

The conveyer belt of moving sand buckets are a matter of side-stepping and jumping the buckets, avoiding the firefoxes and staying out of the belt's path. Even though it sounds confusing, these are all

*By Nintendo

things you can do with reasonable finesse by now, or you couldn't have made it this far in the game play. The big points here are for smashing the buckets with the hammer, worth from 300 to 800 points.

Mario's ability to successfully reach his lady and get his rewarding kiss depends mainly on your ability to make him jump around successfully. Once you have mastered this, all the rest of the maneuvers are mere gorilla play. ■

Available now. Requirements: joystick Controller, 16K RAM. Format: Cartridge. Suggested retail price: $49.95

– by Myrna Rae Johnson

Myrna Rae Johnson is the Editor's Assistant for ATARI CONNECTION.

JUGGLES' HOUSE

In, Out, Up and Down for Preschoolers

JUGGLES' HOUSE, one of Atari's latest educational games for youngsters from ages three to six, captivates preschoolers while teaching them spatial relationships. Juggles, an animated clown, encourages children to explore and review the differences between INSIDE and OUTSIDE and UPPER and LOWER. The friendly clown captures children's attention by sharing colorful toys in his onscreen playhouse.

Juggles' House begins with a picture menu display. Children who haven't yet learned to read can use the onscreen picture clues to help them play the game. Various speed options allow them to go through the program at their own pace.

The first learning game teaches the concepts of INSIDE and OUTSIDE using a blue cardboard window frame (enclosed in the program package) and prompts that are displayed on the television monitor.

An older child or adult can get preschoolers off to a good start by placing the frame over the computer keyboard and showing them that exposed keys are either INSIDE or OUTSIDE the frame. Turn the television volume up and point out that keys pressed outside the frame play a low note. Keys pressed inside play a high note. Exercises enable youngsters to explore and review these concepts. When all the questions are correctly answered, a delightful melody plays and the young game players are awarded an odd-shaped box to explore and review again. Two consecutive wrong answers, however, and hi-ho, hi-ho, it's back to the previous screen they go.

After these learning games are completed, Juggles' house will appear empty. The next challenge is to fill Juggles' house and yard with surprises again by choosing correct INSIDE and OUTSIDE keys. When the last object is correctly placed, the television screen comes to life. Birds fly, a dog wags its tail, logs crackle in the fireplace, and the chimney smokes. Depress any key to stop the animation. By continuing to press keys, players can make the objects onscreen disappear. This action fascinates young children and helps them learn to use the keyboard to interact with figures onscreen.

The next learning game enables children to explore the concepts of UPPER and LOWER. Before beginning to play, remove the cardboard frame from the keyboard, and position the blue cardboard overlay (enclosed in the program package) on the center two rows of keyboard letters. Letters on top of the overlay are UPPER; letters below the overlay are LOWER. Play begins with Juggles the clown sitting in his house juggling balls. A visual prompt says LOWER. When Juggles appears on the roof, the prompt reads UPPER.

The television monitor displays a bar similar to the one on the keyboard. By pressing any key, children can bring colorful toy blocks to life. When UPPER keys are pressed, a high note sounds; when LOWER keys are depressed, a low note sounds. Small children may need help to begin this lesson, but once they learn the basic concepts the review lesson should be a breeze. As in the first lesson, if the player makes two consecutive mistakes, Juggles will send him back to the previous screen.

In the next sequence, lines appear onscreen, producing a display that resembles lined primary-school writing paper. This is especially fun because a child can press any exposed keyboard key and see the corresponding letter, number, or symbol appear on the television monitor.

Once these learning games are completed, Juggles shares the toys on his toyshelf. Game players are asked to put the toys on the shelf by pressing UPPER or LOWER. When all the toys are on the shelves, children can continue pressing keys, rotating the toys.

– by Patti Donovan

Juggles' House comes in diskette or cassette format. Requires 16K RAM. Suggested retail price: $29.95. ■

FAMILY FINANCES:

Where Has All My Money Gone?

IN THE EARLY 1960s the folk song "Where Have All the Flowers Gone?" became quite popular among the younger generation. Today, members of all generations might well be singing an updated version that could go something like this:

Where has all my money gone?
Long time passing.
Where has all my money gone?
Long time ago.

If you have caught yourself singing this song lately, then the release of *Family Finances* by Atari may cause you to change your tune (and possibly your lyrics).

Two Programs in One

Self-taught programmer Jerry Falkenhan first submitted his "Family Cash Flow" and "Family Budget" to the Atari Program Exchange, where it was featured as an APX product for over a year. Atari has recently brought the two programs together into one complete home finance package and renamed it *Family Finances I* and *II*.

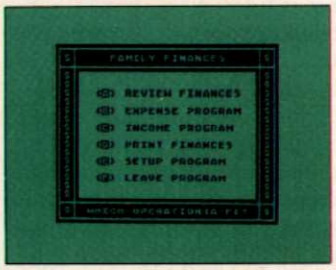
The *Family Finances* package contains two program diskettes—"Family Cash Flow" and "Family Budget." "Family Cash Flow" is used to record income and expense items; "Family Budget" creates a budget to help control how the money is being spent. "Family Cash Flow" will track as many as 100 expense entries and 20 income entries in 13 categories. (With the use of two disk drives, the expense entries can be increased to 220 monthly entries and the income entries increased to 100.) The maximum budget that can be tracked is $999,999,999.99 over a one-year period. If your budget goes higher than that, you can probably afford a megabyte computer system and your own programmer!

Family Cash Flow

Sample data for a hypothetical family is included in the *Family Finances* package to make it easy to understand how the program works. The first few pages of the manual take you through the steps of setting up the program for one or two disk drives. You can then begin a training session using the sample data provided on the "Family Cash Flow" diskette. When the Cash Flow program is loaded

the main menu appears with the following choices:

A) Review Finances
B) Expense Program
C) Income Program
D) Print Finances
E) Set Up Program
F) Leave Program



Option A) presents a review of the year's finances by month, stating the income, expenses and the variance.

Option B) is used to review, add, delete, or change entries in any of the 13 expense categories for a particular month. This option also provides a printed report with a detailed listing of items in expense categories for any month.

Option C) is identical to the "Expense Program" in its features and options, with the difference being that income, not expense data, is used.

The fourth menu item, D), requires a printer to be connected and will print out the information from "Review Finances."

Option E) allows you to set up your income and expense categories and erase the sample data or start a new year.

The final menu item, option F), exits the "Family Cash Flow" program, saves your information, and returns the computer to BASIC. This option *must* be used before turning the computer off, or any changes or additions made will not be saved.

Family Budget

Now that you know where your money is going with "Family Cash Flow," "Family Budget" will allow you to set up a plan to control your cash flow by creating a yearly budget using the data entered in "Family Cash Flow." After loading the "Family Budget" program diskette, you are greeted with the main menu, which provides the following choices:

A) Review Budgets
B) Change Entries
C) Print Budgets
D) Set Up Budgets
E) Leave Program

The A) Review Budgets selection brings up the Presentation Menu choices:

Yearly Presentations

A) Budget expense vs actual expense
B) Actual incomes vs budget incomes
C) Incomes budget vs expenses
D) Incomes actual vs expense budget
E) Single category incomes
F) Single category expenses

Monthly Presentations

G) Actual incomes vs budget incomes
H) Actual expense vs budget incomes

The Presentation Menu is very flexible and allows almost every kind of budget comparison that you can imagine for intelligent home financial management.

The next two items on the main menu, B) Change Entries and C) Print Budgets, are self-explanatory. The Print Budgets option will print data for any items in the Presentation Menu.

After using the sample data and becoming familiar with "Family Budget," the D) Set Up Budgets item is used to create a personalized yearly budget. The more months of entries that are entered into "Family Cash Flow," the more useful is the information obtained from "Family Budget." Therefore, it is better to wait until at least two months of data have been entered into "Family Cash Flow" before setting up your budget. Once the budget is set up, it can be reviewed or revised after each month of new data is entered. The object of the Set Up program is to fill in the budget data while using the actual figures as a guide.

The programs can be used to plan vacations, save for a home or a down payment for a car, or to create a long-term financial plan for your family's future. Although *Family Finances* may be used in a variety of ways, the end result is the same.

Once priorities become established, money is spent more wisely and less is wasted on unnecessary or impulse items. You'll find the guesswork of earning and spending money drastically reduced and a feeling of confidence and self-direction replacing the insecurity of not knowing where your money goes. Now, with a program like *Family Finances*, the sad lament, "Where Has All My Money Gone?" has become a song of the past. —**Jason Gervich**

Family Finances *requires a minimum of 32K of RAM, an Atari BASIC language cartridge, and one Atari Disk Drive. A 40- or 80-column printer and a second disk drive are optional.*

The program comes with a manual and two program diskettes—"Family Cash Flow" and "Family Budget." Suggested retail price: $49.95. ■

Jason Gervich is a Technical Support Specialist with Atari Technical Support.

DIGGING DIG DUG

DIG DUG HAS FINALLY tunneled his way into an Atari Home Computer. The object of the game is to dig tunnels while squashing monsters and eating fruits or vegetables.

The game starts out with Dig Dug tunneling his way to the center of the earth. A monster, Pooka, leaps out of tunnels to get Dig Dug, and a dragon named Fygar breathes fire enough to melt earth. Through tricky maneuvers in tunneling, Dig Dug can drop rocks on the monsters to kill them, or explode them with the red firing button on the joystick—only if he's quick.

After the first two rocks are dropped in any round of play, vegetables and fruits appear at the center of the earth for Dig Dug to scurry over to and eat for extra points.

Scoring is based on the number of monsters squashed with each rock, how much dirt Dig Dug digs when tunneling, the amount of vegetables and fruits eaten, and blowing up the monsters from different positions.

Understanding the action and developing a game plan in *Dig Dug* is relatively easy; mastering this game just takes practice. But be careful, because it is addicting!

—Myrna Rae Johnson

Dig Dug *comes in cartridge form and requires 16K RAM, and a joystick. Suggested retail price:* $44.95 ■

MICROSOFT BASIC II:

More Power For Your Atari Computer

REMEMBER THE LAST TIME you wrote a BASIC program? Did you wish you could delete a block of lines with one stroke? Did you want to make room for that new subroutine without retyping 50 lines? Have you grown tired of "pseudo-string arrays?" Well, take heart, Atari programmers, those days are over. You can do all of the above and much more with Atari Microsoft BASIC II, the new programming language from Atari.

Atari Microsoft BASIC II is very similar to the version of Atari Microsoft BASIC released about two years ago. The major difference is that Atari Microsoft BASIC II comes on a 16K cartridge and includes an improved user manual. Since Atari was unable to squeeze all the features of the original Atari Microsoft BASIC into a 16K cartridge, Atari Microsoft BASIC II includes a diskette that, when used in conjunction with the cartridge, provides all the

features of the disk-based version.

Of the ten BASIC commands requiring the extension disk, seven of them are developmental aids— AUTO, DEL, NAME . . . TO, RENUM, TROFF, TRON, and VERIFY—while the other three commands are used from within a program —DEF, NOTE and PRINT USING.

One of the major differences between Atari BASIC and Atari Microsoft BASIC II is the manner in which strings are handled. In Microsoft BASIC II, each string variable has a maximum length of 255 characters while strings in Atari BASIC can be as long as memory allows. However, Atari Microsoft BASIC II allows for the use of true string arrays as opposed to the "pseudo-arrays" found in Atari BASIC. In addition, *all* Atari BASIC strings have to be dimensioned while only multidimensional arrays or arrays with more than ten elements need dimensioning with Atari Microsoft BASIC II.

Although there is no syntax checking at the time of program entry, Atari Microsoft BASIC II does provide English error messages and the ability to trace errors right to their source with the debugging aids TRON and TROFF. Microsoft BASIC II also provides a choice of four types of numeric variables: integer, single-precision real, double-precision real or hexadecimal. Long subroutines can now be replaced with commands like PRINT USING, which lets you control the format of business reports to the screen or printer.

Microsoft BASIC II makes program conversion much easier. Most home computers use a version of Microsoft BASIC. Now it is possible to take easy advantage of the thousands of programs written for these other machines.

Atari Microsoft BASIC II opens a whole new world of creative programming. From the vast library of existing Microsoft programs to the endless possibilities of new creations for the advanced user, Atari Microsoft BASIC II releases more power than ever for the owners of Atari Home Computers. ■

—Kent Smith and Jason Gervich

PAINT

Create with Impunity!

ART AND TECHNOLOGY. Even the sharpest line drawn between them tends to blur at the edges. Nowhere is this more evident than in *Paint*—a software program that is not intimidating to people encamped on either side of the border that used to separate logic from creativity. *Paint* gives people who like drawing and/or computers a chance to sample both worlds.

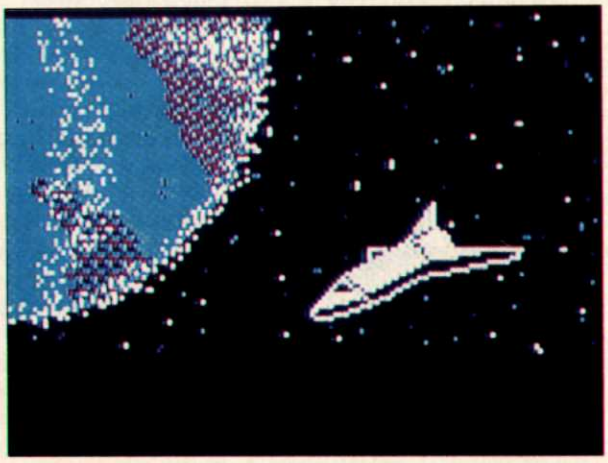
With *Paint*, you can create dazzling computer graphics and home-brew video art without having to program anything. In fact, it only takes about 30 seconds to slide in the program, pick up a joystick and begin doodling. Using the joystick as your brush, write your name on the screen in bold strokes, outline it with a narrow line, and fill in the middle with silly stripes or a colorful plaid. If there ever was a borderline between computers and creativity, *Paint* will have you tap dancing along its colorful contours in no time at all.

Paint includes three separate programs. "Artshow" displays images that are already created. To stir your latent artistic impulses, just sit back and watch as designs automatically appear on your screen. To create your own computer art, select "Simple Paint," or if you're feeling extremely creative, "Super Paint." Each of these programs lets you express yourself beyond your wildest dreams.

You can select the brush width you want to use, choose and mix your own colors, and fill in any enclosed area with any color. *Paint* also provides a delightful, instant op-art effect—patterned paint. How about a checkered horse? Or a striped balloon?

While "Simple Paint" provides only four brushes and four basic colors, "Super Paint" gives you a much more varied brush and color selection. In fact, you make these choices from separate screen displays by tapping simple one-letter commands on your keyboard. Pure colors, textured patterns, and built-in rectangles, lines and circles help you create masterful works of art.

You can paint an entire house in an instant or change the color of the background with three keystrokes. Use your joystick to define the radius of a circle, then watch the complete circle appear. *Paint* even lets you zoom into a drawing to provide a closer perspective, so you can draw lines and color areas almost pixel-by-pixel. Zoom back out to see the overall effect.



After you have completed a masterpiece, use the utilities functions to save your artwork on a diskette. Save an entire series of pictures and watch them appear, one at a time, in the sequence you choose. You can also get your picture back to the screen at any time for retouching, or instant-erase your art if you decide it isn't quite up to par.

A 175-page, spiral-bound instruction manual is included with *Paint*. There's a good section on "How to Paint," but the book also provides a lot of information that's only marginally relevant to the software it accompanies. Fourteen pages are devoted to a description of how computers work, for example, while 56 pages are given over to a history of art in the Western world, from prehistoric cave drawings to contemporary computer imaging. Creative writers ran amok in the manual's whimsical introduction and "Idea Shop" sections. The glossary may prove helpful to first-time computer users, however, and there's a bibliography for those who want to read more about computers and/or art. Fortunately, *Paint* is so accessible that almost anyone will be able to use the intriguing program without having to refer to its rambunctious documentation.

Paint was designed at SuperBoots, the software development lab at the Capital Children's Museum in Washington, D.C. Atari donated 30 computers to the museum in 1981. This contribution enabled the museum to put computer programs in exhibits and to establish a computer learning environment called Future Center. Here, children and adults become familiar with home computers through special classes and activities. *Paint* is among programs used in Future Center and other Capital Children's Museum exhibits.

—Markene Kruse-Smith

Paint requires 48K RAM, an Atari BASIC cartridge, joystick and Atari Disk Drive. Suggested Retail Price: $39.95 ■

KIDBITS

FIND THE BUG WINNER

by Steve Englehart

CHEERIO, CHAPS AND LASSIES! Speaking from Pago Pago, it is I, Lord Motley Bugnut, known throughout the civilized world for my study of *insecta* and throughout Atari for my (dare I say it?) classically elegant hosting of the Centipede game —and now these Find The Bug contests.

D'y'know, when first approached on the matter of these contests, I was, well, frankly hesitant about committing myself—ah, committing myself to the *judging*, eh? Yes, but I've become quite fascinated with your struggles to find these *bugs*. Why, when I began my association with Atari, they had such simple bugs—but in this most recent contest, only one in four could discover the little devil!

You see, my bug hunters, the colors in the pattern were meant to change randomly, according to the value of the letter R (which means Random) that lurks within line 10. Malicious Menconi has set the R value to 53770, which is the hardware register that generates random numbers. Therefore, each time the program obtains a number from this location [PEEK(53770) or PEEK(R)], the number is different, or random, as you would say. The God's Eye Program POKEs this number into three color registers (picked at random) in lines 30 and 40 [POKE 705+ RND(0)*8, PEEK(R)]. This caused the pattern to change color while it was performing its kaleidoscope light show—wonderful and mysterious to behold . . . but, to continue:

That bug was lurking in line 20, where 1538 was added to R when I=3. The bug was wearing R on his sweatshirt and therefore, pointed to the random-number generator only at the very beginning. After that, it mischievously pointed to another location, which always had a 32 in it. When the program POKEd into a color register, 32 is the color red. Thus, each time the sneaky bug picked a color register and put a "random" number into it, it knowingly and with malicious intent, systematically changed each color register to red and the God's Eye slowly disappeared from the screen.

Numerous bugfinders thought they "fixed" this bug by removing the statements which changed the colors. This prevented the pattern from being "eaten" by the bug, but also prevented the colors from changing—definitely not a solution!

Well, such bully sport in the matter of bug hunting has led me to inform the lads at ATARI CONNECTION that my services will remain at their disposal for the duration. And so, let us all salute this issue's winner: Chaplain (Major) Nicholas Czaruk of Ft. Sill, Oklahoma:

Dear ATARI CONNECTION:

*I have just received the Spring edition of ATARI CONNECTION, and the Bug is on page 14, line 20, where R=R+(I=3)*1538 should be deleted. There is already a sufficient RANDOM by PEEKing and R=53770. Line 20 should read POKE I, PEEK (R): NEXT I:FOR I=1 to 31. If this is done, the kaleidoscope of colors will not disappear.*

Our family bought our Atari 800 a year ago when we returned from Europe. We have the Recorder and will buy a Disk Drive later this year. Hopefully, we will buy the new 1025 printer in order to utilize the word processing capabilities of the computer.

My sons Nicky and Gregory, aged 11 and 7, are

Illustrated by Kim Dietch

learning to use the computer. They hope to attend an Atari Summer Camp to enhance their skills, and would like to see more educational software for the Atari, in addition to the games.

Sincerely,
Chaplain (Maj) Nicholas Czaruk
Fort Sill, Oklahoma

Well, who better to find the bug in a god's eye than a chaplain, eh? I shall personally mail this bright young fellow a Defender game cartridge from the post office in Pago Pago, and it will certainly arrive before 1985. In the meantime, all the rest of you should have a go at this issue's bug!

Jolly good luck!

God's Eye

by Craig Chamberland and Dave Menconi

```
10 R=53770:GRAPHICS 10:POKE 704,0:FOR I=705 TO 712
20 POKE I,PEEK(R):NEXT I:FOR I=1 TO 31:R=R+(I=3)*1538
30 POKE 705+RND(0)*8,PEEK(R):POKE 705+RND(0)*8,PEEK(R)
40 POKE 705+RND(0)*8,PEEK(R):FOR J=0 TO 31:COLOR C
50 K=I+J:J3=J*3:K3=K*3:J8=J+8:J71=71-J:PLOT K+7,J3
60 DRAWTO K+7,191-J3:PLOT 72-K,J3:DRAWTO 72-K,191-J3
70 C=C+1.5-7*(C>=7):FOR H=3 TO 1 STEP -1
80 PLOT J8,191+H-K3:DRAWTO J71,191+H-K3:PLOT J8,K3-H
90 DRAWTO J71,K3-H:NEXT H:NEXT J:NEXT I:POKE 77,0:GOTO 10
```

FIND THE BUG

THE COMPUTER LAID AN EGG

I've always been crazy about circles, so one night I sat down to write a program that would draw a perfect circle. Thinking back to my high school math class with Mr. Farquar, I recalled that the equation for a circle is: $1=Y^2 + X^2$. In Atari BASIC this equation can be translated into the statement: Y=SQR(1 $-X^2$). But if you simply PLOT X, Y you only get half a circle. To get a whole circle, you must also PLOT X, −Y. This was simple enough, so I turned back to my computer and wrote the short program you see here.

Somehow, though, my circle program still needs some rounding out—as you can see, the computer laid an egg. It is not a circle, but an oval! I've checked carefully and I'm sure that the equation is correct. Can you debug this program so it will plot a circle that looks like a real 360 degree circle?

Some of you, I'm sure, will *Find The Bug* and solve this mystery, while some may be as lost as I am. In any case, there's another problem with my circle program that you might wish to help me with. As the program now stands it's very slow! With a little brain work, I think you should be able to speed it up so we won't have to wait so long for the perfect circle.

THE EGG

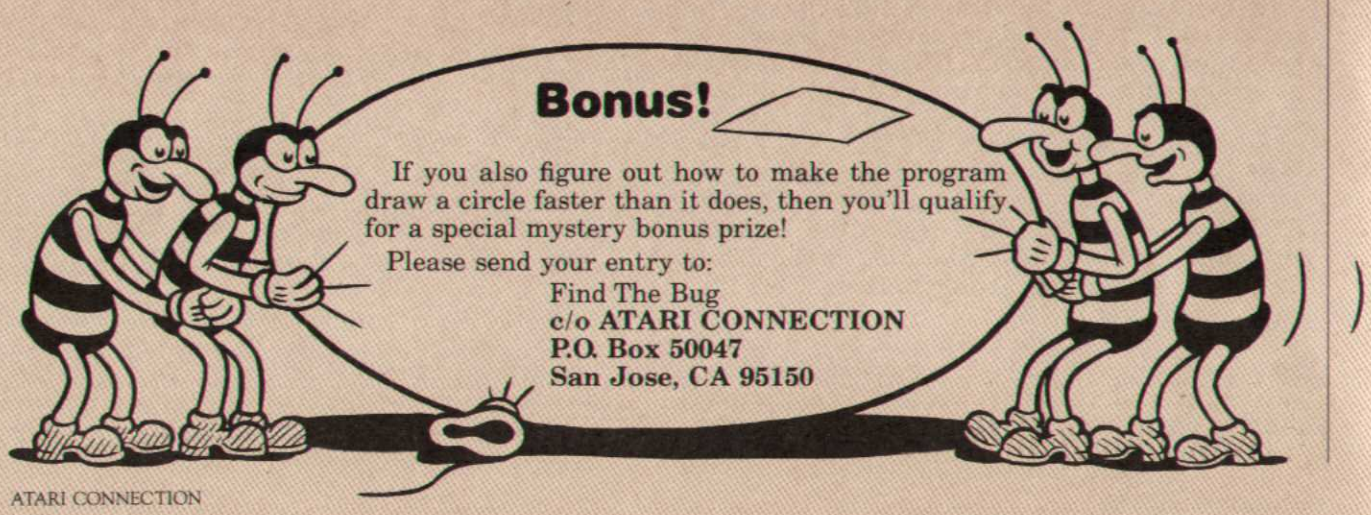
by Dave Menconi

```
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:X=COS(THETA)*SIZE
60 PLOT X+TRANX,Y+TRANY:PLOT X+TRANX,T
RANY-Y:PLOT TRANX-X,Y+TRANY:PLOT TRANX-
X,TRANY-Y
70 NEXT THETA
80 GOTO 80
```

FIND THE BUG CONTEST

If you can Find The Bug and make a perfect circle, then send us your corrected program along with a short story about yourself. If your entry is correct, you'll qualify for a special Find The Bug prize drawing for APX's hot new *Getaway* game!

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



Bonus!

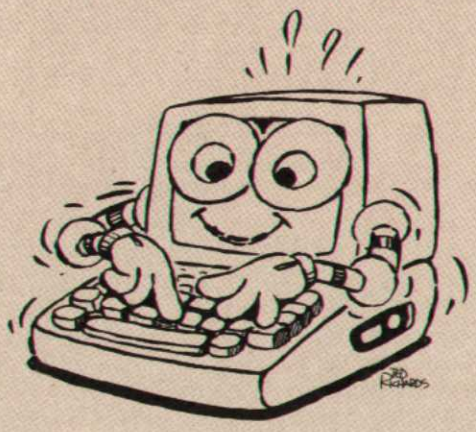
If you also figure out how to make the program draw a circle faster than it does, then you'll qualify for a special mystery bonus prize!

Please send your entry to:

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

RAFFLE

by Tom Hudson and Tay Vaughan



Here's a program you can have a lot of fun with. Not only can *Raffle* be used for "raffles" but it can help your family make tough decisions. Let's say there's a family argument over doing the dinner dishes. Mom and Dad are tired, want to relax, and don't have the energy to settle the argument. The solution? Use *Raffle* as a *Family Decision Maker.* Simply load the program into the computer, then when it's READY to RUN, enter the names of the "contestants" eligible to wash the dishes. Now press [START] and *Raffle* will spin through the names and announce the "winner" with a rolling musical fanfare!

Raffle can take up to ten names and each "raffle" can be repeated as often as you like. To hold your own computerized raffle, simply type the *Raffle* program lines into your Atari Computer exactly as they are listed. (Your screen listing will appear slightly different from the version on this page.) When finished, type RUN, press [RETURN], and just follow the *Raffle* instructions as they appear on the screen.

—Ted Richards

```
 10 REM *** display title ***
 20 GRAPHICS 2+16:PLOT 5,5:PRINT #6;"RA
    FFLE!"
 30 FOR I=1 TO 700:NEXT I
 40 REM *** choose ***
 50 TRAP 40
 60 GRAPHICS 0:SETCOLOR 2,0,0:PRINT :PR
    INT "ENTER NUMBER OF PEOPLE IN RAFF
    LE: ";:INPUT PEOPLE
 70 DIM NAME$(18),ALLNAME$(18*PEOPLE),B
    LANK$(18)
 80 POKE 752,1:FOR I=1 TO 18:BLANK$(I,I
    )=" ":NEXT I
 90 FOR I=1 TO PEOPLE
100 PRINT :PRINT "ENTER NAME ";I;": ";:
    INPUT NAME$
110 IF LEN(NAME$)<18 THEN NAME$(LEN(NAM
    E$)+1)=BLANK$
120 ALLNAME$((I-1)*18+1,(I-1)*18+18)=NA
    ME$
130 NEXT I
140 PRINT :PRINT "PRESS 'START' TO SPIN
    !"
150 IF PEEK(53279)<>6 THEN 150
160 GRAPHICS 2:SETCOLOR 2,0,0:POKE 752,
    1
170 C=0:R=INT(RND(0)*100)
180 FOR S=1 TO PEOPLE
190 C=C+1
200 PLOT 0,5:PRINT #6;ALLNAME$((S-1)*18
    +1,(S-1)*18+18);
210 PLOT 0,5:PRINT #6;BLANK$

220 IF R=C THEN 250
230 NEXT S
240 GOTO 180
250 PRINT CHR$(253):PLOT 0,3:PRINT #6;"
    the winner is"
260 GOSUB 330
270 PLOT 0,5:PRINT #6;ALLNAME$((S-1)*18
    +1,(S-1)*18+18)
280 PRINT "PRESS 'OPTION' FOR A SECOND
    CHANCE!"
290 PRINT "PRESS 'START' TO START OVER.
    "
300 IF PEEK(53279)=3 THEN 160
310 IF PEEK(53279)=6 THEN RUN
320 GOTO 300
330 REM *** play fanfare ***
340 DUR=20:V0=181:V1=141:V2=121:GOSUB 4
    00:DUR=7:GOSUB 400
350 GOSUB 400:DUR=9:V0=162:V1=128:V2=10
    8:GOSUB 400
360 V0=162:V1=128:V2=108:GOSUB 400:V0=1
    53:V1=128:V2=96:V3=193
370 FOR I=2 TO 14:SOUND 3,V0,10,I:SOUND
     1,V1,10,I
380 SOUND 2,V2,10,I:SOUND 0,V3,10,I:FOR
     J=1 TO 20:NEXT J:NEXT I
390 FOR J=1 TO 200:NEXT J:FOR J=0 TO 3:
    SOUND J,0,0,0:NEXT J:RETURN
400 SOUND 0,V0,10,8:GOSUB 420:SOUND 1,V
    1,10,8:GOSUB 420:SOUND 2,V2,10,8:GO
    SUB 420
410 FOR J=0 TO 2:SOUND J,0,0,0:NEXT J:F
    OR J=1 TO 10:NEXT J:RETURN
420 E=PEEK(20)+DUR+PEEK(19)*256
430 IF PEEK(20)+PEEK(19)*256<E THEN 430
440 RETURN
```

Tough Contest Generates Interesting Replies

Interviews with Four "Find The Bug" Near Misses

by Myrna Rae Johnson

"FIND THE BUG" contestants generate some amazing and resourceful "fixes" to the Bug, even when the answers are not quite right. The ages of these young creative readers generally range from 8 to 18 years old, though we do receive a fair share of entries from older Bug sleuths who are still kids at heart.

Curious about the kids who write, we picked four interesting-but-not-winning letters and phoned these dedicated boys. They range in age from a dynamic 10 to a polite 17; two of the four live in New Jersey, one in Michigan, and one hails from the Lone Star State of Texas.

All of these boys learned to program at school—most in a computer lab set up like a classroom. They like to experiment with programs, and enjoy teaching other students the new tricks they have developed. Far from being computer loners at early ages, these kids have joined computer clubs for the social interaction, as well as being a place to demonstrate their talents.

Our first interviewee is a dynamic young man who has done quite a "bit" for his 10 years. He's exuberant, polite, and a real charmer.

Andy Nichols

Almost 11 years old, Andy Nichols is in the 5th grade and lives in Austin, Texas. He earned his Atari 400 Home Computer and Atari 410 Cassette Recorder by mowing lawns in the summer.

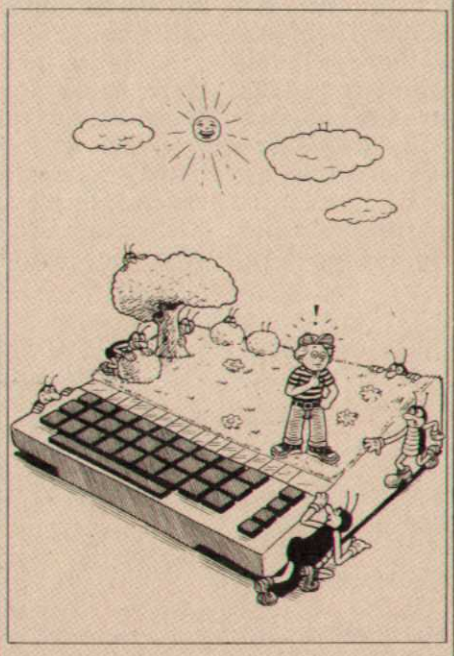
As with the other people we talked to, Andy learned to program at school. He learned to use Atari BASIC by using the book that was included in his Programmer Kit. Now he wants to buy a printer and a PILOT language cartridge to further his knowledge of computer languages.

Andy's school was the first in the state of Texas to set up an entire computer lab. As a result of this educational experiment, he has given a presentation at the Austin Hyatt Regency, has been interviewed by KLBJ radio station and a couple of TV stations. His class' project is to conduct political polls at election time. Their predictions have missed by only a 2–3% margin, which is pretty impressive—better than some national news networks!

Andy likes to write math programs for his own use: percent rates, random number sequences, etc. He has written some small game programs, and is working on a high-resolution game that he expects will take a couple of years to finish. We hope Andy submits his program to APX first—it's sure to be winner!

Steven Jacobs

A high school junior, Steve is from Paramus, New Jersey. Using the computers in his Computer Science I class, Steve used his class textbook to learn programming. He belongs to the Paramus High School Computer Club, and is known as "Mr. Turnkey" by his Computer Science teacher because he likes to write easy-to-use programs.



Because of his high interest in computers, Steve's family bought an Atari 800 Computer with an Atari 810 Disk Drive. Steve said he doesn't really care for video games—he prefers to program. He probably sneaks in a few rounds of games between math programs, because he expressed a preference for the E.T. game cartridge.

His best program application came during his former job at the local Burger King. In order to cut down the confusion that is routine when scheduling part-time employees, he computerized the work schedules.

Steve graduates next year from Paramus High. Prepared for anything, he has sunglasses, tanning lotion and hipwaders ready to go if he's accepted into the University of California, Berkeley, to major in Systems Analysis or Computer Programming.

John Huber

A Durand, Michigan, high school junior, John used the self-paced program on his school's computer to teach himself BASIC. At Christmas, he received an Atari 400 Computer and ATARI 410 Cassette Recorder from his family. Computers run in the family—John's dad works in a data processing group.

A polite, dedicated young man, John belongs to an informal, school-sponsored club of about 10 members. The club brings their own personal computers to a member's house during school breaks to show off the new programs they've written.

John is most interested in game programs. Of the Atari games, he enjoys playing Star Raiders the most. He has written a couple of games on his Atari Computer, and is working on making them longer, with more animation.

Though he is interested in software, John wants to become an Electronics Engineer, where he'd like to work with hardware systems.

Tom Loscalzo

From Martinsville, New Jersey, cheerful Tom started learning about computers in the third grade. Now, at age 10 and in the fifth grade, Tom is a self-admitted computer addict who enjoys taking other students into the lab at school to show them how to use a computer.

Tom's family bought an Atari 800 Computer last year as a family project. Tom and his dad belong to the Jersey Atari Computer Group, and go to meetings once a month to catch up on the latest goings on.

Tom is also a big fan of computer games. His favorite type of program to write is a game that involves player/missile graphics. He knows how to write in joystick and paddle controller movements, as well as the BEEP and BOOP sounds that make games fun.

Besides computing, Tom likes to work on his model railroad, where he is currently finishing the installation of wiring, transformers and switches.

With all of his talents and abilities, it's hard to remember that Tom is so young. But Tom hasn't let any of it go to his head—he's still a realist. When asked if he had any plans for the future, he replied, "It's hard to say. I'm only 10 years old." ■

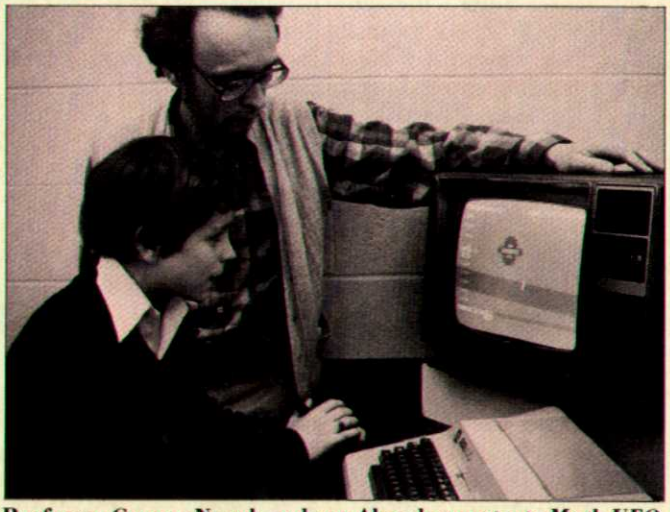
HOME COMPUTER NEWS

A Gift to the Meek

When Indianapolis, Indiana, Professor Gregor Novak brought home an Atari 800 for his sons one Christmas, he had no idea he'd soon be writing an award-winning computer game.

It all started one day when his youngest son, Alex, asked him for some help with basic math drills and handed him a pack of flash cards. "Why not write a math drill program for Alex on the Atari?" Novak asked himself. It was the beginning of *Math*UFO*.

Novak didn't have a programming background, but with a stack of computer books, he set to work. Two weeks later he had the basic outline for *Math*UFO*, an arcade-style math game designed for beginners. The following spring he entered his game in the quarterly Atari Program Exchange (APX) contest. *Math*UFO* won second prize in the educational category.



Professor Gregor Novak and son Alex demonstrate *Math UFO*.

"I was absolutely shocked when I won," says Novak, who teaches physics at Indiana University–Purdue University at Indianapolis. "Basically, I devised the program for Alex, but I also wanted to understand how the computer works so I could apply it to my own teaching."

*Math*UFO* is designed for children ages 8 to 12 and can be played by one or two players. As a space ship with a math problem floats to the top of the screen, two columns of numbers appear, one for each player. The object is to find the correct answer, position your missile launcher underneath, and fire at the UFO. Whoever gets the first strike tallies up the most points. Single players can sharpen their math and coordination skills by changing the game level.

Novak raised the game level to accommodate older children, with options for faster UFO and missile-launcher speeds. Players also have their choice of addition, subtraction, multiplication, or division.

Novak donated his entire prize—$1,500 worth of Atari educational hardware and software—to Brebeuf Preparatory School in Indianapolis, where his oldest son is a senior and his middle son will be a freshman next fall. Atari's Educational Sales Department added another $500 worth of equipment.

Novak specified that his prize be placed away from the school's well-equipped Media Center, which computer-shy students find intimidating. Today, Novak's computer sits in a room by itself, a friendly haven for the curious and the meek.

*Math*UFO is distributed by the Atari Program Exchange, P.O. Box 3705, Santa Clara, CA 95055. Toll-free (800) 538-1862 (outside California) and (800) 672-1850 (in California). Suggested retail price: $22.95 for either cassette or diskette version. Requires an Atari BASIC language cartridge and one Atari joystick per player.*

ATARI YOUTH ADVISORY BOARD

Top Students Gather to Share Ideas

It was raining outside, but the real storm—well, brainstorm, anyway—was taking place inside as 20 teenagers plumbed their minds for ideas, what-ifs, and why-nots that may someday help Atari stay one step ahead of its competition.

The 20 teenagers were the first selected to serve on the Atari Youth Advisory Board, and they were visiting Sunnyvale and San Jose, California, in mid-March to give Atari executives their suggestions, criticisms, and opinions about what personal computers mean to them and their friends now—and what they might mean to them in the future, too.

"Tell us about software in particular, that's the key," Atari Chief Executive Officer Raymond E. Kassar emphasized during his welcoming comments to the Advisory Board. "Look at it from your point of view and tell us about it."

"We've got a lot of things in the back of our minds that you can do for us," Atari Home Computer Division President John C. Cavalier promised later in an informal give-and-take meeting with the Board members. "It's very important to get the group off to a good start, and that means you have to be honest with us."

The calendar said it was the start of spring, but the Advisory Board members were greeted with two days of thunder and lightning, wind and rain as one of the wettest winters on record continued to soak Northern California. The 20 teenagers talked with the press, saw a private screening of a Warner Communications movie, spent a night on the town in San Francisco, saw a laser-light concert, and, naturally, played a few video games in Atari Expo, Atari's arcade game room.

The idea for the Youth Advisory Board came from a friend of Ray Kassar's who asked Helen Gray, Atari's Vice President of Public Affairs, to check into the feasibility of such a program. Eventually, a list of about 120

EDITED BY LIZ METZGER

names of outstanding junior high and high school students was compiled by a consulting firm. The names came via nominations from principals at schools known for their computer education curriculum as well as from Atari Institute Executive Director Ted Kahn.

"We asked the school principals to select students they would pick if they were putting together such a board," said Atari spokeswoman Noreen Lovoi. "Bottom line, the children all had to be computer literate. They didn't have to be Atari literate—they had the opportunity while here to work with Atari computers and tell us what they thought about them in comparison to what they've worked on. And they all had to be well-rounded students, not just interested in computers.

"Another factor was age—students were mostly 14 to 17—and we wanted a representative list of male/female and various ethnic groups. We were very much looking for a good mixture."

What the selection process came up with was a diverse and talented group indeed. Among the Advisory Board members were a ballerina, Anneke Wyman of New York City, who has danced in the "Nutcracker Suite"; the captain of a high school football team, Paul Sunshine of New York; a budding entrepreneur, Tracey Cullinan of Los Altos, California, who has founded his own successful software consulting firm; and two television stars, Matt Laborteaux of "Little House on the Prairie," who couldn't attend the Board's first meeting, and Todd Bridges of TV's "Diff'rent Strokes."

The group also included David Lurie, the son of San Francisco Giants baseball team owner Bob Lurie; Daniel Janz, a 17-year-old Chicago, Illinois, student who has programmed in seven computer languages on five different types of computer; and Musa Mustafa of Walnut, California, who has won awards for his work in film graphics animation.

The Advisory Board visit at Atari included a tour of the company's LSI (large-scale integration) research center, where the integrated circuits that go into Atari Home Computers are developed and designed; a meeting with Howard Warshaw, the free-spirit game writer who programmed *Yars' Revenge*, *Raiders of the Lost Ark*, and *E.T. Phone Home!* for Atari; and several brainstorming sessions in which their minds were probed for ideas about computers, video games, and telecommunications networking.

The prospect of the network was one that obviously had the Advisory Board members excited. They were quick to suggest various uses for such a network—a problem-solving hotline, news about third-party software, and electronic mail were just a few of the ideas that came up.

—Jim Carr ■

A Match Made in Heaven

The Cable TV Connection

If you live in southeastern Michigan, you may already subscribe to "Metro 13." It's the cable TV channel that flashes local milk prices, used car ads, and restaurant guides across your screen.

What you may not know is that the programming source for "Metro 13" is an Atari 800 Home Computer that's been running nonstop since last September.

"Metro 13" is the brainchild of Chuck Miller, a computer professional who leases channel 13 from OMNICOM and operates the station in his "spare time."

Subscribers have access to 24-hour consumer information in seven different categories. These include everything from movie, restaurant, and shopping guides to "Deals on Wheels," a guide to automobile sales, parts and servicing. A comparison shopping guide changes every Monday, while items such as classified ads are updated twice a week.

Miller decided to write a computer program that would update his broadcasts automatically, a process that he previously did by hand.

"It used to be that I'd have to run down to the station a couple of times a day to change the tapes," he explains. "Now all it takes is an operator down there a couple of times a week to enter new information."

Today, over 14,000 homes in five Michigan communities subscribe to "Metro 13." It is, as Chuck Miller puts it, "a perfect blend of two new and exciting high-technology areas—home computers and cable television."

For more details, contact Chuck Miller, 44237 Kingsley Court, Canton, MI 48187. ■

Word Processing, Programming, Games

New All-In-One Paks From Atari

The Writing System

It's 2 a.m. You've just finished retyping the tenth draft of a five-page term paper for a 9 a.m. Shakespeare class. Then you discover it—two missing paragraphs on page four! As you roll another sheet of paper into the typewriter, you remember those word processors on the "Lou Grant" TV series. Ah, wouldn't it be nice to have your own word processor? But don't they cost thousands of dollars? Not *The Writing System*, the new All-In-One-Pak word processing package from Atari.

With *The Writing System*, you have everything you need to write letter-quality term papers, business correspondence, reports, and anything else you type for less than the cost of an advanced electronic typewriter.

The Writing System includes an Atari 600XL, *AtariWriter* word processing program, Atari 1027 80-column, letter-quality printer, an informative guide to writing, and a packet of high-quality letter-writing paper.

Available this fall.

The BASIC Tutor I

This summer, Atari introduces *The BASIC Tutor I*, an Add-A-Pak designed for everyone who wants to learn Atari BASIC. *The BASIC Tutor I* is aimed at novice programmers who haven't yet mastered BASIC for the Atari Computer.

In *The BASIC Tutor I*, you'll find an *Invitation to Programming, 2 and 3; Inside ATARI BASIC* by Bill Carris, an easy-to-read, friendly approach to BASIC; and a book of programs.

Compatible with the entire line of Atari Home Computers. The Atari 400, Atari 800, and Atari 1200XL require a separate Atari BASIC cartridge. Available August, 1983. Suggested retail price: $59.95.

The Arcade Champ

What happens when you take two of Atari's most popular games, pack them into one box, then offer that package for a price your family can afford?

You have Atari's *The Arcade Champ* Add-A-Pak, containing PAC-MAN and QIX, two family favorites.

The Arcade Champ includes the PAC-MAN and QIX game cartridges, a pair of joysticks, and a convenient cartridge case for storing your program cartridges. Compatible with all Atari Home Computers. Available August, 1983. Suggested retail price: $89.95 ■

Expand Your Memory

Now 48K For Atari 400

If you're the owner of an Atari 400 Home Computer, and you dream about expanding its memory, dream no further. This summer, Atari is introducing the *48K Memory Expansion Kit* for the Atari 400.

Applicable to both the 8K and 16K Atari 400 Home Computers, the 48K Memory Expansion Kit expands the memory of the Atari 400 to the maximum memory available for the Atari 800.

In practical terms, this means that a whole world of software and peripherals previously not available for the Atari 400 will now be yours for the asking—peripherals like disk drives, and software like VisiCalc and The Bookkeeper, plus the whole world of APX and independent software for the Atari 800, the largest software library available for any home computer.

The 48K Memory Expansion Kit is available as of June, 1983. It includes a plug-in memory board, wire jumpers to modify the Atari 400 mainboard, instructions, and a 90-day warranty for parts.

You can buy the 48K kit either installed or uninstalled.

The installed version is available for $154.95 at Atari Regional Service Centers. You can also purchase it at all local Atari Factory-Authorized Service Centers. Check the Yellow Pages or call toll-free (800) 538-8543, outside California; and (800) 672-1404, in California.

The uninstalled version is available for $134.95 through Atari Program Exchange, P.O. Box 3705, Santa Clara, CA 95055. Recommended only for people familiar with electronic assembly kits. ■

"E.T." Fan Club Formed

Here's news for all you E.T. fans. Our extra-terrestrial wire service reports that International Fan Club Specialties, Inc., has started an E.T. Fan Club. Its goal, according to IFCSI president and fan club director Ira Friedman, is "to enhance the E.T. experience and help keep the film's magic fresh in mind."

Organized under the direction, supervision, and authorization of movie director Steven Spielberg's office and Universal Studios, the fan club has over 25,000 members all over the United States.

For $6 a year, club members receive a special fan club kit. It includes a five-minute record of highlights from the film; an 8 x 10 color photo of E.T. hugging his friend Elliott; an E.T. coloring poster; an official membership certificate; a fan club membership card; and four illustrated issues of the official quarterly club newsletter, E.T. COMMUNICATOR. ■

For more information write to Official E.T. Fan Club, P.O. Box E.T., Mt. Morris, IL 61054.

NOW THE CLUB MED/ATARI COMPUTER VACATION IS BIGGER AND BETTER THAN EVER.

LAST ISSUE we told you about the exciting Club Med/Atari Computer vacation which happens this summer. Well, Club Med and Atari are moving the computer vacation getaway to Punta Cana, a larger Club Med village with a spectacular array of activities and facilities.

Situated in the Dominican Republic in the Caribbean, the village of Punta Cana is perched on the eastern-most tip of the island of Hispaniola, along a reef-protected white beach. Coconut palms and groves of mangoes line the shore . . . and, of course, so do Atari Home Computers.

Guests will be able to learn about computers in a workshop setting, and through the various activities offered at Punta Cana. Vacationers will use Atari's Tennis Organizer to rank players for tournaments on the village's ten tennis courts. They'll also be able to enjoy a game of RealSports Tennis on the computer while catching their breath between sets. Through a computerized marine life guide, snorkelers will be able to obtain information on which fish, coral and sponges to look for—as well as which ones to avoid. And, after sailing on the reef-protected sea, some old salts will no doubt practice the rules of seamanship by sailing an official regatta course on the computer.

An arts and crafts workshop lets guests use an Atari Computer to design a weaving pattern or paint a tropical landscape. Children will explore computers at the village's Mini Club—a club within a club with activities and facilities designed especially for children.

In addition, the entire village will be linked through an electronic bulletin board. Through this computerized network, guests will be able to make reservations at one of the restaurants, leave messages for new friends or find out what activities are planned for the day.

There's still time to grab your bathing suit and get in on the fun. The excitement starts June 11th and runs through September 10th. Most people stay for a week, some stay for two. For more information on the Punta Cana Club Med/Atari Computer Vacation, call your local travel agent or the toll-free Club Med number, (800) 528-3100, Monday through Saturday. ■

Atari Pilot Wins Software Award

Atari PILOT with "turtle" graphics came out a winner in the Learning Computer Software Awards for 1982 sponsored by The Learning Periodicals Group of Palo Alto, California.

Winners were selected for their quality of instructional design, suitability for the intended audience, creative use of instructional information, and error control.

Atari PILOT was chosen, according to the panel, as "an outstanding program that effectively introduces children and beginners to computers and computer programming." The program was also cited for the ease with which it allows teachers with limited computer experience to design curriculum materials with the computer.

The awards were coordinated by Dr. Odvard Egil Dyrli, Educational Technology Editor of *Learning* magazine, one of three magazines published by The Learning Periodicals Group.

Congratulations, *Atari PILOT*!

Listening for Extra-Terrestrials with a Home Computer

Is There Intelligent Life in the Sigma Draconis Star System? Amateur Radio Astronomer Karl Lind Will Use an Atari 800 to Find Out!

AMIDST A SMALL FLURRY of excitement and controversy, NASA recently announced funding of an official project to start a Search for Extra Terrestrial Intelligence—or SETI, as the acronym flies. SETI is a relatively small affair in this age of megabuck Research and Development—only 1.5 million—yet it has touched a raw nerve.

SETI provokes controversy because the question of whether we are alone in the universe or not has significance far beyond most other astronomical projects. Universally accepted evidence of contact with other intelligent beings in

by Phillip Chapnick

Photographed by Mikkel Aaland

Amateur Radio Astronomer Karl Lind mans his prototype personal SETI (Search for Extraterrestrial Intelligence) Station.

the universe is still nowhere to be found. UFO research remains on the fringes of legitimate science. E.T. notwithstanding, most scientists still don't believe that there's anybody out there.

The resistance somehow seems to satisfy a deep anthropocentric belief that *we* are the only intelligence in the universe. But if you look through an astronomy book at pictures of stars—of which there are billions in our own galaxy—and then realize that there are untold millions upon millions of other galaxies out there, each with billions of stars, it seems totally incredible that there *isn't* other intelligent life in the universe.

Yet, empirical evidence that other beings existed out there with a biochemical-based intelligence similar enough to ours to make communication possible could be profoundly disturbing. We could no longer take solace in the fact that at least we were the center of intelligent life in the universe. Undoubtedly such a historical event would irrevocably change our understanding of the metaphysical cosmos. And so the search to answer the question: Are we alone in the universe?

Scientific SETI

THE FIRST REALLY SCIENTIFIC SETI project was put together for about $1000 by Dr. Frank Drake, an astronomer at Cornell University. That was back in the early sixties; with the primitive equipment he had on hand, Frank Drake's OZMA project, as it was known, could barely scratch the surface of the possible star systems to search, not to mention all the different radio frequencies that might be carrying an intelligent signal. The rapid development of computers and digital technology has dramatically changed the picture. Funded by the private Planetary Society, a SETI project conducted by Dr. Paul Horowitz of Harvard University should be operational as you are reading this article. The NASA project being funded is in the final stages of development of a specially designed system that can analyze 78,000 adjacent frequency channels at once. The NASA system could do more searches in several minutes than the OZMA project could do in a hundred thousand years.

What's even more exciting to us little guys is that even a talented amateur astronomer can get in on the ground floor of SETI research—thanks to the rapid development in home computer technology. It's not for everybody, of course; but for the persistent few there's a chance to be the first to receive intelligent signals from space. Karl Lind is one of us little guys, hot on the trail, searching for extra-terrestrial intelligence.

Deep Space DXer

KARL'S NOT ONE OF THOSE dreamy spaced-out types either. He spends his workdays as an engineering assistant at SRI International, located in Palo Alto, California. He's been working there for the past ten years now and loves every minute of it. "The thing I like most about my job is how varied it is," he told me one evening as we met to talk about his own personal search for extraterrestrial intelligence. "I get to work on just about everything from solid state amplifiers to spectrophotometers. And I've worked on projects ranging from anthropology through zoology."

Karl, like many other radio amateurs, or HAMs as they like to call themselves, is an action-oriented man. Kept out of sports when he was young by a congenital heart defect, he turned his keen competitive spirit towards bridge and chess. "If I were younger, of course I'd be playing with computers. I didn't have the opportunity when I was young . . . It's just a matter of generations."

There're lots of stars out there, so where do you start?

Karl dropped out of Michigan State University in the early sixties and drifted into San Francisco during the heyday of the city's beat poetry movement. He managed a slot car track there for five years and then landed a job repairing Dictaphones. Finally, he heard of a Dictaphone repair job at SRI, secured an entry-level position there, and worked his way up—without much formal training in electronics—to a position roughly equivalent with a starting electronics engineer; an accomplishment of which he is duly proud.

All this time Karl maintained an active interest in amateur radio. An advanced HAM radio operator, he's worked on just about every frequency allocated to amateurs. He's transmitted on the amateur radio operators' satellite system and has been active in HAM television. The system he's designing now is really just a modification of some of his ham radio equipment, which is designed to operate on the 1.3 GHz (that's 1.3 billion cycles per second) amateur band. In essence, he's still doing the same type of thing he's been doing all along—listening for broadcasted signals—only this time he's tuned into the Milky Way!

Where Do You Begin?

THERE'RE LOTS OF STARS OUT there, so where do you start? After much study, Karl has targeted *Sigma Draconis*, a star fairly similar to our sun and also relatively close to us—only 18.2 light-years distant—as his first target. The choice is not capricious. Not only is Sigma Draconis nearby by galactic standards, but just as important, it is a *circumpolar star.* That means that it is close enough to the North Pole never to set. Even though you can't see it during the day because the sun is out, it is always above the horizon and therefore observable by radio telescope. Because amateur radio observations for SETI have to be made over long periods of time—sometimes as much as two weeks or more for just one series of measurements—it helps if you have a star which is continuously in your view so you don't have to worry about your equipment getting out of syncronization in the interim times when the star isn't visible. Observing towards the North Pole also means that the telescope will be pointed away from the noisy radio waves of the ecliptic plane (the sun's path through the sky). It's along this arc that most of our stationary earth orbit satellites are currently placed, and radio waves coming from them could make the detection of a true signal from the beyond much more difficult to identify.

Base Station Earth

THERE ARE TWO BASIC ITEMS needed to turn an ordinary 1.3 GHz amateur station into an amateur SETI station. (One hesitates to use the word "ordinary" here. There're probably only several hundred people in the entire U.S. who have the basic equipment necessary to begin SETI work; maybe a dozen or so in the San Francisco Bay area that Karl knows of.) You have to have a very, very accurate electronic clock to keep all of your equipment precisely syncronized, and you need a computer to perform some pretty sophisticated signal analysis.

Finding the right computer was easy. The Atari 800 is, hands down, the computer best-suited to the SETI task. "To my knowledge, it's the only one that meets federal standards for RF shielding. And it's got the lowest RF noise factor I've seen in a microcomputer. As you can imagine, this makes it the ideal choice for someone, like me, who's got to eliminate every possible source of extraneous, potentially contaminating, noise", says Karl.

What's in a Clock?

FINDING THE RIGHT CLOCK IS another story, though. "Clocks are the only place where the cost factor is a problem for an amateur like me. Everything else has come down in price to where it's reasonable except the clock," he lamented as we looked over his workbench—crowded with test equipment and the wires and cables that form the heart of the system.

"The basic change [to convert a regular system to SETI] is to phase lock the oscillator [in the receiver] to a highly accurate frequency standard, rather than the crystal oscillators as in an ordinary receiver. I need a Cesium or Rubidium standard [similar to the ones used, say, at the National Bureau of Standards in Washington, D.C.]. That's the heart of any system." The clock is what makes everything run together and enables you to make your measurements continuously over the long periods of time necessary to extract any possible signals from the far more plentiful random noise.

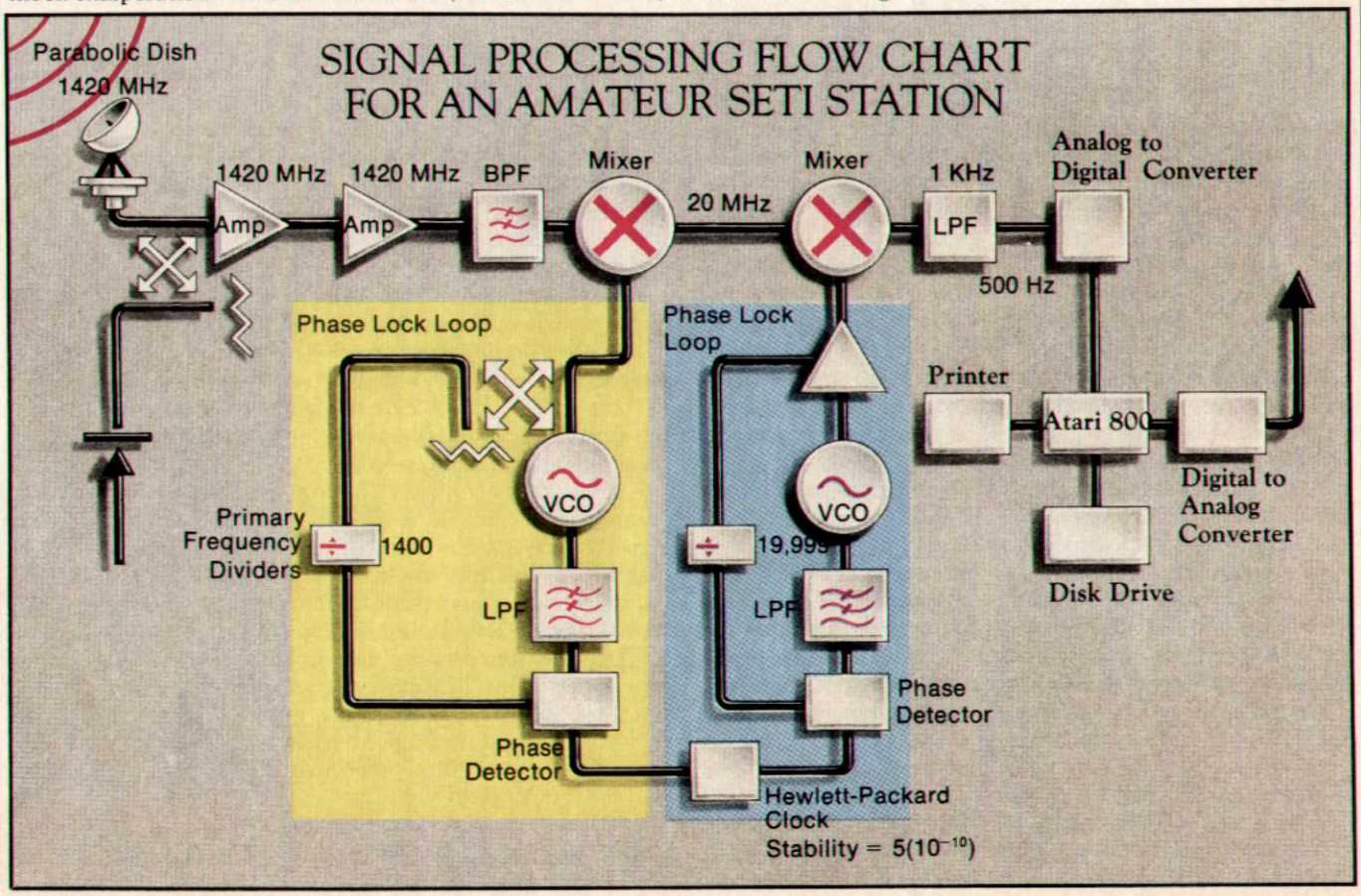
Currently, Karl is using a quartz standard clock accurate to about five parts in 10 billion. For full-scale searching, he's going to need a clock that's a full order of magnitude more accurate, say one part in 100 billion. Cesium clocks are well-nigh impossible to find on the surplus market where Karl gets the majority of his hard-earned equipment, because they rarely deteriorate. "They either work or die," he told me in mock exasperation. "Rubidium standards, on the other hand, are sometimes available. They drift off from their assigned frequencies, so they are no longer useful for the particular test job they were designed to do, but they are still stable enough to be used by the enterprising amateur."

The rest of the equipment is pretty garden-variety, high-frequency electronics gear. The antenna, for example, is the same type of disk antenna used for picking up satellite broadcasts. The bigger the antenna, the better, since the effective gain increases with the square of the radius of your dish. For testing, Karl is using a little six-foot-diameter dish, the absolute minimum that can be used, but he plans to step up to one at least twice that size when full-scale searching gets under way.

Surplus Anyone?

IT'S ONLY THE EXTREMELY RAPID pace of technological development that has made Karl Lind's personal search for extraterrestrial intelligence possible. Over the five years that he's been working to put his system together, he's estimated that he's accumulated more than $20,000 worth of test equipment. If you ask him how much it's cost him, he answers with a hearty laugh of the wise amateur, "Probably more than that, but at least it was spread out over time so I could afford to pay for it." Once the system is fully developed, though, most of the test equipment won't be needed anymore, and Karl estimates that within a couple of years you'll be able to buy an off-the-shelf SETI system, sans clock, for less than $5000. And who knows what'll be possible on the surplus market by then.

The trick to *AMSETI* (you guessed it, amateur SETI) is time. "I have a reasonably good, working microwave lab. I can do anything that can be done in a larger lab; it just takes a little more knowledge on the part of the operator and more time making the measurements, and then more time spent at a calculator working out the measurement results." Companies



Illustrated by Jim Ludtke

"It's somehow satisfying to believe that we're the only intelligence in the universe."

like Hewlett-Packard make equipment that automates these functions, but the equipment is very expensive and out of amateur range—$50,000 and up. The good part is that all this fast, smart, new equipment makes the older, dumb, equipment obsolete in the eyes of industrial users to whom time is money. So, perfectly good, although slow, equipment is becoming more and more available to the amateur through a network of surplus stores and special electronics flea markets springing up all over California's Silicon Valley.

Number Please?

THE BASIC PRINCIPLES BEHIND Karl's procedures come straight out of the signal-processing theorists' textbooks. The basic idea behind it all is that by taking repeated measurements of a signal and averaging those measurements together, you tend to eliminate random noise and enhance any signals buried beneath the noise. Karl will be taking samples of the incoming signals from Sigma Draconis, of approximately one second in duration, then will feed them into the Atari Computer for averaging and analysis.

The Atari Computer performs the actual analysis by first using an A to D converter to transform the samples into digitized data. This raw data is then arranged into a mathematical order that allows it to be searched for sine or cosine waves at a precisely defined frequency. Because most natural sources of radiation have a rather wide band spectrum, finding a spectrally pure signal emanating from space would be a good indication that some artificial source was creating the signal. It's the averaging process which requires the ultra-accurate clock. Any significant jitter in the sampling times makes it impossible to extract the signal buried deep under the noise.

The mathematical algorithm used is called a Fast Fourier Transform. What it does it to mathematically break up the incoming signal into narrowly defined frequency channels and determine how much signal power there is at each frequency. A sharp concentration of energy in any one frequency band is a possible hit. Karl, and many of the professional workers in the SETI field, are looking for pure sine or cosine waves at a frequency somewhere near 1.42 GHz.

Why 1.42 GHz? In space, 1.42 GHz is the frequency at which hydrogen absorbs energy, so space is relatively quiet at that frequency since there's an abundance of hydrogen to dampen signals at precisely 1.42 GHz. This energy absorption creates a kind of corridor of radio silence, which makes 1.42 GHz an excellent choice to use as a carrier frequency on which to send signals out into the depths of space. Presumably, any civilization advanced enough to broadcast into space would also be aware of the radio properties of space and try to concentrate its efforts at this same logical spot in the RF spectrum.

Cosmic Connection

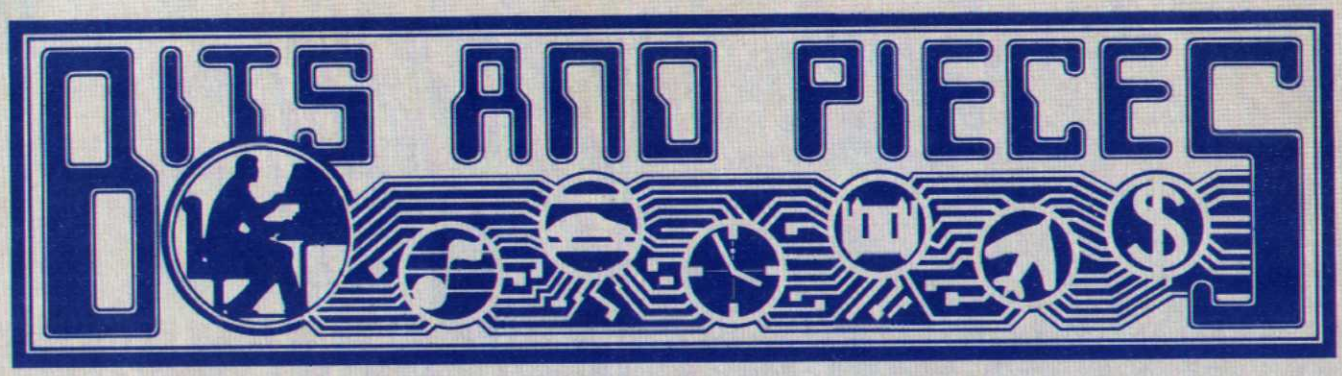
AS YOU MIGHT EXPECT, THE ACtual electronic process is a wee bit more complex. Here's roughly what happens: The signal focused by the antenna is fed to a very low-noise, very sensitive preamplifier, usually mounted right at the focal point. This incoming signal is then mixed with a very precisely controlled 1.4 GHz frequency signal. The output of the mixed signal is equal to either the sum or the difference of the two input frequencies. Although you could look at both, Karl is currently planning only to look at the different frequencies that are generated, which would mean that the signal after one mixdown is now in the 20 MHz range. This mixed-down signal is then mixed again with another frequency source which further decreases the bandwidth—the range of frequencies contained—into selectable 10 kHz segments. It is this 10 kHz signal that is actually fed to the computer for averaging and signal extraction. The second frequency source is program controllable so that the computer itself can automatically step through different 10 kHz pieces of the sample spectrum looking for signals.

Currently, Karl is planning to use an algorithm which will examine about one hundred adjacent frequency channels at a time. Each search of a range, say from 1 to 2 kHz or from 6 to 7 kHz, could take as much as two weeks of continuous signal processing to determine the presence or absence of a signal. Special considerations also have to be made to account for the *Doppler shift*—the phenomenon that causes a signal to change frequency depending upon whether the source is moving towards or away from the receiver, all of which increase the complexity of the computational tasks. Just doing the mathematical calculations to perform a 1000-channel Fourier Fast Transform is enough to keep the Atari 800 computing at full bore, so Karl is planning to use a special-purpose central processor board just to perform the frequency analysis. Ultimate control of the data processing would be under the control of the Atari 800, which would also be responsible for steering the radio telescope so it can keep Sigma Draconis on target for the entire analysis period.

As with every other piece of equipment in the station, the Atari Computer will have to be connected to the master clock so that it will know exactly when to start taking each waveform sample for the averaging. Otherwise there are no special modifications needed.

Karl hopes to get his system running within the next couple of months, but he's in no great hurry. For Karl Lind, it's the pursuit of the unknown and the challenge of communication that keeps him interested. He's picked SETI as a project because he wants something he can work on in his retirement. He's taking his time to do it right. "My search is an amateur SETI because I'm funding the whole thing myself, out of my own pocket, but it'll be professionally planned and executed." Who knows, he may just be the first to separate out of the cosmic crackle the signal from the stars that changes our deepest perceptions of our place in the universe. Or it could be *you*. ■

Philip Chapnick is a free-lance writer and editor specializing in science and technology.



by Tay Vaughan

IN THIS ISSUE OF THE ATARI CONNECTION we begin an exciting new section for and by Atari programmers. Here, beginners and experts alike will find a serious collection of handy routines and short programs that perform useful and interesting programming tasks.

As you study and apply these routines in your own programming efforts, your programming skills will improve and you'll gain easy access to unexplored features of your Atari Computer. Some routines will be simple utilities for tasks like entering dates and managing names and addresses. Others will help you build players and missiles, change characters in memory, animate figures, or play music.

Bits and Pieces can also save you time. It's no secret that computer programming takes time. Professional software programmers use many standardized subroutines that perform common programming tasks. Any technique you can employ that helps eliminate the tedium of writing code that hundreds of programmers have already written is well worth using!

When you take advantage of *Bits and Pieces* we hope that you will enjoy that same sense of satisfaction and discovery that Thomas Jefferson must have felt when Eli Whitney showed him his new method for assembling muskets. From a pile of disassembled parts, Jefferson picked pieces at random from various bins and assembled a complete working rifle. No longer did each individual component of the musket have to be carefully hand-crafted and fitted! As you build up your collection of these bits and pieces and assemble them into working programs, you may find that your programming becomes quicker, more efficient, and easier.

We also hope to see *Bits and Pieces* as a place where Atari programmers can show off their bag of tricks and share them with other users. Atari programmers are creative people, proud of their discoveries and accomplishments. So dust off your reference manuals and join in the fun by using and contributing to *Bits and Pieces*. If you have a few nuggets of logic (or "sweet kernels of brilliance" as one poet programmer has said), *Bits and Pieces* will allow the rest of the world a peek at your handiwork. Just organize your program or subroutine into the "submission format" outlined on page 29 and mail it to ATARI CONNECTION.

How to Use 'Bits and Pieces' Subroutines

TO USE MOST OF THE subroutines listed in *Bits and Pieces* simply type the subroutine into your computer and LIST it to cassette or diskette for storage. Then, whenever you wish to use it, ENTER the subroutine into your main program (reserving the appropriate line numbers). LISTing and ENTERing techniques are discussed in the Atari BASIC Reference Manual on pages 10 and 25.

Be sure to check that the variable names of values and strings in the subroutine don't conflict with those of your main program. Then make the subroutine work for you with a GOSUB to the first line of the subroutine. If you use the subroutine more than once in your main program, you will probably wish to relocate any DIMension statements in the subroutine to the first lines of your main program because you can only DIMension a variable once unless you CLR (Clear) *all* variables. If IOCBs are used in the subroutine (these are the Input/Output Control Blocks used for diskette, screen, keyboard and other peripheral operations), check that the IOCB number(s) don't conflict with IOCBs already OPEN elsewhere in your main program. If you are lost by now, the Atari BASIC Reference Manual will be able to answer your questions!

Once you master this programming technique, you can amass a useful library of handy standardized routines which you can quickly apply in your own programming work, saving many hours of effort. Don't hesitate to customize a subroutine so that it works precisely as you wish. What the subroutines give you is a kernel of logic which can be tailored to your own special needs!

Illustrated by Hank Osuna

Crafty Routines!

Hungry for nifty nuggets you can slip into your own programs to do a job quickly? Try these on for size . . .

Date *by Tay Vaughan*

At some time or another everyone uses a date in their programs: for records of birthdays and anniversaries, or to keep track of things like checkbook balances and supermarket prices. Sometimes it's difficult to decide how to enter the day, month, and year so that your data is organized and can be sorted, displayed, and handled in a standardized manner.

The subroutine uses character I/O (PUT and GET) to give you string variables representing month, day, year, and date. These can be quickly changed to values using the VALue function. By changing position statements in lines 50, 60, and 200, the screen display may be relocated. The ? CHR$(125) statement in line 40 may be removed to avoid clearing the screen. To make the screen look nice, line 60 contains inverse video blanks in the print statement (toggle the inverse video function to "on" by pressing the ATARI key when you type the blanks in this statement line).

```
 10 DIM MO$(2),DA$(2),YR$(2),TD$(
    2),DATE$(6)
 20 OPEN #1,4,0,"K:"
 30 OPEN #2,8,0,"E:"
 40 POKE 752,1:? CHR$(125):POSITI
    ON 3,8:? "ENTER TODAY'S DATE:
    "
 50 POSITION 23,9:? "mo da yr"
 60 POSITION 23,8:? "  -  -  ":PO
    SITION 23,8
 70 FOR I=1 TO 3
 80 FOR J=1 TO 2
 90 GET #1,X
100 IF X=155 THEN ? CHR$(253):GOT
    O 50
110 IF X<48 OR X>57 THEN ? CHR$(2
    53):GOTO 50
120 TD$(J,J)=CHR$(X):PUT #2,X:NEX
    T J
130 IF J<3 THEN FOR K=1 TO 2:TD$(
    J,J)=" ":PUT #2,32:NEXT K
140 IF I=1 THEN MO$=TD$
150 IF I=2 THEN DA$=TD$
160 IF I=3 THEN YR$=TD$
170 IF I<>3 THEN PUT #2,45
180 NEXT I
190 DATE$(1,2)=MO$:DATE$(3,4)=DA$
    :DATE$(5,6)=YR$
200 POSITION 3,15:? "HIT RETURN I
    F OK, 'R' TO REPEAT":GET #1,X
210 IF X=82 THEN GOTO 40
220 IF X=155 THEN CLOSE #1:CLOSE
    #2:RETURN
230 ? CHR$(253):GOTO 200
```

Dollar *by Tay Vaughan*

This subroutine converts any value (AMT) not greater than 9999999.99 or less than −9999999.99 to a right-justified string of ten characters (DOL$) in "dollar format." Thousandths are rounded off, and zeroes are inserted when appropriate. This routine may be used to provide neatly printed or displayed financial data. If a dollar sign ($) is desired, it must be printed or displayed in the calling program.

Sample Application:

10 DIM DOL $(10), AMT $(10): REM move DIMension statement to main program
*20 AMT=RND(0)*RND(1)*1333: REM create random value, positive*
30 GOSUB 110: REM call subroutine
40 PRINT DOL $: REM print results to screen
50 GOTO 20: REM get another value to convert

```
100 DIM AMT$(10),DOL$(10)
110 SIGN=SGN(AMT):AMT=ABS(AMT)
120 AMT=INT((AMT+5.0E-03)*100)/10
    0
130 AMT$=STR$(AMT)
140 DLEN=LEN(STR$(INT(AMT)))
150 CLEN=LEN(AMT$)-DLEN
160 IF CLEN=0 THEN AMT$(LEN(AMT$)
    +1)=".00"
170 IF CLEN=2 THEN AMT$(LEN(AMT$)
    +1)="0"
180 J=0:DOL$="          "
190 FOR I=10 TO 11-LEN(AMT$) STEP
     -1
200 DOL$(I,I)=AMT$(LEN(AMT$)-J)
210 J=J+1:NEXT I
220 IF SIGN<0 THEN DOL$(I,I)="-"
230 RETURN
```

Clock *by Kent Smith and Tay Vaughan*

Here's a routine that keeps track of time by using the Computer's Real Time Clock which counts each television frame. A completely new television picture is displayed each 1/60th of a second, and by counting these pictures or "frames" you get a pretty accurate clock! Now you can keep track of time in your game programs and you can make tests which ring the bell when time is up . . . PRINT CHR$(253) rings the bell.

While the CLOCK subroutine listed here is simple, it should be easy (Ha! Ha!) for you to figure out how to set the clock to a specific time (you will need to add some variables) or to make a 12-hour clock instead of a stopwatch. Try it!

```
 10 ? CHR$(125):POKE 752,1
 20 POKE 20,0:POKE 19,0:POKE 18,0
 30 GOSUB 100
 40 POSITION 16,10:? HRS;":";MIN;
    ":";SEC;"  "
 50 GOTO 30
 90 REM the clock subroutine
100 TIME=(PEEK(20)+PEEK(19)*256+P
    EEK(18)*256*256)/60
110 SEC=INT((TIME/60-INT(TIME/60)
    )*60)
```

```
120 MIN=INT((TIME/3600-INT(TIME/3
    600))*60)
130 HRS=INT((TIME/216000-INT(TIME
    /216000))*60)
140 RETURN
```

Varnames *by Kent Smith*

Attach this routine to the end of any program by ENTERing it off a storage diskette or cassette or just type it at the end of your program. Then call the routine with a GOTO3200 on the first line of your program. You will be able to list the names of all the variables you used in your program (including P, XYZSTART, XYZEND, XYZCOUNTER, and XYZCHARACTER which are used in the subroutine). This is very helpful in a big program when it's easy to loose track of the variable names which you have used.

```
32000 PRINT CHR$(125);"VARIABLE T
      ABLE LIST":PRINT :POKE 752,
      1:P=0
32010 PRINT "PRESS 'P' TO PRINT L
      IST":PRINT "PRESS 'S' FOR S
      CREEN ONLY"
32020 IF PEEK(764)=255 THEN 32020
32030 IF PEEK(764)=10 THEN OPEN #
      4,8,0,"P:":P=1
32040 OPEN #5,8,0,"E:"
32060 XYZSTART=PEEK(130)+PEEK(131
      )*256
32070 XYZEND=PEEK(132)+PEEK(133)*
      256-1
32080 FOR XYZCOUNTER=XYZSTART TO
      XYZEND
32090 XYZCHARACTER=PEEK(XYZCOUNTE
      R)
32100 IF P=1 AND XYZCHARACTER>127
       THEN PRINT #4;CHR$(XYZCHAR
      ACTER-128)
32105 IF XYZCHARACTER>127 THEN PR
      INT #5;CHR$(XYZCHARACTER-12
      8):GOTO 32120
32110 PRINT #5;CHR$(XYZCHARACTER)
      ;
32115 IF P=1 THEN PRINT #4;CHR$(X
      YZCHARACTER);
32120 NEXT XYZCOUNTER
32125 IF P=1 THEN PRINT #4
32130 PRINT #5:CLOSE #4:CLOSE #5
32140 POKE 752,0:POKE 764,255
```

Disklook *by Tay Vaughan*

Put this routine in your program and you can see your files on your diskette at any time! Just call the subroutine (GOSUB 9600) whenever you want without having to take time out to call up the DOS Utility Menu, examine your diskette, and then reload your program.

```
9610 DIM FMS$(18):KEY=1:SCR=2:
     DSK=3
9620 OPEN #KEY,4,0,"K:":OPEN #SCR
     ,8,0,"E:"
9630 ? CHR$(125):? :? :? :? :? "E
     NTER NUMBER OF DISK DRIVE: "
     ;:GET #KEY,X:POKE 752,1
9640 IF X>52 OR X<49 THEN ? CHR$(
     253):GOTO 9630
9650 DISK=X-48:PUT #SCR,X:? :? :F
     MS$(1,1)="D":FMS$(2,2)=CHR$(
     X):FMS$(3,6)=":*.*"
9660 TRAP 9670:GOTO 9680
9670 ? CHR$(253):? :? "ERROR-CHEC
     K DISK DRIVE!":GOTO 9750
9680 OPEN #DSK,6,0,FMS$
9690 TRAP 9750
9700 INPUT #DSK,FMS$
9710 IF FMS$(2,2)<>" " THEN ? :?
     "  ";FMS$:GOTO 9750
9720 ? FMS$;
9730 ?
9740 GOTO 9700
9750 ? :? "  HIT RETURN FOR MENU"
9760 GET #KEY,X:IF X<>155 THEN 97
     70
9770 TRAP 40000:CLOSE #DSK:CLOSE
     #KEY:CLOSE #SCR
9780 POKE 752,0:RETURN
```

Music & Sound

Harmony *by Tom Hudson*

This short routine can be used to create three-part harmony using DATA statements. Each DATA statement contains information to make one three-part chord of any duration. The first three numbers in each DATA statement are the pitch values for each of the three notes which sound simultaneously (See the BASIC REFERENCE MANUAL, page 58, for pitch value equivalents.) The fourth number in the DATA statement sets the duration of play. The sample DATA statements included in the subroutine play eight notes. The last DATA statement (ninth) stops the music, and a zero duration value tells the subroutine to RETURN to the main program. You might use a routine like this to compose special songs for your friends or to add music to a title or menu screen. You can make your tune as long as your DATA statements last. Have fun!

A sample calling program for this routine might be . . .

```
10 FOR REPEAT=1 TO 3
20 RESTORE 1100: GOSUB 1000
30 NEXT REPEAT
40 END
```

```
1000 READ A,B,C,D
1010 SOUND 0,A,10,8:SOUND 1,B,10,1
     0:SOUND 2,C,10,10
1020 IF D=0 THEN RETURN
1040 FOR TIME=1 TO D*8:NEXT TIME
```

```
1050 GOTO 1000
1100 DATA 60,81,96,15
1110 DATA 60,81,108,25
1120 DATA 60,81,96,15
1130 DATA 60,72,91,40
1140 DATA 60,81,96,15
1150 DATA 60,72,91,25
1160 DATA 60,64,81,25
1170 DATA 60,60,96,50
1180 DATA 0,0,0,0
```

GTIA Graphics Power

Farrah

by Bill Carris

The 'Farrah' program listing is presented here because it illustrates a very basic computer graphics technique—that of building images from elements. Using simple PLOT, DRAWTO, and COLOR statements, you can make complex drawings on your Atari Home Computer, element by element.

'Farrah' illustrates the step-by-step method of building a detailed screen image (in this case with fine shadows and highlights in Graphics Mode 9 using the GTIA chip's power).

Author Bill Carris began with a picture in his mind, then actually constructed it on the screen in pieces. In a way, constructing graphic pieces is much like connecting the dots.

If you "connect the dots" of this program by typing it into your computer, an amazing image will emerge in its constituent parts and you'll see why Bill Carris titled his design "Farrah"!

```
 10 REM ** draw first element **
 20 GRAPHICS 9:SETCOLOR 4,0,0
 30 FOR Z=80 TO 96:COLOR Z:PLOT 2
    ,Z:DRAWTO 68,Z:NEXT Z
 40 FOR Z=112 TO 96 STEP -1:READ
    X:COLOR X:PLOT 2,Z:DRAWTO 66,
    Z:NEXT Z
 50 REM ** draw second element **
 60 RESTORE 710
 70 FOR N=64 TO 95:READ KO:COLOR
    KO:PLOT 31,N:DRAWTO 32,N:NEXT
     N
 80 RESTORE 710
 90 FOR BOT=128 TO 96 STEP -1:REA
    D BK:COLOR BK:PLOT 31,BOT:DRA
    WTO 32,BOT
100 NEXT BOT
110 REM ** draw third element **
120 RESTORE 730
130 COLOR 0:FOR XA=33 TO 49:READ
    YA,YB
140 COLOR 2:PLOT XA,YA:DRAWTO XA,
    YB
150 COLOR 3:PLOT XA,YA+2:DRAWTO X
    A,YB-2
160 COLOR 4:PLOT XA,YA+6:DRAWTO X
    A,YB-6
170 COLOR 7:PLOT XA,YA+9:DRAWTO X
    A,YB-9
180 COLOR 10:PLOT XA,YA+12:DRAWTO
     XA,YB-12
190 COLOR 12:PLOT XA,YA+15:DRAWTO
     XA,YB-15
200 COLOR 13:PLOT XA,YA+18:DRAWTO
     XA,YB-18
210 COLOR 14:PLOT XA,YA+21:DRAWTO
     XA,YB-21
220 NEXT XA
230 REM ** draw fourth element **
240 COLOR 2:FOR XB=38 TO 40
250 IF XB=39 THEN PLOT XB,47:DRAW
    TO XB,64
260 PLOT XB,48:DRAWTO XB,64:NEXT
    XB
270 COLOR 6:FOR XC=42 TO 44
280 IF XC=43 THEN PLOT XC,47:DRAW
    TO XB,64
290 PLOT XC,48:DRAWTO XC,64:NEXT
    XC
300 COLOR 10:PLOT 41,28:DRAWTO 41
    ,47
310 COLOR 3:PLOT 40,28:DRAWTO 40,
    46
320 COLOR 6:PLOT 42,28:DRAWTO 42,
    46
330 COLOR 2:PLOT 41,65:DRAWTO 41,
    60
340 FOR Z=21 TO 28:COLOR Z+12:PLO
    T 28,Z:DRAWTO 54,Z
350 IF Z>27 THEN COLOR 2
360 IF Z>22 THEN PLOT 27,Z:DRAWTO
     55,Z
370 NEXT Z
380 REM ** draw fifth element **
390 COLOR 2:PLOT 71,100:DRAWTO 71
    ,135
400 COLOR 4:PLOT 70,90:DRAWTO 70,
    133
410 COLOR 6:PLOT 69,86:DRAWTO 69,
    132
420 COLOR 8:PLOT 68,87:DRAWTO 68,
    132
430 COLOR 12:PLOT 67,88:DRAWTO 67
    ,132
440 PLOT 66,100:DRAWTO 66,132
450 PLOT 65,101:DRAWTO 65,132
460 COLOR 10:PLOT 64,102:DRAWTO 6
    4,133
470 COLOR 8:PLOT 63,104:DRAWTO 63
    ,134
480 COLOR 5:PLOT 62,105:DRAWTO 62
    ,135
490 COLOR 14
500 PLOT 62,135:PLOT 63,136:PLOT
    64,137:PLOT 65,138
510 PLOT 66,138:PLOT 67,138:PLOT
    68,137
520 PLOT 69,136:PLOT 70,135
```

```
530 REM ** add sound & artifacts
    **
540 FLIPCOL=0
550 COUNT=0:SETCOLOR 4,FLIPCOL,0
560 FOR DR=137 TO 142 STEP 2:COLO
    R 15:PLOT 63,DR:DRAWTO 63,DR+
    4:NEXT DR
570 FOR DR=137 TO 186 STEP 2:COLO
    R 15:PLOT 63,DR:DRAWTO 63,DR+
    4
580 COLOR 0:PLOT 63,DR:DRAWTO 63,
    DR+2
590 NEXT DR
600 FOR DR=190 TO 188 STEP -1:PLO
    T 50,DR:DRAWTO 70,DR:NEXT DR
610 FOR DN=77 TO 10 STEP -7:SOUND
     0,DN,10,DN:NEXT DN:SOUND 0,0
    ,0,0
620 X=INT(RND(0)*20):Y=INT(RND(1)
    *20)
630 IF X<>Y THEN GOTO 620
640 COUNT=COUNT+1
650 IF COUNT<=5 THEN GOTO 560
660 IF FLIPCOL=15 THEN GOTO 540
670 IF FLIPCOL=0 THEN FLIPCOL=15:
    GOTO 550
680 REM ** data statements **
690 DATA 0,1,2,3,4,5,6,7,8,9,10,
    11,12,13,14,15,15
700 DATA 80,81,82,83,84,85,86,87,
    88,89,90,91,92,93,94
710 DATA 1,1,2,2,2,3,3,3,4,4,5,5,
    6,6,7,7,8,8,9,9,10,10,11,11
720 DATA 12,12,13,13,13,13,13,13
730 DATA 77,115,77,115,74,116,70,
    119,67,123,65,124
740 DATA 64,125,64,126,63,127,63,
    126,64,126,64,125,65,124,67,1
    23
750 DATA 70,119,74,116,77,115,77,
    115
```

Contributors to Bits and Pieces

Bill Carris lives in the fast lane, writing programs the way T.S. Eliot wrote poetry—all in a night's sitting. Bill says he's your BASIC kind of guy.

Tom Hudson is into both computers and music and has trained his Atari Computer to sing like a nightingale. He is able to make simple things be indeed simple.

Kent Smith has been programming since he was fourteen. Now 19, he knows the Atari Home Computer inside and out! See his article on PEEKing and POKEing in this issue!

Tay Vaughan says ". . . Creating nifty routines like 'DATE' or 'DISKLOOK' turns me on like sculpting or painting probably turns on a classic artist." Tay has a Coast Guard captain's license for passenger vessels, flies airplanes, plays the cello, and programs his Atari Computer. ■

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 your 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 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. Attach a printed listing of your program or subroutine (or a listing on cassette or diskette) 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!

By submitting a program you consent to its publication and use in the ATARI CONNECTION and elsewhere. Media and manuscripts which are submitted for review will be returned if you include a self-addressed, stamped envelope. If your program or subroutine is published in ATARI CONNECTION, you will receive a free $20 gift certificate good for purchases from the Atari Program Exchange (APX)!

COMPUTER COMFORT

A Guide to Using Your Atari BASIC Text Editor

by Jane Sokolow

WHEN YOU TYPE in the lines of a computer program, the computer obeys each numbered line as a separate command. If it cannot understand any of the instructions given in the program line, then the computer will respond with an *ERROR Message*. Nine times out of ten the error will be a *syntax* mistake—you failed to correctly write a specific command. And in the case of most beginners, copying a program from a magazine, the error is a simple *typo*. The computer can't tell the difference.

For a program line to be seriously considered by the computer, it must follow the format that was built into the computer and the BASIC cartridge. The general format for a line is: a line number, followed by a command (like PRINT, GOTO, IF . . . THEN, etc.) or an indicator of some operation (some mathematical formula), followed by "variables"

(meaning items to be acted on by these commands or operations).

The maximum length a line can take is three lines (on your screen). A buzz will warn you if you are about to exceed this length, giving you seven more characters to go. Anything more than that will not be accepted into memory. Some magazines have been known to make program lines too long. Usually this is due to stringing too many commands together, such as:

```
40 GRAPHICS 7
50 DIM A$(30)
60 ON X GOTO 200,300,400,500
```

shortened to:

```
40 GR.7:DIM A$(30):ON X GOTO
200,300,400,500
```

If such a line goes on for more than three lines and gets a beep, remove the colon and the last command and make a new line of it, such as:

```
40 GR.7:DIM A$(30)
45 ON X GOTO 200,300,400,500
```

Common ERROR Messages:

There are other kinds of errors which the computer will not catch instantly. Some of these are either programming errors (errors in logic), or mistypings which create logic errors. Many people have gotten the errors discussed below when typing in the *Basic Reference Manual* programs, but be advised that these are only a few of many possible ones. Errors come up after you have typed in the program, and you then type RUN and press RETURN. Sometimes the message comes up immediately, and sometimes it runs for awhile and then stops and gives you an error. Let's take a simple music program such as *Computer Blues* as an example.

ERROR 8: You type the *Computer Blues* in, type RUN, hit RETURN, and then the program begins with a question on the screen: BASS TEMPO (1=FAST)? This is intended to instruct you to put in some number; the number 1 would give a fast tempo, while a higher number would be a slower one. We admit that this is not too clear: Do you type in a number and the word FAST? But there you are. So you misunderstand, and type in 1=FAST and get an ERROR 8. This means you have put in letters where only numbers should go (meaning the letters of the word FAST). Simply restart the program and put in a number.

ERROR 6: The program begins working, and pretty soon you get ERROR 6 (out-of-data error) on line 98. You look at the listing in the book to see that line 98 has a READ command in it. This means the computer looks to whatever lines start with the word DATA and begins to pick up the numbers. Lines 1000 to 1050 are DATA statements. Line 98 is the last in a series of statements picking up DATA, and if there are not exactly the right amount of numbers in those DATA statements, the program will run out. Check to see that line 1000 has 12 numbers in it (remember . . . no RETURNs in the middle of the lines), line 1010 has 3 numbers, line 1020 has 16 numbers, and lines 1030, 1040 and 1050 have a total of 21 numbers altogether. If not, then that's the reason for the error.

Remember that the line referenced in the error message may not be in error. Some line in RELATION to that line must be in error, however.

ERROR 9: This happens when a variable occurs in a program and there is no DIM statement before it. DIM stands for "dimension" and sets aside a space for the variable. This error will also happen if the variable in the DIM statement is not spelled the same somewhere else. For example, if line 26 DIM JAM(3,7) were entered that way, and line 98 read READ A:JAN(X , Y) . . . In this case, you would get ERROR 9.

ERROR 17: This is the nerve-wracking one, as the description mentions possible hardware failure. 999 times out of 1000 the reasons for this error is that an ERROR message was not deleted from the listing (the user didn't hit RETURN after erasing it), and the computer interprets this as garbage.

There is another error occasionally encountered in this program: everything works, but nothing happens. Try turning up the volume on your TV. This program makes music, remember?

Brackets:

If you see brackets (not parentheses) within a program, it indicates that some special combination of keys is being used for typing in control characters that cannot be shown in print (not because they are "X-rated," but because it is outside the capacity of the printer). Refer to the heading of the article; in the case of many magazines, there is a set of instructions somewhere telling you what the convention is. A good example is the *Special Listing Instructions* that appear in the *Computer Animation Walking Man* in this issue of ATARI CONNECTION.

With any program, if you run it and nothing happens, hit the BREAK key during a RUN, and it will tell you on which line the program stopped. This will help you begin to unravel the mystery of its failure. Happy sleuthing! ■

Program Typos in Current Atari Home Computer Publications.

Atari 800 Owner's Guide

Basic Reference Manual

All errors and additions are gathered in the Product Update (aka BASIC Update), which either came with your *Basic Reference Manual*, or can be ordered free from Atari. Program (H Section) corrections are:

Text Modes Character Print, H-7:

```
160 IF K=125 OR K=155 THEN 180
```

Light Show, H-8:

```
50 PLOT 0,0:DRAWTO 159,DR
```

Hex-Decimal Conversion Program, H-18:

```
200 GOTO 30
390 GOTO 30
```

Seagull Over the Ocean, H-11:

Line 20 shows four spaces between quotation marks. At the top of the page it is explained that the printer could not make the graphic symbols; the line should actually be written with four symbols within the quotes, and be executed by holding down the CTRL key while typing a G, an F, an R, and an R (without the commas between). However, it looks as if there is a space between the quotes and the first graphics symbol, as well as a space between the last symbol and the closing quotes. Since the program was designed to shift back and forth between the first two symbols (upraised wings) and the last two (outstretched wings), it will not do this correctly if there are spaces in the way. So eliminate the spaces and the seagull will fly.

DOS II Manual

Page 65 has a program so garbled it is useless. Contact Atari Technical Support for revised version. (See phone numbers following.)

There are several typos and errors in the program listings. Call Atari Customer Service for help. This book has been replaced in later Atari Home Computer packaging with Bill Carris' little jewel, *Inside Atari BASIC*, which is a wonderful introductory manual. Or, it can be obtained in software stores, or ordered toll free from Reston Publishing Company, Reston, Virginia (800) 336-0338.

We have a feeling that some of you out there have your own examples and questions you would be willing to share with our readers. Perhaps this can even become an ongoing forum providing "computer comfort" for new Atari users. Send your comments and suggestions to *Computer Comfort*, c/o ATARI CONNECTION. For those with questions on the above publications you can call Atari Customer Service for errata sheets, etc. In California call (800) 672-1404; the Continental U.S.A. (800) 538-8543. ■

Jane Sokolow is a Product Specialist in Customer Relations at Atari. She has previously been a linguist, a newspaper editor, and a holistic health counsellor, and is especially interested in the interaction of computers and people.

PEEKS AND POKES

Commonly Used and Helpful Memory Locations

by Kent Smith and Lee Sherman

YOUR ATARI 400 or Atari 800 Computer is very much like a human brain when it comes to memory. . . . The human brain contains an infinite number of memory locations with different areas of the brain controlling different body functions. Your Atari Home Computer operates in much the same fashion. Altering different memory locations with PEEKs and POKEs will perform various functions within your programs.

What exactly are these PEEKs and POKEs you keep reading about? The PEEK statement allows a programmer to see what numeric value is stored in a particular memory location. The POKE statement allows that same user to change the value in that memory location. So what does this mean to the Atari Computer programmer? More *power* in your programs.

The following Memory Locations are a sampling of the most common addresses to PEEK and POKE and can be extremely useful in your ATARI BASIC programs:

18, 19, 20: Real-Time Clock Location

Three bytes of memory are used to keep track of time by utilizing the television frame counter. Every 1/60th of a second a complete new image is displayed on your TV screen. Location 20, the least-significant byte of this three-byte counter, is incremented every 1/60 of a second. Location 19 is incremented after Location 20 reaches 255. Location 18 is incremented after Location 19 reaches 255.

You can see that from the time you power up until there have been 16.7 million television frames displayed, this clock will keep counting. When it has counted for about 17.67 hours the Real-Time Clock is automatically reset to zero. A program listing which allows you to create a regular hours-minutes-seconds clock is found in the "Bits and Pieces" section of this issue of ATARI CONNECTION.

77: Attract Mode

Each time a key is pressed, Location 77 is reset to zero. If no keys are pressed before the Location counts to 128, the attract mode is started. POKEing Location 77 with a zero will stop the attract mode for approximately seven minutes. *Use caution when turning off the attract mode* for long periods, as you run the risk of permanently etching an image onto your screen. The attract mode was built into the Operating System by Atari engineers who understand that people may forget to turn off their TVs when they go away for the weekend. While it does take quite a while to incur this sort of damage, be careful!

```
10 POKE 77,0:REM TURN OFF ATTRACT MODE
```

82,83: Left and Right Margins

Initially, Location 82 contains a 2 and Location 83 contains a 39. These are the "default" margins pre-set by the Operating System. POKEing a zero into Location 82 will set the left margin to column 0 (at the far left-hand side). Or, if you want a huge left margin, POKEing 82,20 will set the left-hand margin to column 20. The first column available on the screen is 0, the last column is 39, thus giving you 40 characters per line. The left screen margin is preset to 2 because some television sets overscan, and the first columns might get tucked away off the left side of the screen.

```
10 POKE 82,0:REM RESET LEFT MARGIN
```

84,85: Row and Column for Cursor

In GRAPHICS Mode 0, Location 84 contains the current row for the cursor. Location 85 contains the current column for the cursor. Changing the values in these Locations moves the cursor to the row and column POKEd. You may not exceed normal screen margins.

```
10 POKE 84,10:POKE 85,20:PRINT "ATARI":REM
MOVING CURSOR
```

752: Cursor On/Off

Location 752 contains a 0 when the cursor is on and a 1 when the cursor is off.

```
10 POKE 752,1:REM CURSOR TURNED OFF
```

764: Last Internal Key Code

When a key is pressed, the value of the key pressed is read by the Pokey chip and placed into the keycode Location 764. If no key has been pressed, a value of 255 is the default value at this Location. See the "Technical Reference Notes" (ATARI Catalog Number CO16555 Rev. A), Page 185, for the Pokey keyboard codes or PEEK into Location 764 with a one-line program like:

```
10 PRINT PEEK(764):
GOTO 10.
```

You can use this Location in the following way:

```
10 POKE 764,255:IF PEEK(764)=255 THEN 10
20 IF PEEK (764)=63 THEN PRINT "ATARI"
```

53279: Console Keys

All three console keys are worked by PEEKing and POKE-

ing Location 53279. Each key represents one binary bit of one byte. When no keys are pressed, the Location contains a value of 7. Each key will reset its bit to 0 when pressed. Pressing the OPTION key will return a value of 3. The SELECT key returns a value of 5 and the START key will return a value of 6. Varying combinations of the three produce different values. The following is a table of values found in Location 53279 when the console keys (or combinations of the keys) are pressed:

START	6
SELECT	5
OPTION	3
START + SELECT	4
START + OPTION	2
SELECT + OPTION	1
START + SELECT + OPTION	0
NO KEY PRESSED	7

16 AND 53774: DISABLE BREAK KEY

To prevent the BREAK key from stopping a program while it is running, POKE the value 64 into Locations 16 and 53774. Changing graphic modes or OPENing some IOCBs resets the break key. Therefore it must be re-disabled after GRAPHICS and some OPEN commands.

```
10 POKE 16,64:POKE 53774,64:REM DISABLE
BREAK KEY
```

580: ALTER SYSTEM RESET

A value of 1 in this Location will reset the cold start flag. When the SYSTEM RESET key is then pressed, the system acts as though it has just been turned on and will purge any resident programs in RAM. Used in conjunction with the break key disable and the AUTORUN.SYS feature of DOS II, resetting cold start is used by some owners to "protect" their programs from prying eyes by making it difficult to obtain a LISTing without LOADing the program directly from disk. Commanding NEW before ending the program "erases" the program from memory before displaying READY. A cold start (when you press SYSTEM RESET) will automatically load DOS and the program will be again AUTORUN. So the operator is forced into a "loop" which will only end when the programmer allows it.

```
10 POKE 580,1:REM SET FLAG FOR POWER UP
```

65: NOISY I/O FLAG

To prevent data pulses from being heard over your television speaker during Input/ Output operations from cassette or disk, POKE a value of 0 into Location 65. A value of 1 will turn the I/O flag back on.

```
10 POKE 65,0:REM TURN OFF DATA PULSES
```

1913: VERIFY AFTER WRITE

Each time the computer writes information to the disk drive, it also reads back the data to verify a "good write." POKEing a value of 80 into Location 1913 will turn off the verify operation, which will speed up disk writing operations. Remember, however, if there is a "bad write" (rare!) you may foul up your data.

```
10 POKE 1913,80:REM TURN OFF DISK DRIVE
WRITE VERIFY
```

3118: DISK FILE OPERATIONS

A value of 0 in Location 3118 will cause DOS to operate only on the first of two or more files with the same filename. By POKEing 0 in Location 3118 you can use DOS Option E ("Rename File") to change the first file name. If two files accidently have the same filename on your disk, you can have trouble trying to access them. By POKEing 3118,0 and calling DOS, the first file could be renamed, leaving the second file with the original file name, and you would be out of your fix.

```
10 POKE 3118,0:REM SINGLE FILE OPERATIONS
```

1801: NUMBER OF FILES

This Location will allow the user to open more than 3 files on disk at one time. Up to 8 files can be opened for use within a program by POKEing the number of open files into Location 1801.

```
10 POKE 1801,5:REM ALLOW 5 OPEN FILES
```

1802: NUMBER OF DISK DRIVES

The standard DOS Master Diskette will only allow two drives to connect to the system. To use more than two disk drives, POKE a value of 7 into this Location to use three drives, and 15 to use four drives.

```
10 POKE 1802,15:REM USE FOUR DRIVES IN
SYSTEM
```

You can rewrite your DOS files and avoid POKEing to Location 1802 every time you power up with more than two drives. Poke the correct value (a 7 or 15 for a three- or four-drive system respectively) into Location 1802, call DOS, and rewrite the DOS file using Option H!

54018: 410 CASSETTE MOTOR CONTROL

The 410's motor can be turned on by POKEing a value of 52 in this Location. To turn off the motor, POKE a 60 into this Location. This can be useful when recording audio data onto the left track of your cassette tapes in conjunction with program data on the right track (as is done in "Invitation to Programming" and "States and Capitals"). For additional information on putting an audio track on your program tapes, refer to DE RE ATARI, Appendix C.

```
10 POKE 54018,52:REM TURN 410 MOTOR ON
20 POKE 54018,60:REM TURN 410 MOTOR OFF
```

838 AND 839: PROGRAM RUN TO PRINTER

POKEing the value 166 into Location 838 and the value 238 into Location 839 will send anything the computer normally PRINTS on the screen to the printer instead. However, anything typed from the keyboard will also go to the screen. To return to normal operation, POKE Location 838 with a 163 and Location 839 with 246.

```
10 POKE 838,166:POKE 839,238:REM SEND RUN TO
PRINTER
20 PRINT "ATARI":REM PRINT 'ATARI' ON PRINTER
30 POKE 838,163:POKE 839,246:REM SEND RUN
BACK TO SCREEN
```

For additional ATARI Computer memory locations, refer to the "Technical Reference Notes" (Atari Catalog Number CO16555, Rev. A). Publications such as "Master Memory Map" by Educational Software or "Mapping the Atari" by Compute! Books may also be helpful. ■

Kent Smith and Lee Sherman are Product Specialists for Atari Customer Service.

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 Grace" 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.
- Entries into the $5000 talking or singing game contest can be written using the **VOICE BOX II**–send for contest information.
- 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.

All About Atari LOGO

The Computer Language That Speaks Your Language

by Carlos Greth

IF YOU'RE FRUSTRATED because your computer doesn't speak your language, don't despair. This summer, you can experience the power of Atari LOGO, a computer language that makes programming possible for just about anyone.

Originally designed for younger students, LOGO is a remarkable, high-level programming language that uses simple English commands to produce sophisticated programs that both young beginners and advanced programmers will appreciate. Since you use commonplace words instead of complex, technical programming statements, you can begin controlling and changing text and shapes on the TV screen the first time you use it.

What sets LOGO apart from other programming languages is that it is both easy to learn *and* easy to use. The simple Atari LOGO vocabulary can be expanded to suit each individual's needs. For example, you can choose your own words to name a special program you have written, such as a program, let's say, that draws a car. You can simply call that program, "Car." You can then enter the word "Car" into a second program that draws a street. LOGO will remember the name "Car" and draw a car on the street. Of course this is but a simple illustration. With Atari LOGO, you can define far more complex program functions—in effect, create your own program "building blocks" or modules. You're limited only by your desire and imagination.

The LOGO language is the result of over 10 years of research in artificial intelligence and education led by MIT Professor Seymour Papert. Atari LOGO is a special version developed by Logo Computer Systems, Inc., of Montreal, Canada.

One of Dr. Papert's primary goals in developing LOGO was to create a computer environment or "culture" where learning could take place naturally. In his landmark book, *Mindstorms*, Papert writes: "It is possible to design computers so that learning to communicate with them can be natural, more like learning French by living in France than like trying to learn it through foreign-language instruction in the classroom."

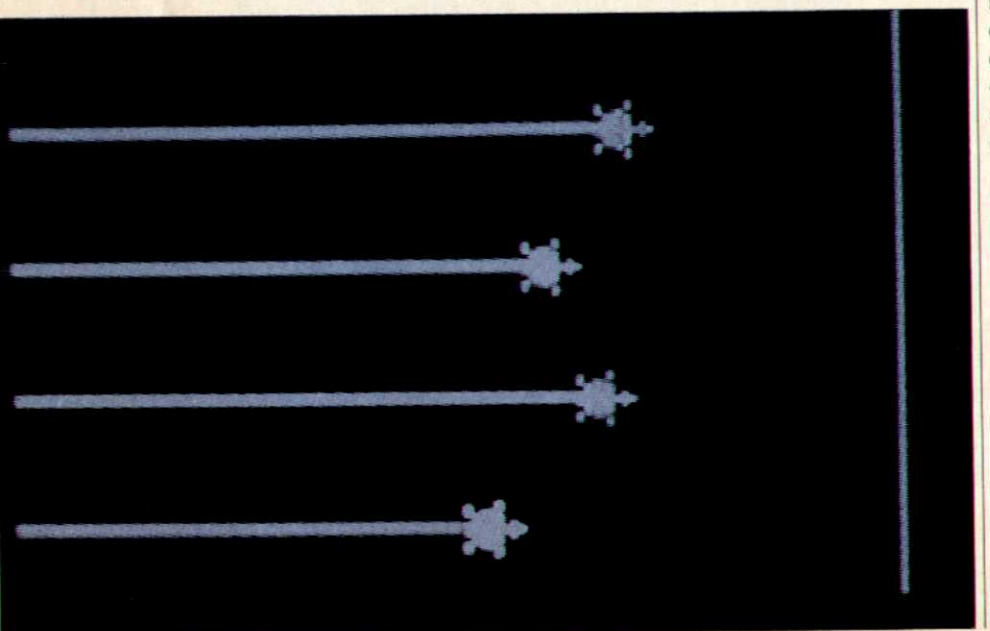
With Atari LOGO, users of all ages evolve from passive into active learners. They set up problems to solve, and learn to experiment with different solutions, while trying to make appropriate choices. The computer becomes a learning tool for exploration in the realms of geometry, math, science, and language. Like explorers in a new world, users begin anew to name and organize their experiences—the essence of learning.

The Turtle graphics featured in Atari LOGO offer a direct and creative approach to what Swiss child psychologist Jean Piaget calls the "active discovery of reality." Learners from grade school to college have taught themselves about scientific and mathematical concepts while manipulating LOGO graphic symbols. The "turtles" draw paths at various speeds and directions as they move across the TV screen. The computer-controlled turtles are made even more enticing by the graphics, sound, and color capabilities of Atari Home Computers.

One educator who worked with students who used LOGO asked them what they felt they had learned. The students replied that LOGO had helped them become more comfortable with math and problem solving and had gained valuable insights into language by naming their own programming instructions. Students also learned problem-solving strategies, and developed a positive attitude about the trial-and-error process by debugging their own programs.

The new Atari LOGO will be a convenient program cartridge that will work on any Atari Home Computer with as little as 16K RAM. The package will include a handy reference card, and two excellent books: *Introduction to Programming Through Turtle Graphics* and *Atari LOGO Reference Manual.*

The Atari LOGO programming promises to be available by the end of this summer. Also, the Fall issue of ATARI CONNECTION will feature an in-depth interview with LOGO creator, Seymour Papert. ■



Carlos Greth is a writer in the Atari Home Computer Division.

Programs to Star Gaze By

Chart the Heavenly Bodies with Celestial BASIC

Celestial BASIC

Astronomy on Your Computer
by Eric Burgess
Sybex, 300 pages, $14.95.

by Steve Englehart

ONE OF THE THINGS A NO-LONGER-brand-new programmer learns is that, after devouring one or two "Learning BASIC" and one or two "Learning Graphics" books, he's pretty much on his own. There just isn't much literature on the thousands of other directions a computer can go. And, of course, there's a reason for that. The computer revolution is happening even as we speak, so the potential audience for special programs is small in the aggregate, and miniscule when split into a thousand different interests. Nevertheless, if it's *your* interest that you can't find programs for, you can't help but wish they'd made an exception for you. Well, if you're interested in astronomy, they have. And if you're not . . . you just might get to be.

Celestial BASIC is an excellent book in all respects—well-designed, well-made, and packed with the programs an amateur or serious astronomy buff would want. Eric Burgess starts at the beginning, with the calculation of time, then works with features of the moon, the planets, and finally the solar system as a whole. The opening program creates a perpetual calendar, which can provide the day of the week for any date, display a calendar for any month of any year, or list the holidays in any year. This is pretty "down to earth"—something anyone could use. But then Burgess provides a program to change local time to sidereal time, the time measured relative to the fixed stars and used for astronomical calculations. This is very useful for all further work with the planets. Ah, but those "fixed stars" actually do move, albeit extremely slowly, and there's a program to determine those movements, as well as one specifically for Polaris, the North Star.

In other words, whatever your level of interest in time, Burgess is ready for you. But more than that, each new program is backed with a text piece explaining the importance of the concept involved, the history of its development, and the theories utilized in the program's mathematics, all in the style of a scientist who wants non-scientists to understand. Therefore, whenever you say to yourself, "This is enough; I don't need to know any more," Burgess will show you why the next step is worthwhile and enjoyable, too.

That's particularly true with the members of our solar system. Having shown the reader how to produce the positions of the hardly moving stars, the book turns next to the position of the quickly moving moon. Here the accuracy is not so great, since the number and complexity of factors involved surpasses the capacity of a home computer, but the readouts are more than sufficient for amateur observation. Coming off that program, Burgess offers other programs to determine lunar eclipses and the phase of the moon at any time.

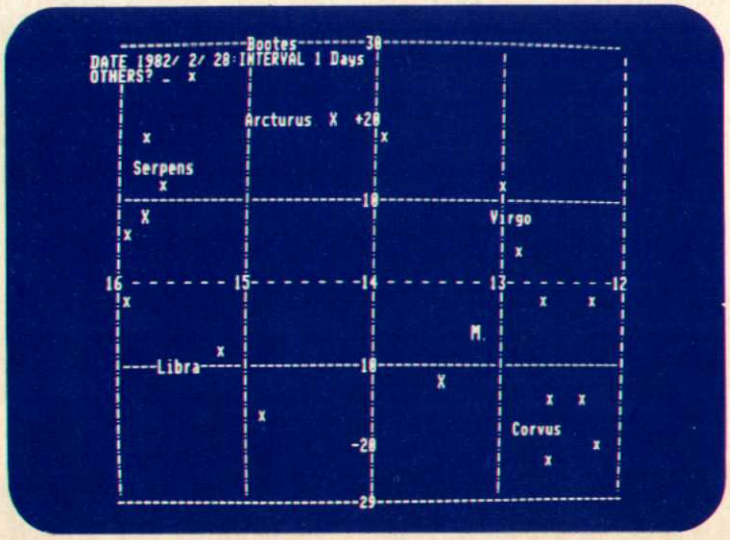
Then he moves to the planets, providing similar information for the nearby bodies and such esoteric tidbits as elongations, angular diameters, and distances (in the cases of Venus and Mercury). Next, all the information is brought together for a program which plots the visible planets, sun, moon, and stars for any date, time, and location, and a program which plots the planets, sun, and moon against the constellations.

Finally, Burgess throws in programs to chart the positions of Jupiter's four largest moons, the dates of meteor showers, and even the best method of photographing celestial objects.

As far as I know, there is no other book like this available. But even if there were, I can't imagine how it could surpass *Celestial BASIC*, either in thoroughness or readability. As noted, this is not a book for everyone, but those who pursue other disciplines should pray that they're as well-served when their time comes.

One final note: the programs in this book are written for Apple BASIC, but the information necessary to convert them to Atari BASIC (a minor procedure) is provided in the text. ■

Sample star-charting program screen from Celestial BASIC



Steve Englehart is Senior Writer for the Atari Home Computer Division

Users' Group News and Networks

Fear of Programming and How to Overcome It

by Earl Rice

PROGRAMMING—THE VERY WORD USED to send waves of conflict through me. It seemed like such a great idea, but it looked so hard! Well, I finally *did* learn to program, and as Chris Crawford says on the Atari Video Visits tape, "You can, too!" It *was* a little difficult at first, because way back then there wasn't anybody to help me out when I got stuck. I didn't know about users' groups, and neither did the people who sold me my first computer. That was a long time before the Atari 800 came along.

Today there are plenty of users' groups (about 300 of them), and here at the Atari Home Computer Division we work pretty hard to be sure they're successful and healthy. That's because we know, sooner or later, you're going to want to learn a lot more about your computer—a users' group is a great place to learn.

Eventually, you are going to want to write a program for your computer. Maybe to solve a science class problem, or keep track of the value of your collectibles. Maybe you want to know what level of programming skill you need to write a game program. Who can better help you than someone who's had a few years' experience in programming, or a science student, or collector, or skilled computer game programmer? You'll meet other Atari Computer owners with interests like yours at a users' group meeting.

Many groups have programming classes to help you get started. Meeting with other people who program helps stimulate ideas and speed your learning process. Soon you can gain the level of mastery of your computer that you want.

All at your own pace. Among your friends.

Of course you don't have to become a programmer to get benefits from joining a users' group. Most groups review products and software and can help you pick the programs you need to get things done, without any pressure. But the theme of this issue is programming, so that's what we're talking about.

To help groups teach programming, the Atari Home Computer Division has a videotape library series called Video Visits. It helps the programmers in a group present concepts so everyone can understand how their computer works. We also provide groups with a library package to help answer the really technical questions you might come up with. It pays off. Groups in Oregon, Michigan, Oklahoma, Texas, New York and elsewhere give great classes to get you started. As a result, many of the software selections available through the Atari Program Exchange were written by users' group people. One program, called *I'm Different*, was written by a woman who had no programming experience at all.

If you're beginning to program, there's a lot of help available through the users' group community. Many groups have excellent collections of programming aids and program examples to share, which makes the task of mastering your machine much easier.

Fortunately, that's easily done. Call Atari Customer Service for the name of the group nearest you. If you're interested in learning a lot about users' groups and which ones have bulletin boards, an excellent resource book will be available in mid-June. The book is titled *Free Software for Your Atari Computer* by David and Dorothy Heller (published by Enrich/OHAUS) and it lists many groups with bulletin boards and public domain software libraries. It also lets you know which groups specialize in education, programming support, and the like. For $7.95 it's a good deal and can serve to open up a much wider world of computing for you.

The first section of *Free Software* introduces you to the world of Atari Users' Groups. Many of these groups have special interests. The detailed profiles of selected groups introduce you to users' groups that specialize in Education, Amateur Radio, Engineering, Computer Art, Computer Hardware, Science, and Business. All the users' groups have been listed and you'll learn how to become a member of many of these groups without leaving your home! Or, if you like, the list includes the location and address of your nearest group, and the right person to contact, so you can attend their next meeting to get software and support from other Atari owners.

The "Free Software by Phone" section shows you how to get the programs available in the public domain. This detailed section begins by explaining the terms used in computer telecommunications, gives you a comprehensive equipment buyers guide, and includes program lists that turn your computer into a telecommunications terminal. There's even an interview with a genuine Electronic Bulletin Board operator (called a SYSOP) who tells you how to start your own "base" station, and how to get the software you'll need to begin operation. You'll also find a starter BBS phone directory of over 500 phone numbers that put you in contact with other Atari users and a wide variety of free services.

Free Software is particularly valuable to parents and educators who use the Atari Computer in a learning environment. The comprehensive section on "Educational Software and Resources" shows you where to find the support you need to expand the use of your Atari Computer at home and in the classroom. The book also includes information on where to get educational software, how to network with other parents and teachers to solve teaching problems, and ways to share ideas and programs.

The point is, what makes computers so wonderful is your ability to program them. You can make your computer do so many things. Whether you want to program "for fun," for some practical application, or to seek fame and fortune with your game idea, learning to program is what makes it possible. And learning in the company of friends makes it so much easier and enjoyable. That's where your friendly neighborhood users' group comes in. If you can't find one to join, we'll help you start one! ■

Earl Rice is the Manager of Atari Users' Group Support

Networking with AtariWriter

Transmit Manuscripts and Reports by Phone

by Steven Young

THE PRESIDENT OF THE company has asked you and a colleague for a yearly sales report—by the end of the week. You have the facts and figures; she can write polished prose that the president will actually read. But you're up to your chin in an audit at the Kansas City office, and she's chained to her desk in New York.

As chairperson of your Atari Users' Group newsletter committee, you spend most of your evenings driving manuscripts around three counties, from writers to editors and back again, and again, and again . . . The gas bills are killing you, and you're developing a shady reputation in your neighborhood as a sort of literary bagman.

The day after your deadline, your editor across the country calls and asks when he's going to see your copy—"If ever," he snidely adds. You say you put it in the mail a week ago, as you did. He says he's heard that before, as he has, and hangs up.

You've got problems.

But if you and your partners use AtariWriter, the new word processor from Atari, you may already have a solution—just as near as your phone, because AtariWriter talks to itself. If your computer system includes a disk drive and an *acoustic* modem, you can send AtariWriter text files (or any other files that AtariWriter can read) over the telephone to anyone who also has an AtariWriter program cartridge, a disk drive, and an acoustic modem.

Here's the procedure for telecommunicating a file from a "sender" to a "receiver" using Atari 830 Acoustic Modems and AtariWriter:

1. The sender and receiver must both have their interface modules turned on, and both must use their DOS II Master Diskettes when loading AtariWriter into their computers. They should also be sure to have equivalent amounts of RAM in their computers—a long file created on a computer with 48K RAM will overflow a computer with only 16K or 32K RAM.

2. The sender and receiver reach out and—well, establish a telephone connection.

3. The sender loads the file to be sent into his computer's memory.

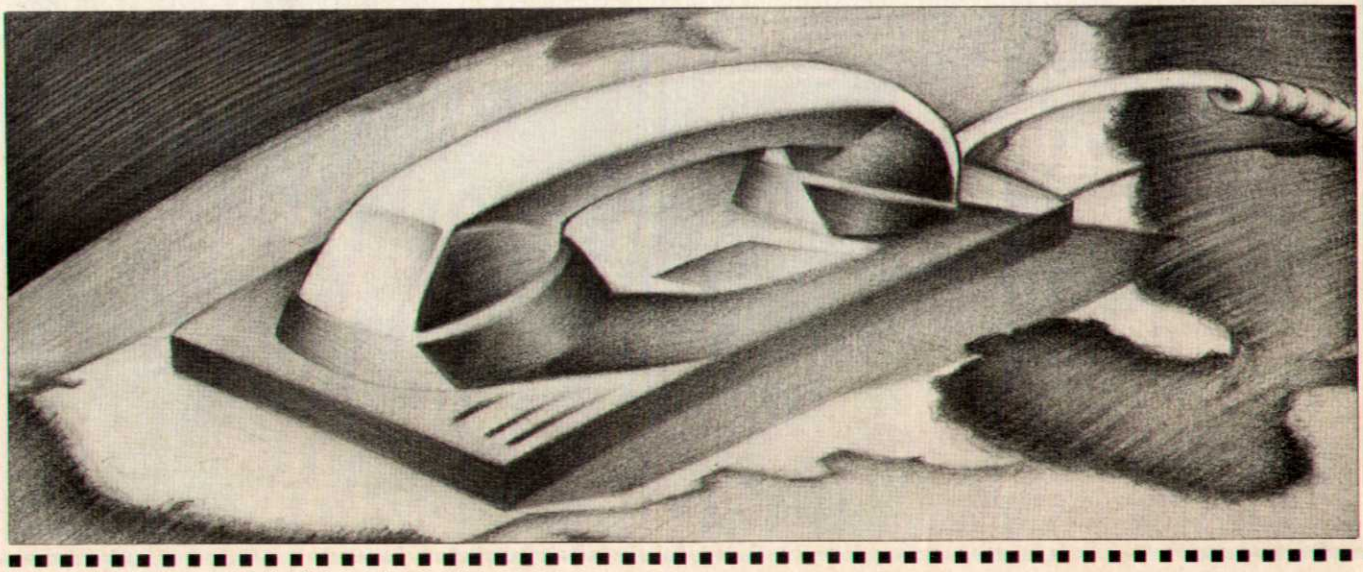
4. The receiver selects LOAD File from the AtariWriter menu. In response to the prompt LOAD DEVICE: FILENAME, the receiver types R—R1:. Then he switches his modem to ANSWER, places his telephone handset into the couplers on the modem, and presses RETURN on his keyboard.

5. Meanwhile, the sender selects SAVE File from the AtariWriter menu and responds to the prompt SAVE DEVICE:FILENAME with the same instructions as the person receiving: R—R1.

When the receiver has placed his telephone handset into his modem, the sender should hear a distinct buzzing noise at his end of the line. The sender then switches his modem to ORIGINATE, places his telephone handset into the couplers on his modem, and presses RETURN on his keyboard. Transmission then begins.

6. While the sender hears a transmission noise from his TV speaker, the receiver neither hears nor sees any response from his computer until transmission is complete. At that time the prompt SELECT LETTER reappears on both TV screens. The receiver may then edit, save, or do whatever he wishes with the file received. ■

Steven Young is Supervisor of Software Writing, Marketing Publications, in Atari's Home Computer Division.



Illustrated by Lance Jackson

High-Tech Fun at Computer Camps

Telecommunications Comes to Camp Atari

by Robin Bernheim

IF YOU LISTEN QUIETLY, you can hear the sounds of summer. The wind through the trees. The noisy little brook. The chirping birds. The echoes of laughter. And the whirring of computers.

An unlikely combination? Perhaps, but it's what Atari Computer Campers can expect this summer. After successfully operating three camps in 1982, Atari now runs computer camps in Greenfield, Massachusetts; Faribault, Minnesota (a suburb of Minneapolis); East Stroudsburg, Pennsylvania (in the Pocono Mountains); Asheville, North Carolina; Glencoe, Maryland (a suburb of Baltimore); Danville, California; and San Diego, California.

As the campers (boys and girls ages 10 through 16) will quickly tell you, Camp Atari is not just another summer camp. Nor is it summer school. And it's certainly not a place where you go to practice your video game skills. But it *is* a place where the future touches the lives of young people, firing their imaginations with a glimpse into tomorrow.

A Total Computer Experience

The high-tech excitement and learning happens both inside and outside the classroom. Inside the classroom, campers learn to become computer users and, depending on their level of expertise, either acquire some fundamental programming skills or polish and build on already existing programming knowledge. Outside the classroom, campers use computers in a variety of settings, gaining exposure to some of the more fantastic applications of computers in everyday life.

The challenge of making this kind of learning experience fun and enjoyable for campers falls to the curriculum development team at Atari, headed by Robert Kahn. Kahn and his group have been involved with the camp project since its inception and have developed a program of instruction for beginning, intermediate and advanced level campers. What a youngster gains from the camp experience, Kahn points out, depends entirely on the individual, as well as the amount of time he or she spends at camp. While most campers spend two or four weeks, some enthusiastic youngsters will be staying for the entire summer.

At Camp Atari, lessons and applications are practical, relevant, and oftentimes unique. Campers do not merely learn how to operate a word processor—they actually use it to write letters home. They learn about the application of database management software by using it to sort and store information on their favorite athletes or organize their record collection. In essence, campers develop an understanding of how the computer can help them manage and simplify large or complex tasks. And, someday, they may be using this same software to write a term paper or track business expenses.

Other traditional camp activities at Camp Atari have been given a face-lift through the use and application of the home computer. If you think arts and crafts means finger painting and pressing leaves, take another look. Atari Computer Campers are using the computer to aid in the design of weaving patterns. By preselecting colors and generating patterns through the computer, campers are able to see what their projects will look like on the video monitor before they even pick up their yarn. If they don't like what they see the first time around, they can experiment with different combinations of color and pattern until they come upon the one they like the best. According to consultant Judy Kohl, who was instrumental in the design of the weaving program, "campers are only limited by their own imagination" when it comes to creating a design with the computer.

Campers' Network to Share Information

Imagination seems to be an operative word at Atari Computer Camps. Perhaps the most intriguing aspect of the camps this summer is the computer network which links the camps together, giving camp participants firsthand experience with the concept of telecommunications. A network, as any Atari camper will surely know, is the linking of a common storage facility. A network can be local—that is, several computers within an office or classroom can be "hooked together," or a network can link computers over great distances, using regular telephone lines. The computers used at camp are connected to the network through a modem, a device which transmits electronic data from the computer over the telephone lines.

The host computer used to link the camps this summer is the same one used by a special department at Atari, the Atari Institute, to share information among people involved with its various projects. The Institute's custom-designed computer system, located in Sunnyvale, California, has five automatic answering modems, permitting campers to log on the computer at any time, day or night.

Specially designed software makes it possible for Atari campers to use the network for a variety of functions. "Chat," for example, is very much like a telephone call—it allows two or more users who are hooked into the system at the same time to exchange information. Through a function called "mail," campers at one camp are able to leave messages for another camp, even if the intended receiver is not logged on the system at the same time. Thus, when it is 8:00 a.m. in North Carolina, Camp Atari-Asheville can leave a message for Camp Atari-San Diego, even though campers in San Diego are still asleep. Then, when Camp Atari-San Diego logs on, campers receive the message "you have mail" and can review the information left by Asheville at their convenience. Through these "computer pen pals," youngsters not only can make friends at their camp, but can get to know campers at other locations across the country—a real-life example of the extended electronic community.

The network also allows its users to send programs from one place to another, so that camps can share programming ideas and data. This powerful feature makes it possible for teachers at one location to share successful programming projects, as well as problems, with teachers at another location. In this way, campers benefit not only from the expertise of their instructor, but also from the collective knowledge of all the individuals tied into the camp network. "Cross execution," a related function, permits two or more locations to use a program at the same time and even receive output in different languages—Spanish and English, for example.

According to Robert Kahn, campers can use the network to tie into existing databases and can browse through information on everything from airline schedules and stock quotations to a list of the best restaurants in New York City. The implications of networking are far-reaching, and Kahn is eager to have campers become aware of telecommunications as a concept. "Telecommunications is a big aspect of where computers in the home are going," he points out, "and it makes good sense to have our kids exposed to that in camp."

One last feature of the camp network merits mentioning. Teleconferencing allows campers at all locations to carry on a running commentary on a particular topic. Unlike "mail" or "chat," the computer keeps a permanent record of all the teleconference discussions.

So, what kind of information has been recorded on the topic of Camp Atari?

"Any foxes at Asheville?"

"Had a pillow fight in San Diego."

"Found several unidentified life forms in yesterday's lunch."

"Camp Atari-Poconos is number one!"

The world of tomorrow has arrived at Camp Atari. But some things never change. . . .

For more information on Atari Computer Camps, call toll-free (800) 847-4180 or write: Atari Computer Camps/Department AA/40 East 34th St./New York, NY/10016. ■

Robin Bernheim is a professional writer and is the Manager of Special Projects for Atari.

ONLY

ATARI GIVES YOU SUCH EASILY ACCESSIBLE GRAPHICS AND SOUND CAPABILITIES IN A 64K COMPUTER.

ONLY *the ATARI 1200XL Home Computer combines 64K RAM computing power with outstanding graphics and sound capabilities that are so easy to access. You can create sophisticated graphics and sound effects using quick and simple commands. This allows you to write complex programs in much less time than with many other 64K computers.*

ONLY *the ATARI 1200XL Home Computer has a specially designed keyboard which is fully redefinable and features 8 programmable function keys controlling 16 functions. That's twice as many as the Commodore 64. In addition, the ATARI 1200XL keyboard locks and unlocks electronically to protect your programs.*

ONLY *the ATARI 1200XL Home Computer provides a revolutionary "help" key that performs two important functions. First, it can provide the user with helpful instructions on selected programs used with the computer. Second, the "help"*

THE NEW

key performs self-diagnosis—as requested by the user on the memory, audio visual circuitry and keyboard—to ensure that all components are in peak operating condition. And if you need additional help only Atari gives you a toll-free "help line" to call (800) 538-8543; in California 1-(800) 672-1404.

***ONLY** the ATARI 1200XL offers you a home computer compatible with virtually all ATARI Computer peripherals and software (compatibility that other new computers like the Commodore 64 don't offer). There are over 2,000 programs currently available for the ATARI 1200XL, including the new AtariWriter™ word processing system and VisiCalc.**

***ONLY** Atari puts so much more in the new 1200XL Home Computer so you get so much more out of it.*

ATARI® 1200XL™

HOME COMPUTER

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

CAREERS IN COMPUTERS

Why Computing Isn't Just for Boys

by Teddi Converse

"I KNEW FROM THE FIRST TIME I saw Captain Kirk talk to the computer on 'Star Trek' that I wanted to work with computers," says Lisa Van Stone, a 21-year-old programming enthusiast from Palo Alto, California, and chemistry major at Princeton University.

"When I was in high school there was a particular computer-center crowd that consisted of all guys," she reflects. "So my friend and I decided we were going to break into the group. The boys were really threatened by us, and I never really understood why . . . I've always been attracted to high-tech things."

Just this year, *Time* magazine named the computer the "Machine of the Year," thereby formally acknowledging a technical revolution that has been "four decades in the making." Admittedly, most of the forerunners in this burgeoning technology have been male. But after encountering Lisa Van Stone, we began to look for other female whiz kids. We found that Lisa wasn't an anomaly among young women from the South San Francisco Bay Area. In fact, Santa Clara County, now known worldwide as California's Silicon Valley, is a prime location for finding youngsters weaned in the glow of the video screen.

Women as Programmers

THESE DAYS, MORE AND MORE women are getting into programming. This comes as no surprise. After all, the very first computer programmers were women. They were hired by the U.S. Navy to program ENIAC, one of the first operational computers built by the United States, which was used to calculate shell trajectories.

For increasing numbers of young women, a knowledge of programming has proven to be helpful in school, at home, and in their careers. Perhaps even more important, the ability to program a computer imparts self-esteem, a sense of accomplishment, and a leading edge in the "information revolution"—which without a doubt is here to stay.

Like many young programmers these days, Lisa Van Stone was introduced to her first computer at school. She was one of the first participants in the "Computer Tutors" program at the Jordan Middle School in Palo Alto, where children are taught to use computers and then teach their friends and peers. "The Computer Center was like my second home," explains Lisa. "It was a place where the usefulness of computers was really emphasized, as opposed to hacking for the sake of hacking."

Teachers involved in computer education usually focus on the practical applications of programming. With the recent emphasis on "computer literacy" in the schools, this practical approach is proving to be the most effective way to get girls, as well as boys, started in computers. It also helps to spur young people's interest in programming on their own.

Richard Pugh, a computer teacher at John F. Kennedy Junior High School in Cupertino, California, helps students get hooked on programming by emphasizing the sound and graphics capabilities of Atari Home Computers. "I start them out writing small programs that print their initials across the screen or creating simple designs," he explains.

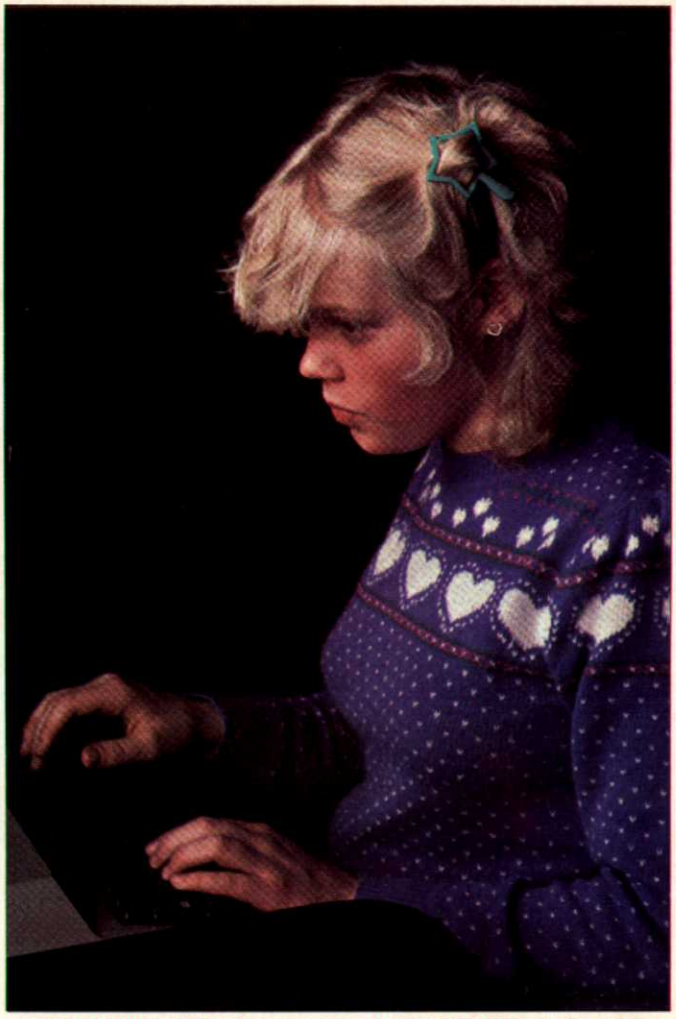
"I've found that, at first, taking the computer apart and analyzing a chip or lecturing the kids on the different statements used in BASIC doesn't really sink in. But writing small programs that mean something to the students—as a way to express themselves—gets them interested in learning more about the programming process." Pugh's approach seems to be working. The proportion of girls taking his computer class has risen from 10% of the class last year to over 40% this year.

Cori Grimm is no stranger to the possibilities of the graphic element in designing programs. Just twelve years of age, she has been creating graphic designs for products of The Learning Company—the same company that designed *Juggles' House* and *Juggles' Rainbow*—for over three years. And she gets paid for doing it!

"I've always been interested in art," says Cori. "But, I can't seem to make things look like I want them to on paper because I usually end up erasing *everything*. With the computer you can erase and redraw as much as you want. And you can also change colors easily."

The Grimms are what you might call a computing family. Cori's father and mother are programmers. Her sister, Cindy, "crash tests" The Learning Company programs and, on occasion, helps Cori conceptualize her graphics characters.

"Young women programmers should find a mentor."



CONCENTRATION OF A PROGRAMMER: Kris Klein Studies a Program to See Where the Bugs Are

Cori's electronic artwork appears in over 30 of The Learning Company's products. She uses graphics utility programs and graphics tablets to do most of her work. Graphics tablets can work in a number of different ways, depending on the software used with them. For example, the graphics tablet can be used to draw colorful, detailed pictures which can be then saved as a subroutine onto a diskette. The "picture subroutine" can then be inserted into a master learning program that may teach spelling or math. This technique saves hours of programming time plotting the individual X, Y coordinates that make up piece computer graphic art.

Like many girls who program, Cori learned most of her programming skills from playing with the Grimm's home computer. "I took one computer class at school," she explains. "I ended up practically teaching it. I somehow got dumped into a beginning class with a substitute teacher who didn't know very much about computers," she laughs. "So I answered everyone's questions."

What does it take to be a programmer? "Well, first it takes convincing yourself—or having someone convince you—that computers aren't frightening," says Lisa Van Stone. After initially being exposed to computers at school, Lisa became involved in a project programming the SmallTalk computer at the Xerox Research Center in Palo Alto.

In the tenth grade she combined her bookkeeping and computer-programming skills to design an accounting program for her mother's bookkeeping business. When she saw that her mother needed to have some knowledge of programming to understand how the programs worked, Lisa taught her

How to Get Started in a Computer Career

Industry experts estimate nearly four million computers will be purchased by individuals and businesses next year. It's apparent that practically *every* industry will be using computers in the near future. So now is the time for women serious about their careers and getting ahead to take advantage of the opportunities available.

"You can sell yourself on the skills you have right now, if you understand what your skills are and how to communicate them to a company," write Bower and Kunkin-Heller in *Computer Confidence: A Woman's Guide*. "You don't always need more education to get a job."

Many computer-related occupations require specific training in computer science or engineering. But in large corporations, even those in high-tech fields, professional and management positions exist in many nontechnical areas—sales, marketing, finance, accounting, merchandising, and manufacturing, for example. Enter the electronics industry with strengths in any of these areas and you can become a valuable asset to a company, *and* learn more about their specific technology.

In addition to traditional technical positions—like **systems analyst, applications programmer** and **systems programmer**—electronics companies need **training** and **customer support specialists.** Job responsibilities of training and customer support people can include anything from training retailers on a particular system to answering phone calls and fielding consumer problems.

The electronics revolution in telecommunications, computer graphics, and office automation provides opportunities for people who are versed in other areas but interested in entering a high-tech field. Electronics companies often seek **writers, artists, illustrators** and **designers** to produce documentation—instruction manuals, display boxes and promotional materials, for example. Artists and writers are often hired to work with programming teams to create story lines and video images for consumer products.

Individuals with some programming experience or a background in electronics or engineering may want to consider **technical writing,** another fast-growing field within the computer job market. Written documentation is becoming increasingly important for both computer systems (hardware) and software programs. People with technical skills who can also demonstrate "a way with words" can apply their skills here.

Entry-level positions in **word processing** and **data entry** are becoming more available, as a result of increased use of computers to manage information. People working in these positions often command higher salaries than do clerks, secretaries, and other workers who possess traditional office skills.

In order to lure prospective employees, many electronics companies also offer attractive benefits—flex-time, bonuses, royalties and comfortable work spaces that include company-subsidized cafeterias and health facilities—especially for skilled programmers and technical specialists. As the computer industry continues to grow, so will the need for good and talented employees.

If you're interested in making a transition into a high-tech job, "personnel departments aren't necessarily the best place to begin your search for a position," say Bower and Kunkin-Heller. "Instead, take a high-technology person out to lunch, and get information on jobs and opportunities."

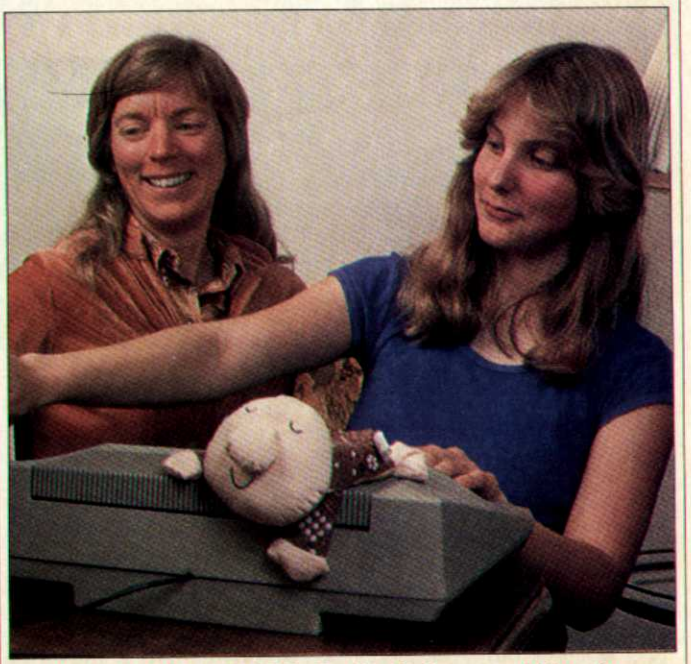
mother how to program in BASIC. "That took me about a year," laughs Lisa. "But once my mother learned how to program, she was able to modify the program to fit each of her client's individual needs." Her mother's business rapidly grew from one client to twelve.

Programming and Math: The Great Misconception

ALTHOUGH A GREAT DEAL OF discussion is being bandied about these days concerning computers and mathophobia, one thing learning how to program doesn't require is a large amount of mathematical training. Unfortunately, females suffer the most from the fear of computers, machines and technology.

"Part of the problem," says Joan Targ, educator and president of Interactive Sciences, Inc., Palo Alto, California, "is that people have traditionally presented computers as a male domain." Cori's mother, Leslie Grimm, concurs. "Many beginning computer classes use programs for teaching purposes that involve high-level mathematics, for example," she adds. "And, really, you can do a lot of things with programming that require no more than simple addition."

Judy Bogart, a professional games programmer at Atari, agrees. "The connection between programming and high-



TWO GENERATIONS: Leslie Grimm and Daughter Cori Create Graphics for Educational Software by The Learning Company

level mathematics is very overstated," she says. "The type of skill you really need for programming is logical, clear thinking. You need to be able to organize a series of steps to tell the computer what to do in a logical manner."

According to Targ, many techniques can be employed to combat the notion that you need a lot of math to program a computer. For example, you can get involved with programs that use graphics and sound or word-processing applications. "We try to provide the message that computer programming is not just a boy's activity," says Joan Targ. "And we do this not only by de-emphasizing mathematics, but also by exposing children to women who are active in the industry and to their female peers. We have girls teach other girls how to program."

Advice from Women Programmers

MOST EXPERTS WE INTERVIEWED agree that to become proficient in programming, girls should sit down at a computer and start experimenting. "You just have to kind of play around with the computer and see what works," says Kris Klein, who, at the age of twelve, has decided that the consumer-electronics field is "definitely the field I want to major in when I get to college." Although presently taking only her second computer class, she shows great promise in her programming ability.

"My advice to young women wanting to get into programming is to find a mentor," says Judy Bogart. "Most of my programming knowledge has come from family, friends, and staying after school to ask the teacher questions. Programming does take patience, but it's very rewarding and very creative. I put a lot of time and energy into every program I do, but in the end there is a finished product—you're actually *making* something."

Even if you have a computer at home, taking a computer course at school is probably the best way to get formally introduced to programming. Most classes teach BASIC or PILOT languages to the novice programmer, although LOGO is becoming increasingly popular among beginners. "I really like LOGO," says Cori Grimm. "It's a higher-level language, and graphics are so easy to use." A few Assembly Language courses are also being offered to more advanced computer users.

If your school doesn't have any computers or computer classes with which to work, your community might consider lobbying for them. "Now is the time for parents and teachers to push for reasonable access to this technology at all grade levels," says Jeff Levinsky of the Jordan Middle School computer center in Palo Alto. Major electronics firms provide support through special organizations such as the Atari Institute for Educational Action Research to provide computers and funding for computer curricula. All across the country, schools are holding fundraisers such as carnivals, walkathons, and bake sales to earn money to purchase computers.

Another way for young girls to get started in computers is through public-access computers available at many local libraries. Or they can rent computer time at computer stores. Girls' clubs also offer an alternative. According to Carol Campbell of the Santa Clara County Girl Scout Council, young girls can earn a "Computer Fun" badge through the Girl Scouts by completing projects such as using a word processor to publish a troop newsletter or inviting the Atari IEC Mobile Computer Van to their school.

A less traditional way for young girls to learn programming is to spend some time at a computer camp. A unique combination of fun and learning, a computer camp offers all the usual summer camp activities—tennis, swimming, hiking and cookouts—and usually three to four hours a day of program-

Silicon Valley Girls: The Next Generation

BY NOW MOST EVERYONE in the country has heard of "Valley Girls," those "like, totally awesome" denizens of Southern California's San Fernando Valley, immortalized in Moon Zappa's famous song as being teenage hip, "like totally."

Well, California's *other* famous valley, the Santa Clara Valley, has some "Valley Girls" of its own. These kids don't spend much time aimlessly meandering through shopping malls looking for the neatest miniskirts. You're much more likely to find them browsing in computer stores—looking for the neatest new software—and hanging out at school computer centers.

Located in the rolling hills of Northern California just south of San Francisco Bay, the Santa Clara Valley is home base for hundreds of electronics companies. Kids growing up in this area are exposed to computers at an early age, both in school and at home. Here, teenage slang is based on high-tech acronyms and abbreviated computer terms, and computer literacy is a requisite for being able to communicate with other kids.

In the Silicon Valley, youngsters are programmed for success at a very young age. Encouraged by parents, teachers and older friends' contagious enthusiasm for dynamic new high-technology fields, these kids see computing as a way to make exciting new discoveries, a creative way to become involved with the greatest technological revolution the world has ever seen. Here, excitement's in the air, and even elementary school children get caught up in the fun of computer graphics and learning programs.

Many Silicon Valley kids are more pragmatic about the future than their older brothers and sisters were. They're interested in channeling their computer skills into high-paying careers. In fact, some of them are turning these skills into profitable enterprises even before they graduate.

Girls of California's Silicon Valley see themselves as innovators in a fascinating new world of electronic adventure. This new breed of kids is making waves that are rippling across the continent, and forging an exciting new future—one they're looking forward to creating.

Judy Bogart, a Games Programmer at Atari, Teaches Ten-year-old Audrey Li Some Programming Shortcuts

ming instruction on Atari Home Computers from trained staff members. "We'd really like to get more girls involved with the camp," says Bob Kahn, director of the Atari Special Projects team that develops curricula for the camps. "Everybody at camp has a better time when there is an equal mixture of boys and girls."

"Camp is really fun," says Shirley Stas, who taught at a computer camp last summer. "And not only did the kids learn a lot . . . but I learned a lot from the kids, too!"

Career Opportunities

CAREER OPPORTUNITIES FOR GIRLS who are both skillful in and comfortable with computer technology are nearly unlimited. "Computers can help you obtain a job now or in the future and advance you in your current job," say Dorothy Kunkin Heller and June Bower, authors of *Computer Confidence: A Woman's Guide*. (Acropolis Books, Ltd.; hardcover: $16.95, softcover: $9.95.)

Heller adds that a firm foundation of computer knowledge can help young women in their present jobs and make them more marketable for the future. "Getting *comfortable* with computers can change your attitude, your paycheck, and your life!"

"Women are really starting to see that they have different choices in the computer field," says Judy Bogart. "There are many programming-related fields for women, too," she continues. "For example, I started out answering questions from other programmers in the Customer Service department at Atari. I found that it was a great place to work and learn more. Technical writing is another good field."

Do girls and young women approach programming differently from the way their male counterparts do? Cori Grimm thinks so. "The way girls and boys relate to their computers and programs is different," she explains. "Most boy programmers I know are more interested in the technical accuracy of a program. And most girl programmers I know pay more attention to what the person who will be *using* the program is going to see and feel. I think about how the person sitting in front of the TV is going to react to my pictures on the screen."

"I really think there's no reason for sex distinction in the field," adds Judy Bogart. "Although professionally it's essential to know how to work as a team and have good communications skills, it seems women have always been encouraged to develop those kinds of communicative skills. No doubt those skills are reflected in the kinds of programs they write."

Endless Choices

WOMEN ENTERING THE COMPUTER FIELD find a wide range of specialties from which to choose. Most women in the field confirm that the possibilities are almost limitless.

"I'm interested in studying the relationship and interaction between humans and computers more," says Lisa Van Stone. "I'd also like to study the effects of technology on people's lives and get involved with introducing computer technology to third world countries."

"Women are really coming up," says Judy Bogart. "I see a bright future ahead for anybody who is willing to put some time into learning to program computers. I'm so delighted when I come across other women who have achieved in my field; I really get excited and want to work that much harder. I encourage every little girl out there to do the same."

Industry experts think that by the year 2000 there will be close to a million new jobs generated for programmers in the U.S. With so many computer companies looking for bright talent, now is the perfect time for women to take advantage of career opportunities, changes, and advancements in computer-related fields.

"A couple of years ago I thought I wanted to be a soccer player," said one bright-eyed sixth grader as her hands flew across her computer keyboard. "Then I decided I wanted to be a doctor. Now I think I'll be a computer programmer. But *maybe* I'll be a doctor who uses computers and plays soccer." ■

Teddi Converse is a Writer for Marketing Publications in the Atari Home Computer Division.

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

Made in U.S.A.
Atari, Inc. Home Computer Division
P.O. Box 61657
Sunnyvale, CA 94086

Games People Feel

COMPUTER CLASSROOM

DATA PROCESSING PART II: *Storing Data on Disk*

By Bill Bartlett

THIS ISSUE, *Computer Classroom* presents the final part of a two-part series describing techniques for data processing with your Atari Home Computer. The first part (see the Spring, 1983, issue of ATARI CONNECTION) explained data storage and retrieval using cassette tape and the Atari 410 or 1010 Program Recorder. This continuation of *Computer Classroom* shows you how to store and retrieve data on diskettes using the Atari 810 Disk Drive. You will notice that many commands and procedures are similar to those used for cassette operation.

The example programs at the end of the article provide you with a working mailing list which you can use just as printed, or it can be customized to suit your own particular needs.

Storing Data

Floppy diskettes are thin plastic disks coated with a metallic oxide which can be magnetically arranged to hold information. Storage of data on diskette is many times more efficient than storage on cassette tape. As you will remember from last issue's *Computer Classroom*, cassette storage of data is *sequential*. Diskettes, however, allow *nonsequential* or "random access" storage of data and save a great deal of time during the storing and retrieving process. In both storage methods the same "bits" and "bytes" are magnetically printed on the medium, but the cassette tape is long and sequential while the diskette is neatly round, merry and quick. Here's how it works!

Formatting

Before you can store data on a diskette, the diskette must be formatted into 40 concentric tracks, each with 18 sectors, for a total of 720 sectors. As each sector is 128 bytes long, total storage capacity per diskette is 92,160 bytes. Use the Atari DOS Utility Routine, Menu Selection "I" to format your diskette.

DOS

The Atari 810 Disk Drive uses a *boot file* to control the initialization process when you power up your computer and disk drive. This means that the computer and disk drive must introduce themselves to each other before they can communicate—a "handshake" in other words. During this introduction, the DOS File Management System (FMS) is loaded into the computer's memory. The FMS is responsible for allocating available sectors as they are needed for file storage, and the boot file itself uses 3 of the 720 sectors on a freshly formatted disk.

DOS creates and maintains a *directory* of up to 64 files which uses 8 sectors of disk space. DOS also maintains a *bit map* of the sectors that have been allocated for file storage. When DOS needs to allocate a starting sector for a data file, it uses the lowest-numbered free sector as indicated in the bit map. Because the disk drive can skip directly to this location without reading information sequentially (as is required along the length of a cassette tape), this capability is commonly referred to as *random access*. The bit map takes up 1 sector of disk space. DOS also uses 3 of the 128 bytes in each sector to identify the file to which that sector belongs and to point to the next sequential sector in the file. File storage space available on a freshly formatted diskette (using Atari DOS version 2.0S), then, is 707 sectors of 125 bytes each, or a total of 88,375 bytes or characters.

To create a data file you write an OPEN command to "open" a file, a series of PRINT #N or PUT #N commands (outputs) to record your data, and a CLOSE command to end the file.

The OPEN Command

There are four modes for using the OPEN command with a disk file. As with the cassette OPEN command, any one of the I/O buffers 1 through 5 are commonly used. "N" represents your choice of available buffers.

1. DATA RETRIEVAL: OPEN #N,4,0,"D:FILENAME" opens a file in the READ mode. DOS locates the FILENAME in the directory and positions its pointer at the first byte of the file. INPUT#N or GET#N are the commands for retrieving data from opened files.

2. DATA STORAGE: OPEN #N,8,0,"D:FILENAME" opens a new file in the WRITE mode. DOS first searches the directory for the FILENAME and, if it exists, deletes it. DOS then creates the new FILENAME in the directory, allocates a free sector as the first sector of the file and positions its pointer at the first byte of that sector. PRINT#N or PUT #N are the commands that store data in this mode. Be careful not to OPEN a file in mode 8 using the name of a

file already on the disk unless you wish to overwrite it with new information. The old file with the same name will be lost!

3. UPDATE: OPEN #N,9,0,"D:FILENAME" opens a file in APPEND mode and is used to increase the size of a file. DOS first locates the FILENAME in the directory. It then allocates a free sector appended to the end of the file and positions its pointer at the first byte of that sector, ready to write new data. Mode 9 is for writing only, so PRINT #N and PUT #N are the correct commands.

4. REPLACE: OPEN #N,12,0,"D:FILENAME" opens a file in READ/WRITE mode. DOS locates the FILENAME in the directory and positions its pointer at the first byte of the file. PRINT, PUT, INPUT, and GET are all correct commands for this mode. However, the PRINT and PUT commands will write over any data at the location on the disk to which the drive is pointed. It is therefore important that you always know your location on the disk when working in this mode, or you may destroy needed data by overwriting. The NOTE command is used to find your current location by returning to you the sector number and the precise byte offset within that sector to which the drive is pointed. The POINT command, on the other hand, is used to direct the drive to a precisely specified location. Mode 12 may be thought of as the RANDOM ACCESS mode because you can skip at will to any location that is within your data file.

Close

The CLOSE command releases the I/O buffer connected to the disk drive so that it may be used for other devices or operations. It is good practice to CLOSE every file that is opened in a program, but it is *imperative* that you CLOSE any file opened in modes 8, 9 or 12. If these are not CLOSEd it is probable the bit map or forward pointers on your disk will not be properly updated, resulting in a "File Number Mismatch" (Error 164). Your disk will have to be reformatted and you may lose important data!

Input/Output Commands (I/O)

The commands INPUT, GET, PRINT and PUT transfer data between the computer's memory and your disk file. As in all I/O operations, a buffer is used as a staging area for the data. When the buffer fills up with 125 bytes, DOS writes the contents of the buffer to your disk file, allocates another free sector, and clears the buffer for more data.

Two types of Input and Output (I/O) commands are used to write data to a file on disk: *Character I/O* and *Record I/O*. Character I/O (PUT) means that you write data one byte at a time with none of the values interpreted as special control characters—in other words, as "raw data." The statement PUT #N,X transfers one character to the data file at a time. The character itself is stored on the diskette as a code number (represented here as the variable "X") according to the American Standard Code for Information Interchange or "ASCII" rules. Record I/O (PRINT) means that your data is recorded one "field" at a time. A field can be a string of characters or a single value. In this mode, an *END of Line* (EOL) character (ASCII code 155) marks the end of each field and is automatically generated by the PRINT #N statement. The proper syntax for the PRINT statement should include a *semicolon* not a comma.

As with cassette I/O, there are three ways to retrieve all your data from a file and exit without an error (End of File, Error 136). First, if you know how many fields were written in this file, you can simply retrieve exactly the same number of fields. Secondly, if the number of fields changes, you can create a field with a special value at the end of the file and check for this value after each input. If you don't know what's in the file, you can use the TRAP command. When the *End of File, Error 136* occurs, the TRAP command can prevent your program from "bombing out" because the program will next execute any line you direct it to instead of stopping the whole works. If you want to be sure that the error really was an *End of File, Error 136*, simply TRAP your program to a statement that PEEKs(195) and see what's there. It should be 136!

Note/Point

The command NOTE #N,S,B is used to determine the precise location on the disk to which your drive is pointed at the time of the command. Variable "S" is the sector, variable "B" is the byte offset within that sector. This "address" is a specific and precise point on the disk. Conversely, the command POINT #N,S,B positions your drive to the point you specify by giving the variables S and B your address requirements. When updating an existing file be careful to replace or overwrite the exact number of bytes which are in your data field. When using the *random access* Mode 12 and the NOTE and POINT features you are navigating in concentric tunnels marked with logically sequenced addresses. While you can open doors between tunnels and skip about within the boundaries of your open file (random access), as a navigator you always need to know your location in order to avoid losing important data or causing cave-ins and disasters requiring reformatting of the diskette.

Using the Sample Mailing List Programs

The following program listings create and maintain a simple mailing list using the Atari 810 Disk Drive. Mailing List programs provide a good example of storing and retrieving data in data files. The first program (DSKADD) sets up your file and stores records. The second program (DSKUP) updates the information in the file. The third program (DSKPRINT) allows you to print the contents of your mailing list. With a small amount of programming effort on your part, you might even wish to combine all three into a single mailing list management program.

You may want to save your mailing list program on a separate "program diskette" and keep your mailing list files on a second "data diskette."

Key concepts illustrated by these programs are the use of OPEN, PRINT, and TRAP commands. You will also notice that more than one data file may be opened and running concurrently on the same disk.

Pay attention to the REMark statements, as they help explain what the program is doing. ■

Bill Bartlett is Manager of Product Support for the Atari Consumer Product Division.

Program 1

DSKADD

Use this program to add new records, and to create a mailing list file named CUSTOMER.DAT. *The program creates a temporary file containing your records, adds new records, then deletes the old file and renames the temporary file so that it becomes the permanent* CUSTOMER.DAT *file.*

```
 10 PRINT CHR$(125):REM clear screen
 20 PRINT "THIS PROGRAM ADDS RECORDS"
 30 PRINT "FOR NEW CUSTOMERS.":PRINT
 40 PRINT "INSERT THE PROPER DISKETTE,"
 50 PRINT "PRESS 'START' TO CONTINUE..."
 60 IF PEEK(53279)<>6 THEN 60:REM wait for start key
 65 REM --set up variables and filenames--
 70 DIM ID$(9),NAME$(24),ADDR$(24),CITY$(16),STATE$(2),ZIP$(5),PHONE$(12)
 80 DIM FILE1$(16),FILE2$(16)
 90 FILE1$="D:CUSTOMER.DAT":FILE2$="D:CUSTOMER.TMP"
 95 REM --open the files--
100 CLOSE #1:CLOSE #2:REM close I/O buffers if currently open
110 TRAP 200:OPEN #1,4,0,FILE1$:TRAP 40000:REM check for 'no file found' error
120 OPEN #2,8,0,FILE2$:REM if no error, open temporary file for writing
130 GOTO 300:REM if no error, skip to get new records
180 REM xxxxxxxxxxxxxxxxxxxxxxxx
190 REM --error routine--
195 REM if file is not on this disk then create a file, or insert proper disk
200 PRINT CHR$(253);"CUSTOMER FILE NOT ON THIS DISK,":PRINT :REM sound buzzer
210 PRINT "PRESS 'START' TO TRY ANOTHER DISK-"
220 PRINT "PRESS 'SELECT' TO CREATE ON THIS DISK-"
230 IF PEEK(53279)=6 THEN 100:REM if start is pressed, try opening again
240 IF PEEK(53279)<>5 THEN 230:REM check for select
250 CLOSE #1:OPEN #1,8,0,FILE1$:REM if creating new file, open it
260 PRINT #1;"ENDOFFILE":REM write file with no records
270 CLOSE #1:GOTO 100:REM now that there is a file, go try again
280 REM xxxxxxxxxxxxxxxxxxxxxxxx
290 REM --transfer existing records to temporary file--
300 INPUT #1,ID$:REM get record number
305 IF ID$="ENDOFFILE" THEN 400:REM last record, go to add-record routine
310 PRINT #2;ID$:REM transfer number to new file
320 PRINT "TRANSFERRING TO TEMP FILE...";ID$
330 INPUT #1,NAME$:PRINT #2;NAME$
340 INPUT #1,ADDR$:PRINT #2;ADDR$
350 INPUT #1,CITY$:PRINT #2;CITY$
360 INPUT #1,STATE$:PRINT #2;STATE$
370 INPUT #1,ZIP$:PRINT #2;ZIP$
380 INPUT #1,PHONE$:PRINT #2;PHONE$
390 GOTO 300:REM get next record
395 REM xxxxxxxxxxxxxxxxxxxxxxxx
396 REM --add new records to file--
400 PRINT CHR$(125);"SPECIFY RECORD TO ADD";CHR$(29):REM (move cursor down)
410 PRINT "ID NUMBER OR 'END'...":INPUT ID$:IF ID$="END" THEN 600
420 PRINT "NAME...";:INPUT NAME$
425 PRINT "ADDRESS...";:INPUT ADDR$
430 PRINT "CITY...";:INPUT CITY$
435 PRINT "STATE...";:INPUT STATE$
440 PRINT "ZIP...";:INPUT ZIP$
445 PRINT "PHONE...";:INPUT PHONE$
450 PRINT :PRINT "PRESS 'SELECT' TO ADD RECORD..."
460 PRINT "PRESS 'OPTION' TO RE-ENTER..."
470 IF PEEK(53279)=3 THEN 400:REM option is pressed, re-enter record
480 IF PEEK(53279)<>5 THEN 470:REM check for select key
485 REM xxxxxxxxxxxxxxxxxxxxxxxx
490 REM xxxxxxxxxxxxxxxxxxxxxxxx
495 REM --write new record to temporary file--
500 PRINT #2;ID$
510 PRINT #2;NAME$
520 PRINT #2;ADDR$
530 PRINT #2;CITY$
540 PRINT #2;STATE$
550 PRINT #2;ZIP$
560 PRINT #2;PHONE$
570 GOTO 400:REM go get more new records
580 REM xxxxxxxxxxxxxxxxxxxxxxxx
590 REM --closing routine--
600 PRINT #2;"ENDOFFILE":REM write end-of-file marker record
610 CLOSE #1:CLOSE #2
620 PRINT CHR$(125);"DELETING OLD FILE..."
630 XIO 33,#1,0,0,FILE1$:REM delete old file
640 PRINT :PRINT "RENAMING NEW FILE..."
650 XIO 32,#1,0,0,"D:CUSTOMER.TMP,CUSTOMER.DAT"
660 PRINT :PRINT "--END OF PROGRAM--"
670 END
```

Program 2

DSKUP

Use this program to change or delete existing records in the CUSTOMER.DAT *file. The program reads records from* CUSTOMER.DAT *file, updates a temporary file, then deletes the old file, renames the temporary file and converts it into a new permanent* CUSTOMER.DAT *file.*

```
 10 PRINT CHR$(125):REM clear screen
 20 PRINT "THIS PROGRAM CHANGES OR DELETES"
 30 PRINT "EXISTING RECORDS IN THE DATA FILE.":PRINT
 40 PRINT "PRESS 'START' TO CONTINUE...":PRINT
 50 IF PEEK(53279)<>6 THEN 50:REM wait for start key
 60 REM set up variables and file names
 70 DIM ID$(9),NAME$(24),ADDR$(24),CITY$(16),STATE$(2),ZIP$(5),PHONE$(12)
 80 DIM FILE1$(16),FILE2$(16)
 90 FILE1$="D:CUSTOMER.DAT":FILE2$="D:CUSTOMER.TMP"
 95 REM -- open the files --
100 CLOSE #1:CLOSE #2:REM close any files which are open
110 TRAP 200:OPEN #1,4,0,FILE1$:TRAP 40000:REM trap file-not-found error
120 OPEN #2,8,0,FILE2$:REM if no error, open write file
130 GOTO 300:REM skip error routine
140 REM xxxxxxxxxxxxxxxxxxxxxxxx
150 REM -- error routine --
160 REM if file is not on this diskette, try another one
200 PRINT CHR$(253);"CUSTOMER FILE NOT ON THIS DISK,":PRINT :REM sound buzzer
210 PRINT "PRESS 'START' TO TRY ANOTHER DISK..."
220 IF PEEK(53279)<>6 THEN 220:REM wait for start key
230 GOTO 100:REM try again
240 REM xxxxxxxxxxxxxxxxxxxxxxxx
250 REM -- read a record from the old file --
300 INPUT #1,ID$:REM get record number
305 IF ID$="ENDOFFILE" THEN 600:REM last record, go to closing routine
310 INPUT #1,NAME$,ADDR$,CITY$,STATE$,ZIP$,PHONE$:REM read rest of record
315 REM -- display the record --
320 PRINT CHR$(125);"DATA IN OLD FILE";CHR$(29):REM (move cursor down)
330 PRINT "ID",ID$
```

```
335 PRINT "NAME",NAME$
340 PRINT "ADDRESS",ADDR$
345 PRINT "CITY",CITY$
350 PRINT "STATE",STATE$
355 PRINT "ZIP",ZIP$
360 PRINT "PHONE",PHONE$
370 PRINT :PRINT "PRESS 'OPTION' TO MODIFY RECORD-"
375 PRINT "PRESS 'SELECT' TO KEEP RECORD AS IS-"
380 PRINT "PRESS 'START' TO DELETE RECORD-"
390 IF PEEK(53279)=6 THEN 300:REM get another record, don't save this one
391 IF PEEK(53279)=5 THEN 500:REM add this record to the new file
392 IF PEEK(53279)<>3 THEN 390:REM check for option key
395 REM xxxxxxxxxxxxxxxxxxxxxxx
397 REM -- modify data in record --
400 PRINT :PRINT "ENTER NEW DATA FOR RECORD":PRINT
410 PRINT "ID NUMBER...";:INPUT ID$
420 PRINT "NAME...";:INPUT NAME$
430 PRINT "ADDRESS...";:INPUT ADDR$
440 PRINT "CITY...";:INPUT CITY$
450 PRINT "STATE...";:INPUT STATE$
460 PRINT "ZIP...";:INPUT ZIP$
470 PRINT "PHONE...";:INPUT PHONE$
480 PRINT :PRINT "PRESS 'SELECT' TO ADD RECORD..."
485 PRINT "PRESS 'OPTION' TO RE-ENTER..."
490 IF PEEK(53279)=3 THEN 400:REM re-enter the data
491 IF PEEK(53279)<>5 THEN 490:REM check for select key
495 REM xxxxxxxxxxxxxxxxxxxxxxx
496 REM -- add the new record to the temporary file --
500 PRINT #2;ID$
510 PRINT #2;NAME$
520 PRINT #2;ADDR$
530 PRINT #2;CITY$
540 PRINT #2;STATE$
550 PRINT #2;ZIP$
560 PRINT #2;PHONE$
570 GOTO 300:REM go read another record from old file
580 REM xxxxxxxxxxxxxxxxxxxxxxx
590 REM -- closing routine --
595 REM write end-of-file marker record, close files,
596 REM delete old file and rename new one to permanent file.
600 PRINT #2;"ENDOFFILE"
610 CLOSE #1:CLOSE #2
620 PRINT CHR$(125);"DELETING OLD FILE..."
630 XIO 33,#1,0,0,FILE1$:REM delete old file
```

Program 3

DSKPRINT

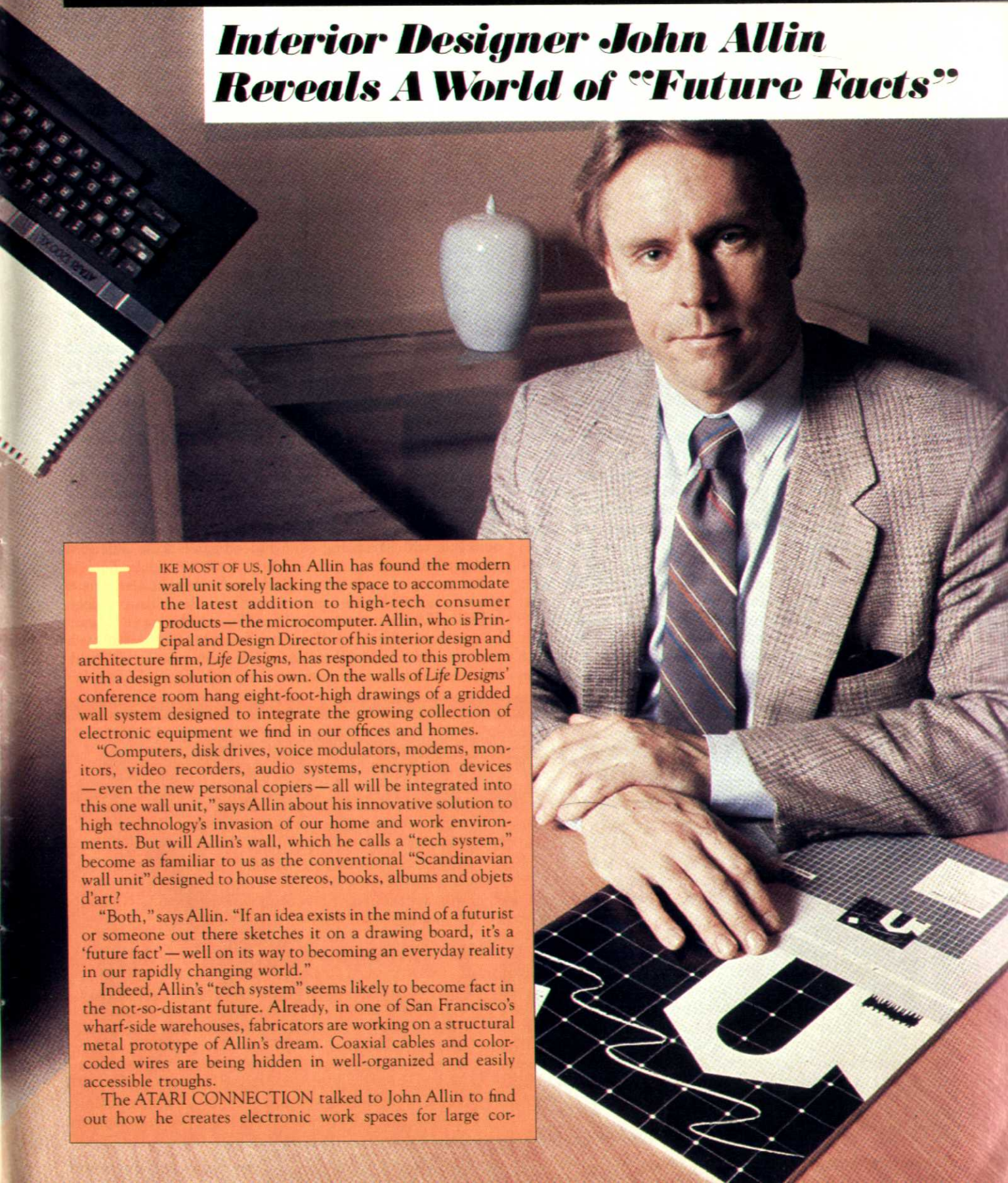
Use this program to print out records from the CUSTOMER.DAT *file.*

```
 10 PRINT CHR$(125):REM clear screen
 20 PRINT "THIS PROGRAM PRINTS ALL RECORDS"
 30 PRINT "FROM THE DATA FILE ON A PRINTER.":PRINT
 40 PRINT "PRESS 'START' TO CONTINUE...":PRINT
 50 IF PEEK(53279)<>6 THEN 50:REM wait for start key
 60 REM set up variables and file names
 70 DIM ID$(9),NAME$(24),ADDR$(24),CITY$(16),STATE$(2),ZIP$(5),PHONE$(12)
 80 DIM FILE1$(16),FILE2$(16)
 90 FILE1$="D:CUSTOMER.DAT":FILE2$="P:"
 95 REM -- open the files --
100 CLOSE #1:CLOSE #2:REM close any files which are open
110 TRAP 200:OPEN #1,4,0,FILE1$:TRAP 40000:REM trap file-not-found error
120 TRAP 250:OPEN #2,8,0,FILE2$:TRAP 40000:REM trap printer-not-ready error
130 GOTO 300:REM skip error routine
140 REM xxxxxxxxxxxxxxxxxxxxxxx
150 REM -- error routine, file not found --
200 PRINT CHR$(253);"CUSTOMER FILE NOT ON THIS DISK,":PRINT :REM sound buzzer
210 PRINT "PRESS 'START' TO TRY ANOTHER DISK..."
220 IF PEEK(53279)<>6 THEN 220:REM wait for start key
230 GOTO 100:REM try again
235 REM xxxxxxxxxxxxxxxxxxxxxxx
240 REM --error routine, printer not ready--
250 PRINT CHR$(253);"PRINTER NOT READY,":REM sound buzzer
260 PRINT "PRESS 'START' TO TRY AGAIN...":PRINT
270 IF PEEK(53279)<>6 THEN 270:REM wait for start key
280 GOTO 100:REM try again
285 REM xxxxxxxxxxxxxxxxxxxxxxx
290 REM -- read a record from the disk file --
300 INPUT #1,ID$:REM get record number
305 IF ID$="ENDOFFILE" THEN 500:REM last record, go to closing routine
310 INPUT #1,NAME$,ADDR$,CITY$,STATE$,ZIP$,PHONE$:REM read rest of record
315 REM -- display the record --
320 PRINT CHR$(125);"DATA IN FILE";CHR$(29):REM (move cursor down)
330 PRINT "ID",ID$
335 PRINT "NAME",NAME$
340 PRINT "ADDRESS",ADDR$
345 PRINT "CITY",CITY$
350 PRINT "STATE",STATE$
355 PRINT "ZIP",ZIP$
360 PRINT "PHONE",PHONE$
370 PRINT "PRESS 'SELECT' TO PRINT RECORD..."
380 PRINT "PRESS 'START' TO READ NEXT RECORD..."
390 IF PEEK(53279)=6 THEN 300:REM get another record
391 IF PEEK(53279)<>5 THEN 390:REM check for select key
395 REM xxxxxxxxxxxxxxxxxxxxxxx
397 REM -- print record on printer --
400 PRINT #2;ID$
410 PRINT #2;NAME$
420 PRINT #2;ADDR$
430 PRINT #2;CITY$
440 PRINT #2;STATE$
450 PRINT #2;ZIP$
460 PRINT #2;PHONE$
470 GOTO 390:REM go wait for ok to read new record
480 REM xxxxxxxxxxxxxxxxxxxxxxx
490 REM -- closing routine --
500 CLOSE #1:CLOSE #2
510 PRINT CHR$(125):REM clear screen
520 PRINT "-- END OF PROGRAM --"
530 END
```

Designs for The Computer Home

by Paula Polley

Interior Designer John Allin Reveals A World of "Future Facts"



LIKE MOST OF US, John Allin has found the modern wall unit sorely lacking the space to accommodate the latest addition to high-tech consumer products—the microcomputer. Allin, who is Principal and Design Director of his interior design and architecture firm, *Life Designs*, has responded to this problem with a design solution of his own. On the walls of *Life Designs*' conference room hang eight-foot-high drawings of a gridded wall system designed to integrate the growing collection of electronic equipment we find in our offices and homes.

"Computers, disk drives, voice modulators, modems, monitors, video recorders, audio systems, encryption devices—even the new personal copiers—all will be integrated into this one wall unit," says Allin about his innovative solution to high technology's invasion of our home and work environments. But will Allin's wall, which he calls a "tech system," become as familiar to us as the conventional "Scandinavian wall unit" designed to house stereos, books, albums and objets d'art?

"Both," says Allin. "If an idea exists in the mind of a futurist or someone out there sketches it on a drawing board, it's a 'future fact'—well on its way to becoming an everyday reality in our rapidly changing world."

Indeed, Allin's "tech system" seems likely to become fact in the not-so-distant future. Already, in one of San Francisco's wharf-side warehouses, fabricators are working on a structural metal prototype of Allin's dream. Coaxial cables and color-coded wires are being hidden in well-organized and easily accessible troughs.

The ATARI CONNECTION talked to John Allin to find out how he creates electronic work spaces for large cor-

porations as well as for home offices. In the course of the interview, this prominent designer and futurist offered a sneak preview of some intriguing environmental "future facts" that may change our lives and living space as well.

CONNECTION: What more can you tell us about this "tech system" you're designing?

ALLIN: The project is one of several models for future interiors that *Life Design* has been contracted to develop. Right now, we're just completing the prototype for a system that will first appear in offices and then be modified for the housing industry. Born of the need to integrate technologies and reduce climbing fabrication and installation costs, this modular approach is finally coming of age.

CONNECTION: When will we actually be able to go out and purchase one of your "tech system" walls for our homes?

ALLIN: Sooner than you think. The "tech system" prototype already *looks* real, but the buttons don't work. Yet the planning and research—the design science—behind our prototype brings such advancements nearer to reality. It will be operational late next year.

CONNECTION: John, we've all seen how computers have affected our lives in terms of communications, work, and entertainment. But how will computers themselves affect the physical look and design of our homes?

ALLIN: We're literally living in two "Space Ages." First there's the outer space age—as reflected in our interest in and exploration of what's beyond. But there's also the minimal space age imposed on us by limited housing and work space left for us in a populous world.

As a designer, I deal primarily in creating the most space from a given set of dimensions. So my work is directly affected by the streamlining of computer hardware, like the Atari 1200XL. As computer components become smaller and more integrated, the look of our computer stations will change.

In fact, as computer electronics become fundamental to everyday life, the importance of their invisibility will disappear, just as with light switches and electrical sockets.

Think of the evolution that other technologies have undergone. Remember the first electric refrigerators? They were monolithic boxes with huge domes on top that housed large coils and whirring fans. Or the first stereo systems—remember how they invaded our living rooms all housed in chunks of Mediterranean turned-wood cabinetry, velvet speaker covers, and frilly wrought-iron grills? Now you can buy stereo components that take up no more space on your bookshelf than a dictionary—and perform with finer fidelity.

Just eight years ago, I remember seeing a model of a computerized home office of the future constructed by a San Francisco designer. The computer itself took up as much space as a large wardrobe closet, and it generated so much heat that you couldn't stay in the room for more than a few minutes. Now you can slip your computer into your briefcase and take it for a walk in the park—while you communicate electronically, verbally, and visually with distant terminals.

CONNECTION: You're saying that we can expect much greater flexibility in our future work spaces.

ALLIN: That's right. Your new office could be under an oak tree in a forest, atop a skyscraper in the city, or on a blanket at the beach.

With our computers, we're beginning to spend more time in the home—time we would otherwise spend in the office or at school. This will probably be acceptable for a while; but ours is a very mobile society of diverse interests and a unique sense of social independence. We're not accustomed to staying home hour after hour, day after day. That's why the ability to take our computers with us wherever we go will become a very important aspect of future living.

CONNECTION: Then integration of computer components will be as important as the streamlining of their overall design?

ALLIN: Integration will play a key role in all aspects of interior, industrial, and architectural design.

One of my favorite parts of ATARI CONNECTION is the "Home Computer Photo" section, showing readers' photos of their computer setups. Most of these setups have seven to twelve components, including disk drives and printers, modems and so forth.

I'm impressed by the creative job these people have done in organizing the elements of their systems in the most efficient way possible. But as the home computer industry settles down, we'll see more integration. We won't have to concern ourselves with designing cubbyholes for all those different components.

For example, my firm has already worked on plans for a new electro-chemical video screen that will eliminate the problem of bulky TV monitors. As thin as cardboard, these screens can be applied to different walls in your home or office and all tied into the same Central Processing Unit. This means that you'll be able to use the screen in your kitchen for recipes and the

Putting Together Your Home Computer Work Station—

Some Tips From Life Designs' Anthony Spelding

REMEMBER WHEN YOU FIRST brought your home computer home? There it was: your gateway to a new age of education, entertainment, and personal enrichment—and your escape from the tedium of information management.

Then you took the components out of their boxes, and suddenly your electronic "solution" was an interior design "problem." What room should you put your system in? Which corner would most easily triple as an effective educational area, a comfortable video-entertainment center, and a productive workstation? What about furniture? And temperature? And light? How could you arrange your components so that they'd all be within reach? While keeping that tangle of cords and cables out of sight?

If you're still struggling with such questions, you might find some guidance in the following suggestions from Anthony Spelding on setting up your home computer work station.

The Environment

Your computer should be in a cool (below 80 degrees F.), dry place with good ventilation. The room should be as free as possible of dust and smoke, which can affect disk drives and switches.

Keep your computer and disks or tapes out of direct sunlight and away from any other source of heat.

If you're purchasing carpeting for your computer area, make sure that it has static-controlling properties. Static

Photographed by Michel Nafziger

one in your family room for games, without having separate computer systems in each room.

CONNECTION: *Won't computer shopping influence interior design—or at least the kinds of items we purchase for our homes?*

ALLIN: For smaller products, like a set of dishes, yes—the choices will be much greater. But for larger products, like housing, vehicles, and appliances, the choices may not be as broad, but more modular.

We'll have the opportunity to customize our homes through off-site prefabrication and modular construction. We'll be able to select a plan, or even design our own plans for a home of the future using 3-D telecommunicated computer graphics. We'll choose the style of roof we like as well as the material covering it. All the information we need will be processed at the manufacturing site—which may itself be completely robotized. Every house—or sections of it—would be constructed to a broader range of individual specifications.

CONNECTION: *So we'll actually be able to act as our own designers and architects?*

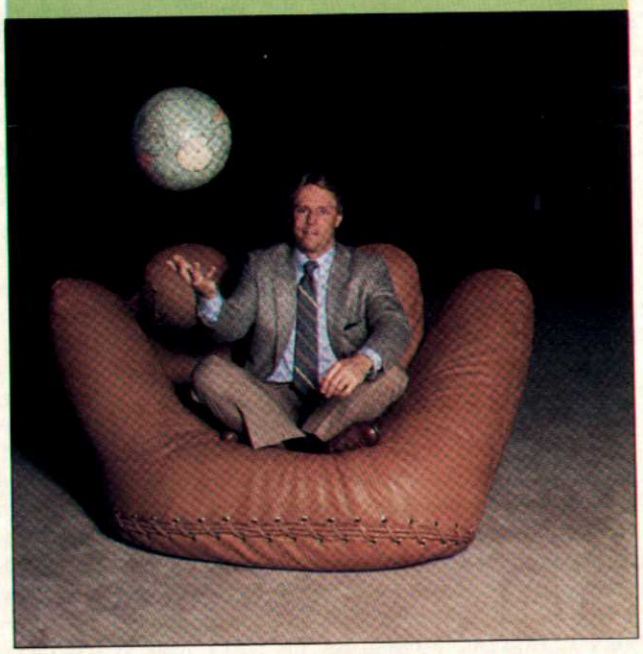
ALLIN: In a manner of speaking, yes—provided that the available choices match our expectations for individuality and affordability. We'll have access to professional-quality design solutions at a reasonable cost.

Since we'll be sharing information through the networking capabilities of telecommunications, resources that were once the exclusive domain of a given industry or profession will be available to more people who value selective advice.

CONNECTION: *How will telecommunications contribute to the introduction of new products in the marketplace?*

ALLIN: That's an interesting question. In the future, through electronic design and communication, it will actually be possible to introduce products that don't yet exist.

Your new office could be under an oak tree, atop a skyscraper, or on a blanket at the beach.



Let's say, for example, that you wanted to design, manufacture and market a new type of hair dryer. With computer-aided design, you could create three-dimensional prototypes, right on your monitor screen, for both the housing and the electronic circuitry of the hair dryer. Then you could test your computerized prototype for aerodynamics, durability, engineering, safety, and feasibility of construction. Through telecommunication networks, you could trace and price all the parts you'd need. Next, you could run a cost analysis to arrive at the most profitable way to manufacture your product.

Then, with the built-in "live audience" available through your home shopping network, you could survey prospective buyers and get instant feedback on the potential success of your product, before you'd even purchased a single part. If your market survey indicated a need to alter the design or re-evaluate the pricing, you'd just tap a few keys on your computer console to implement the changes.

And when you actually advertised your hair dryer in the computer-shopping marketplace, you wouldn't have to start making it until the orders started coming in.

CONNECTION: *Is your hair dryer analogy a projection or a "future fact?"*

ALLIN: Well, let's see. We've been talking for an hour. . . It's probably already a "future fact." ■

electricity can damage computer equipment—even erase stored information.

Furniture

Flexibility is the most important factor in arranging your computer station. If your equipment is too "locked in," it may be difficult to add components or rearrange your system later.

Modular, adjustable shelving provides the most flexibility—especially for an ever-changing, growing software library.

Allow some space behind your shelves or desk system to manage electric wires. Power strips provide the best solution to the management of cords and adaptors.

For security, a storage cabinet or set of file drawers provides a safe place for software and "out-of-sight" storage for hardware.

The surface on which your keyboard sits should be a standard 26 inches high.

Seating for work at the computer should be mobile and adjustable. A chair on a five-pronged base with casters is ideal. Woven fabric upholstery is much cooler and more comfortable than vinyl or leather.

Lighting

Images on a monitor screen can easily be obscured by glare and reflections from poorly placed lighting fixtures. Too much or too little light can also be a problem.

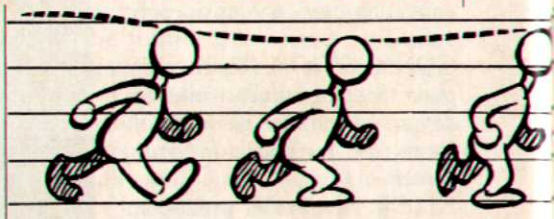
To check the current lighting around your computer, hold a mirror flush to the screen. The "hot spots" you see (from lamps, windows, or overhead fixtures) will be sources of glare when you're working at the keyboard.

A lighting source directly overhead and slightly behind the screen will help reduce glare and reflection. "Task-oriented" lights, like a drafting lamp, allow each user to control the position and amount of light in the immediate work area.

An alternative is to use three incandescent fixtures in and around your work station, creating a balanced triad of lighting sources. The idea is that the level of light over the entire area should be as even as possible.

—**Paula Polley**

From *Computer Animation Primer* by David Fox and Mitchell Waite. A BYTE Book



Cartoon Computer Animation

Create Your Own Character Sets! Make a Walking Man

by Mitchell Waite and David Fox

LAST ISSUE WE showed you how to animate a flying bird using the Atari Computer's built-in graphics set. Although our program was simple to implement and the bird easy enough to get flapping on the screen, it did take a certain amount of imagination to see a real bird flying. The problem is not that the Atari characters have too little resolution. Rather it is because the dots and lines that make up the Atari's built-in graphics character set are not bird wings, beaks, or feet. The actual resolution of each character is quite high, its just that we have the wrong pieces.

In this installment we'll show you how to make your own characters and get over the fixed design of the Atari character set. You'll learn how to redefine the Atari Computer character set to make your own detailed custom "alphabet." You will see how to make up your own individual characters and fill them with any pattern you want. Then by putting these characters together on the screen in the proper order you will make up a larger, more precisely detailed object. Using the same animation techniques we learned about in the first program, you'll make the new character set perform its tricks of movement for you.

THE CHARACTER SET—No Lack of Character in Atari

WHEN YOU FIRST turn on your Atari Computer you will see a word or words printed on the screen ("READY" if you are using your BASIC cartridge). What happens inside your computer to display those words? A series of number codes are placed in an area of RAM called screen memory, one code for each character. For example, the letter "A" would cause the code number 65 to be placed in screen memory. These codes are then interpreted in a predetermined way (depending on which Graphics Mode you are in). In the standard text mode, GRAPHICS O, the numbers in screen memory are translated as "addresses" which are used to look up some permanently stored information. This information, stored in "Read Only Memory," or ROM, is called a *character set*. Each character in the character set is composed of an array of dots (eight dots long and eight dots high), which can either be turned on or off. The information which defines one character is called the *character definition*.

Try typing some letters on your screen and see if you can make out the individual dots. If your television is sharp enough, you will be able to see them.

The information in each character definition is stored in a series of eight bytes in the ROM, with each byte representing *one horizontal row of eight dots*. Since there are eight bits in a byte, whether or not a bit is ON will determine whether the corresponding dot on the screen will be turned ON. Each character in the character set is defined in this manner.

There are 128 distinct characters in the Atari Computer character set. If we multiply this number of characters (128) by the number of bytes needed to define each character in the character set (8) we get 1024, or 1K ($128 \times 8 = 1024$ bytes). This is how much ROM space is needed to store the Atari Computer's built-in character set. Since each character can also be represented in reverse video, there are a possible $128 \times 2 = 256$ codes (from 0 to 255) which appear in screen memory and can be interpreted as characters. The codes 0 to 127 represent normal video characters and the codes from 128 to 255 are reversed for inverse video characters.

Roll Your Own Character Set

In many personal computers, the built-in character set is all you get. But the Atari Computer has the capability to display *user-defined character sets!* As we said, the ROM character set is permanent. You can't change any of the character definitions in this ROM. However, what if we were to create our own set of character definitions and POKE them into RAM? How would we let the computer know where to find this customized set of character definitions?

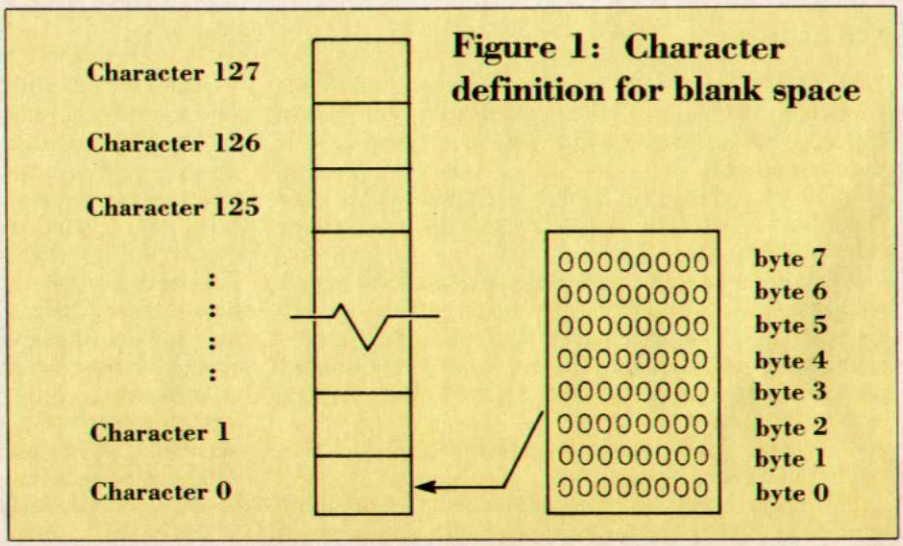
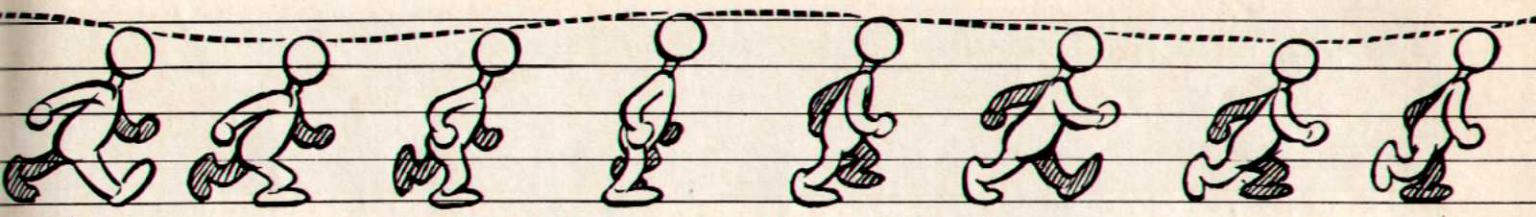


Figure 1: Character definition for blank space



The answer is simple: the Atari Computer has memory location 756 reserved for this program (unless otherwise noted, our values are in decimals). This RAM location always contains the "page" address at the start of the current character set. (A "page of memory" is 256 bytes; therefore, to convert a page address to an actual address, multiply by 256.) When you turn on your computer, press the SYSTEM RESET button, or change GRAPHICS modes, the value in 756 is automatically initialized to page 224 (the address of the ROM character set). Thus we say that 756 *points* to the character set in ROM. But you can change the value in 756 so it points instead to an address in the computer's RAM. By POKEing the page address of your customized character set into 756 you "turn on" the new character set. As we shall soon see, the results are instantaneously visible.

Let's try a simple program to see what happens if we simply change the page value in 756. Try typing in the following short program:

```
10 GRAPHICS 0
20 FOR I=0 TO 255
STEP 4
30 POKE 756,I: REM
Switch "character sets"
40 NEXT I
50 GOTO 20
```

Now try running this program. Don't worry, your computer isn't broken! What you will see on your screen is a rapidly changing, finely drawn display which fills the entire screen. Let's see exactly what's happening. When line 10 is executed, the screen is cleared. The Atari Computer clears its screen by filling screen memory with O's. These O's are used to look up the Oth character in the current character set. In the ROM character set, this Oth character is the space. A character set must always begin on a "1K boundary." This means it can begin at any address which is evenly divisible by 1024. Converting to pages, the value in 756 must be evenly divisible by 4 (there are 4 pages of 256 bytes in 1K). So in line 20 we increment I by four. As we POKE the different values of I in 756, the current "character set" is changed.

Of course, we really aren't switching to different character sets, just to whatever random information happens to be at that memory location. What we see on the screen is the current definition for the Oth character (the space character). Whatever happens to occupy the first eight bytes in each "character set" determines how the space character will be displayed. When the first eight bytes are O's, the screen goes blank.

Now press RESET and change line 30 by substituting the letter I with the value 200. The new line should read: 30 POKE 756, 200. The screen immediately becomes a mass of swirling, ever-changing interference patterns. But how could this be—there's no program running! Ah, but there is. We have discovered an address which is being used by the *Atari Computer Operating System*. It is changing the contents of the first eight bytes at machine language speed—a new pattern every 1/60th of a second.

Case of the Reserved Character Set

After we have designed a character set, we must find a safe place in memory for it. A good location is immediately below screen memory. Where is screen memory? In most computers, screen memory is always located at the same address. However, the Atari Computer automatically reserves space at the very beginning of RAM for screen memory. This means you'll find the screen memory at different addresses, depending on how much memory your computer contains and which Graphics Mode you are using. In GRAPHICS O, 1K of memory is used to display the screen.

The Atari Computer uses memory location 106 to store the number of pages of memory it thinks it currently has. Imagine the memory locations as "building blocks" stacked on top of one another. The first block is for screen memory and takes up 4 pages or 1K. The second block is for RAM character set and also takes up 4 pages. The third block is where we'll store our character set. By taking the value of 106 and subtracting 4 from it for screen memory and another 4 for the size of the character set, we can calculate the location of the third block for our character set.

Character Building

Now that we know the why and wherefore of user-defined character sets, we can create one of our own. There is a difficult and a not-so-difficult way to create user-defined character sets. The difficult way is the "manual method" as follows:

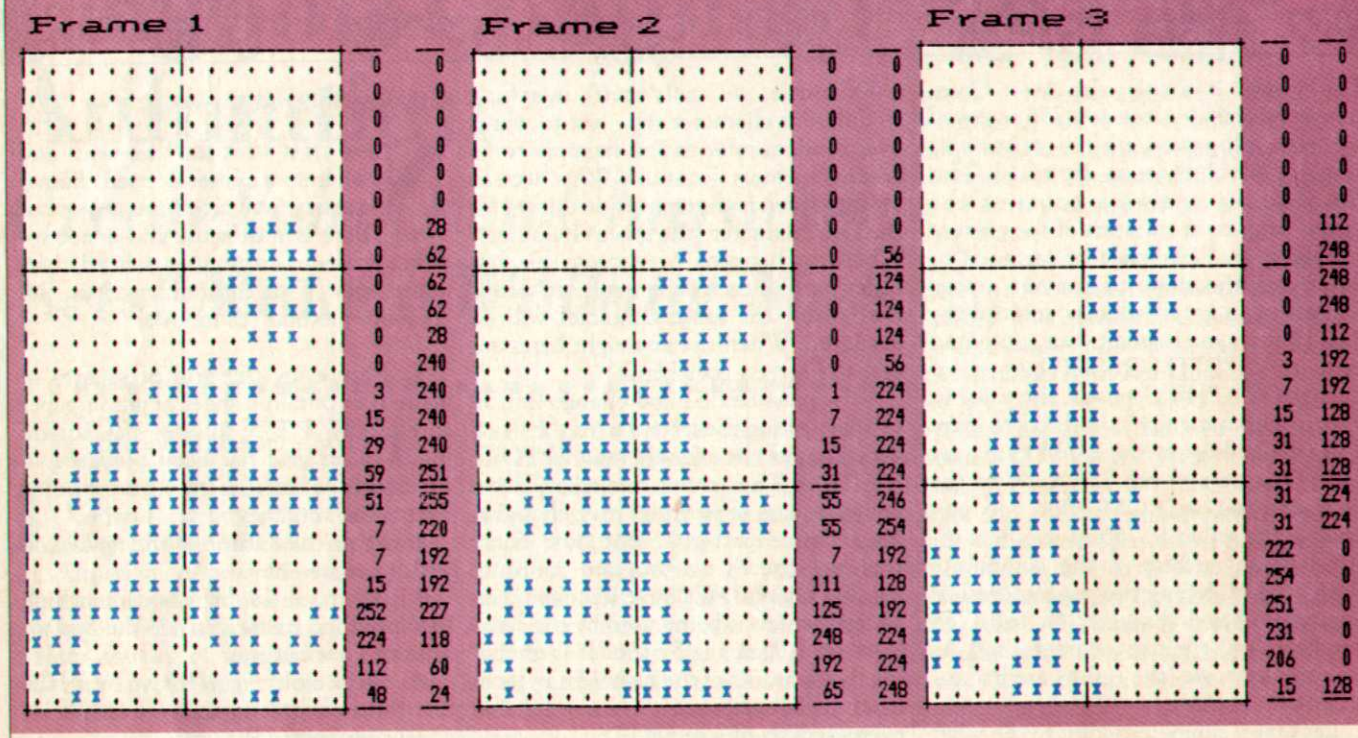
1. Photocopy the grid of squares in figure 6 or obtain a sheet of graph paper (preferably 3 x 3 squares per square inch). (Our grid has been prepared to accurately reflect the true proportions, 7:8, of each character.
2. Draw the outline of the figure you wish to represent.
3. Fill in all the little squares which lie more than halfway inside the boundaries of your outline.
4. Break your figure down into the 8 × 8-byte character cells (already done if you use our grid).
5. Calculate the decimal value of each row of each character cell using a binary 1 for a filled square and a binary 0 for an empty square.
6. Enter these byte values into your program.

This method is difficult in that it involves the manual transference of graphic information from paper to numbers in a program. If you only needed to do this once, it wouldn't be so bad. But, as we mentioned earlier, creating an effective animation requires a large degree of trial and error—it's exceedingly rare to get it right the first time. So once you've done all your work, tried the program and discovered that your animated figure looks as if it's critically ill, you must go through the entire process again.

A more efficient approach is to use one of the commercial "Font Editing" programs currently available to consumers as a software product that allows you to work with your characters in an interactive environment and lets you see the characters on the screen as you create and edit them. The word *font* refers to the style of the characters on the screen. You can design an Olde English font, a computer-like font, a script font, or even a walking-man font! The computer can also take care of the laborious calculations necessary to determine the

Walking man Reprinted From *ANIMATION* By Preston Blair Reprinted With Permission Of Foster Books, Tustin, Ca.

Fig. 2: Grid for Creating Character Set Figure



byte values for each character. You will still spend much time trying and erring, but the computer will handle much of the tedium.

Most interesting computer animations with a pleasing amount of detail are created by connecting several custom characters together to make up a larger picture. For example, we could have the letters A, B, C, D defined as graphics characters. Then we could arrange them with PRINT statements so they are sent to the screen like this:

AB
CD

The particular combination you pick is limited only to the possible combinations of those 4 letters. However, as the number of characters PRINTed out increases, so does the time to PRINT them, and therefore the fastest animation speed is lowered.

The secret to getting an animation sequence to work is the same secret for drawing the bird with flapping wings from the fixed graphics set. You construct a character set containing several frames that represent different degrees of movement of the object. Selecting the characters is tricky business. The best way to learn this is to try someone else's program—in this case, ours.

The Walking Man Program

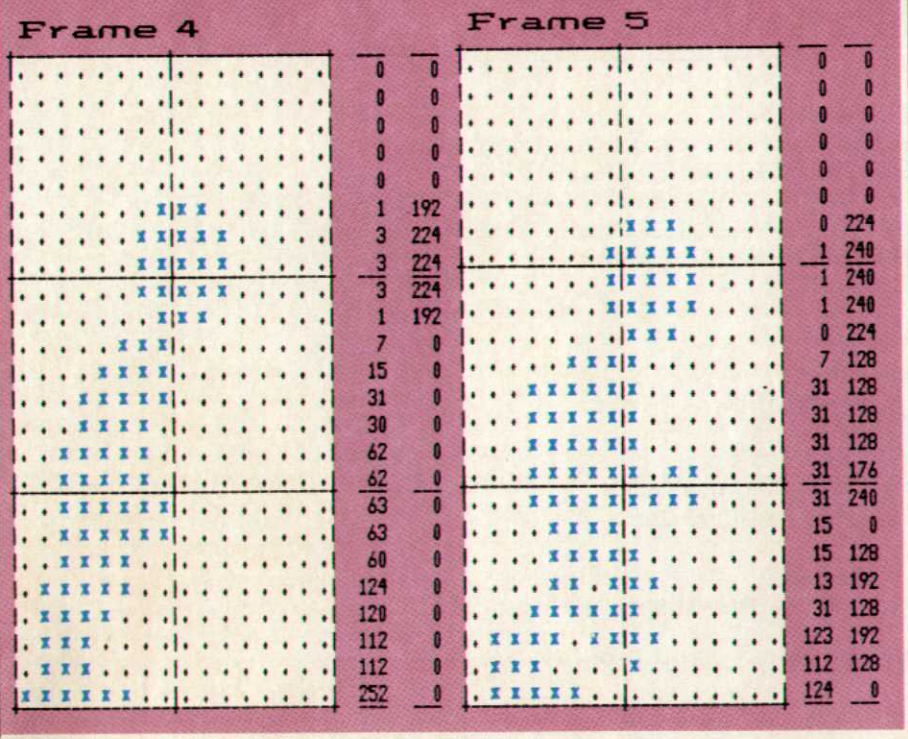
Our next program demonstrates the real power of user-defined character sets. We will define a character set which we then can use to draw a picture of a little man walking across the screen. Above are the character definitions for our "Man character set." Each frame is made up of six characters arranged in a 2 × 3 array (see figure 7). We are using five frames for the walking cycle, and each frame is displayed only once during each of his steps. This means that we need only 30 characters (6 characters per frame × 5 frames) to animate the man. Actually, we only need 26 characters since four of the characters which appear within the frames are blank. To the right of each frame are the byte values we need to POKE into the character set RAM:

Animation implemented on microcomputers is often considered crude. This is usually because the programmer is not an animator or an artist, *not* because the computer isn't capable of handling the job. Attention to detail makes animation come alive. Take a look at the position of the man's head in these five frames. As he walks, his entire body bounces up and down. This is much more realistic than a walking man with moving feet and stationary head!

To create this character set, *Animation*, by Preston Blair (published by Walter Foster Art Books, Tustin, California), a book on conventional animation, was used. This is an excellent yet simple book showing how to draw your own animated characters. We placed graph paper over a set of drawings from the book of a walking man and filled in the appropriate squares. A Font Editing program (*FONTEDIT* from the software package *IRIDIS 2* by The Code Works, Goleta, California) was used to help convert the filled-in squares to character set data. One technique you can try (if you can't find the figure you wish to animate in an animation book) is to cut out a drawing of your animated character from paper. At each of your figure's joints (i.e., knees, elbows), use paper clasps or string to create a hinge. Then position your figure in each frame and outline its shape onto graph paper. It will still take some practice to create smooth, realistic motion, but the proportions of each body part will be correct.

Exercise

Using a user-defined character set, write a program which shows a man walking across the screen. Use the joystick button to control his forward movement. Give him "life" with



a bounce in his step. Include the sounds of his footsteps.

The Walking Man Program has four main sections: the Initialization Section, the section which reads in the new character set, the actual Character Set Data, and the Animation Loop.

Initialization (lines 100-230). Each of the five "man" frames is made up of six graphic characters and six cursor control characters for a total of 12 characters. The variable FRMSZE (line 120) is set to this value. On our line 130 we reserve string space for our frames with the DIMension statement. All our frames are stored in one string variable, called MAN$, rather than a series of strings as we did in the Bird program (Example 1). This reduces the size of the program code needed to access a specific frame and makes the program more flexible if we want to use a different set of frame data. We could have initialized MAN$ in one statement, but it would have been much more difficult to understand and enter. The variable FRME$ will temporarily hold the current frame to make it easier to manipulate.

Make sure you enter lines 140-160 *exactly* as they appear, including the four spaces. Even though you are entering letters of the alphabet now, when we switch over to the new character set these will be printed as sections of the man.

ERASE$ (line 170) is used to erase the man every time he moves one character position to the right. If this wasn't used, our friend would leave behind a trail of old body parts as he moved across the screen.

In line 230, the color of the screen background and foreground is changed.

In line 240, the address of the new character set is POKEd into memory location 756. This is done to turn on the new character set and turn off the old Atari Computer character set.

Set up Alternate Character Set (lines 8000-8150). Here the new character set is POKEd in RAM. First, on line 8010, some memory is set aside for our character set. Recall that address 106 is where the Atari Computer stores the number of pages of memory it thinks are in the computer. We've set the variable HICHRB (High byte of CHARACTER set Base) to the total number of RAM pages in the computer minus eight pages (2K), four pages for screen memory and four pages for the character set.

In line 0020, the RAM page number HICHRB is converted to an actual RAM address by multiplying it by 256 (number of bytes in a page) and then stored in CHRBAS.

The next step is to read in the character set data. The first letter of the character set will replace the lower case "a," the second letter, lower case "b," etc. In some programs, you may need to copy the ROM character set into our RAM character set.

In line 8040, the OFFSET for the lower case "a" (number of bytes from the beginning of the character set) is calculated and the number of characters we are redefining is stored in CHARS.

To assure you enter our character set data accurately, a "checksum" value is used. To obtain this value, all of the bytes in our data statements were added together. Then this value, which came to 16845, was placed in a DATA statement on line 20020. This checksum value is READ into the variable TOTAL (line 8060) and all the bytes in our DATA statements are added together and stored in TEMP as the character set is READ and POKEd into RAM (lines 8070-8090). If the checksum value in TOTAL doesn't equal the calculated sum in TEMP, an error message is printed out. If this happens, recheck the values typed into the character set data statements.

On lines 8120 through 8140 the first character in the character set is filled with O's. As stated before, this is the character definition for the space character. You already know what kind of designs can appear on the screen if the space character isn't a blank!

Character Set Data (lines 20000-20380). This is where the data on our walking man is stored. As previously mentioned, the first value (16845) is the sum of the rest of data. Each line, starting with 20050, contains one character definition—the eight bytes which define a single character.

Animation Loop (lines 300-420). The logic behind this section is similar to the animation loop in the Bird program though a few new techniques have been added. Since all the frames are stored in one long string, the desired frame can be pointed to directly with the formula in line 330. In Atari BASIC, a "substring" (section of a string) can be accessed by indicating the first and last characters:

STRING$(first, last)

The formula in line 230 allows access to the "Ith" substring of MAN$ which is FRMSZE characters long. When I equals 1, the first 12 characters of MAN$ are stored in FRAME$ (Frame 1). When I equals 4, the fourth set of 12 characters is stored in FRME$ (Frame 4).

Walking Man Character Set

by David Fox and Mitchell Waite

```
 10 REM
 20 REM
 30 REM
 40 REM
 50 REM Copyright (C) 1982 by David Fox and Mitchell Waite
 60 REM
100 REM Initialize
110 FRAMES=5: REM Number of frames
120 FRMSZE=12: REM Characters in frame (including cursor control chars)
130 DIM MAN$(FRAMES*FRMSZE),FRAME$(FRMSZE),ERASE$(7)
140 MAN$="ba{DOWN}{2 LEFT}bc{DOWN}{2 LEFT}de f{DOWN}{2 LEFT}gh{DOWN}{2 LEFT}ij"
150 MAN$(25)="bk{DOWN}{2 LEFT}lm{DOWN}{2 LEFT}nopq{DOWN}{2 LEFT}rs{DOWN}{2 LEFT}t "
160 MAN$(49)="uv{DOWN}{2 LEFT}wx{DOWN}{2 LEFT}yz"
170 ERASE$=" {LEFT}{UP} {LEFT}{UP} "
180 GRAPHICS 0
190 POKE 752,1: REM Turn off cursor
200 PRINT "One moment please..."
210 GOSUB 8000: REM Read in Character Set
220 PRINT "{CLEAR}"
230 SETCOLOR 1,0,14:
    SETCOLOR 2,1,2
240 POKE 756,HICHRB: REM Switch to new Char Set
250 REM
300 REM Animation Loop
310 X=3: REM Set starting horizontal position of Man
320 FOR I=1 TO FRAMES
330   FRAME$=MAN$(I*FRMSZE-(FRMSZE-1),I*FRMSZE)
340   POSITION X,14:
      PRINT ERASE$;FRAME$;
350   IF I=1 THEN
        SOUND 1,4,0,14: REM Footsteps
360   IF I=2 THEN
        SOUND 1,4,16,14
370   SOUND 1,0,0,0: REM Turn off sound
380   FOR W=1 TO 10:
      NEXT W: REM Slow him down a little
390 NEXT I
400 REM Walk man across screen if Joystick button is down
410 IF STRIG(0)=0 THEN
      X=X+1:
      IF X=36 THEN
        PRINT "{CLEAR}":
        GOTO 310
420 GOTO 320
430 REM
8000 REM Set Up Alternate Character Set
8010 HICHRB=PEEK(106)-8: REM Reserve memory space (1024 bytes) below screen
8020 CHRBAS=HICHRB*256: REM Find start of Character Set
8030 REM Read in data, skip first 97 characters
8040 OFFSET=97*8:
     CHARS=26
8060 READ TOTAL:
     TEMP=0
8070 FOR I=CHRBAS+OFFSET TO CHRBAS+OFFSET+CHARS*8-1
8080   READ BYTE:
       POKE I,BYTE:
       TEMP=TEMP+BYTE
8090 NEXT I
8100 IF TOTAL<>TEMP THEN
       GRAPHICS 0:
       PRINT "ERROR In Character Set Data":
       END
```

```
 8110  REM Clear out first char (background)
 8120  FOR I=CHRBAS TO CHRBAS+7
 8130     POKE I,0
 8140  NEXT I
 8150  RETURN
 8160  REM
20000  REM Character Set Data
20010  REM . Checksum
20020  DATA 16845
20030  REM
20040  REM . Frame 1
20050  DATA 0,0,0,0,0,0,28,62
20060  DATA 0,0,0,0,3,15,29,59
20070  DATA 62,62,28,240,240,240,240,251
20080  DATA 51,7,7,15,252,224,112,48
20090  DATA 255,220,192,192,227,118,60,24
20100  REM
20110  REM . Frame 2
20120  DATA 0,0,0,0,0,0,0,56
20130  DATA 0,0,0,0,1,7,15,31
20140  DATA 124,124,124,56,224,224,224,224
20150  DATA 55,55,7,111,125,248,192,65
20160  DATA 246,254,192,128,192,224,224,248
20170  REM
20180  REM . Frame 3
20190  DATA 0,0,0,0,0,0,112,248
20200  DATA 0,0,0,3,7,15,31,31
20210  DATA 248,248,112,192,192,128,128,128
20220  DATA 31,31,222,254,251,231,206,15
20230  DATA 224,224,0,0,0,0,0,128
20240  REM
20250  REM . Frame 4
20260  DATA 0,0,0,0,0,1,3,3
20270  DATA 0,0,0,0,0,192,224,224
20280  DATA 3,1,7,15,31,30,62,62
20290  DATA 224,192,0,0,0,0,0,0
20300  DATA 63,63,60,124,120,112,112,252
20310  REM
20320  REM . Frame 5
20330  DATA 0,0,0,0,0,0,0,1
20340  DATA 0,0,0,0,0,0,224,240
20350  DATA 1,1,0,7,31,31,31,31
20360  DATA 240,240,224,128,128,128,128,176
20370  DATA 31,15,15,13,31,123,112,124
20380  DATA 240,0,128,192,128,192,128,0
```

On line 340, the cursor (now invisible) is positioned on the screen. ERASE$ is used to clear away any of the previously drawn man, and then the current frame is drawn. On lines 350 to 370, the sounds of a footstep are added. One sound is made when the man's heel hits the ground and another when the rest of his foot makes contact.

Finally, on line 410 the man's horizontal position on the screen is incremented if the joystick button is pressed. The screen is cleared when he reaches the right edge of it, and the starting horizontal position (X) in line 310 is reinitialized.

Running the Program

Before you RUN the program, plug a joystick into the first joystick port (on the left). Now type RUN and you'll see the man walking in place on the left side of the screen. No mistaking him for a bunch of wobbly pick-up sticks—he really looks like a walking man! Adjust the volume of your television set so you can hear the footsteps. When you press the joystick button, the man will begin walking eastward. The next problem to overcome is BASIC's slowness when it comes to animating more than one figure at the same time Next time we'll see how this can be accomplished without the use of machine language. ■

SPECIAL LISTING INSTRUCTIONS

1. All inverse video characters (characters entered after pressing the "Atari Key" —light background and dark letters instead of dark background and light letters) will be underlined. In the following example, the letters C, E and F should be entered in inverse video:

S$="ABCDEFGHI"

2. Control characters (those entered while holding down the Control button) will be surrounded by curly brackets { }. All of the Atari's graphics characters are accessed while depressing the [CTRL] key. In the following example, the letters B, G and H are control characters:

C$="A{B}CDEF{G} {H} IJ"

3. Special cursor and screen keys will be represented by printing the name or description of the key within curly brackets { }. To enter these special keys into a string, you will need to press the ESC key first. This puts the code for the key into the string instead of actually carrying out the action. In the following example, we want to clear the screen on line 100. To do this, first press the ESC key, then hold the shift key down and press the key with the word CLEAR on it (it has a < on it). When the line is executed, the screen will clear:

100 PRINT "{CLEAR}"

4. In the next example, the cursor key with the arrow pointing down is used. When this line is executed, the computer will print the word "HI." Now move the line down one line and then print "BYE." To enter this character, first press the [ESC] key, then hold the [CTRL] key down and press the key with the down arrow on it. Here's an example of what this program line looks like in our listing:

110 PRINT "HI {DOWN} BYE"

When executed, you'll see the following on your screen:

HI
BYE

5. When a number appears before a curly bracketed word, it means we want you to enter that character the number of times indicated. In the following example, we ask you to enter "ABCDE," then move the cursor down one space, then five spaces to the left, and finally the letters "FGHIJ":

120 PRINT "ABCDE [DOWN] [5 LEFT] FGHIJ"

When this line is executed, you will see the following on the screen:

ABCDE
FGHIJ

6. When spaces are important to an animation, as they are in the program, we will represent a space with a lower case b that has a slash through it:

ƀ

As before, if the b character is underlined, enter the space as an inverse video character.

When the Halo Begins to Fade

How to Survive Computer Boredom

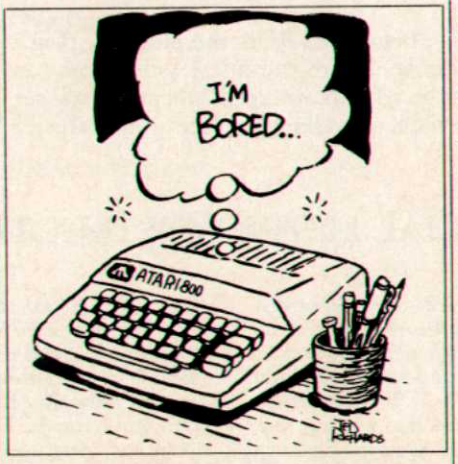
by Herb Kohl

I REMEMBER WHEN WE first got our Atari 800 Computer system about two years ago. For the first few weeks we kept it busy all the time. My son Josh and his friends flew video rocket ships, firing mazers, phasers and lazers for hours at a time until they seemed to be scoring in the millions and billions. I spent nights teaching myself BASIC and even occasionally trying to develop hand-eye coordination with a joystick so I wouldn't embarrass myself in front of the 11-year-old gamemasters who monopolized the computer in the afternoon.

My wife and two daughters felt very differently about the computer. They didn't see the halo Josh and I did. For them the computer was an alien, dehumanizing object they wanted no part of. In our house, at least the computer was male territory despite the fact that Josh and I tried to persuade them to use the Atari 800 and felt guilty about the apparent male chauvanism the computer created in our family. Separating our activities into male and female domains was counter to our beliefs and everyday lives.

After several months, the glow began to fade for Josh and me too, and the computer sat on a desk in our living room, a neglected, forlorn-looking objective whose function was almost purely decorative. The games we had were boring, and new games were too easy to figure out. I learned BASIC and tried my hand at programming a few games and doing a little home management, but found a paper and pencil more convenient and portable for most of the simple tasks I tried on the computer.

Then all of a sudden some interesting things began to happen. I, for example, spent a lot of time thinking about how I could use a computer even though I didn't touch it for weeks. It occurred to Josh that the computer might be useful to help him organize his baseball card collection. His idea was that he could store information about each card so that by pressing a key he could get a list of all the shortstops he had for 1976, or a list of all the Pittsburgh Pirates he had from 1965 to 1977. He also wanted to get information like who was the National League's highest batter in the Seventies.



Initially, Josh's challenge seemed easy. I set out to create a program for him that would accept all of his data and organize it in most any way. I knew nothing about creating database programs or how to store and retrieve information from a diskette. The problem was wonderful because it allowed me to return to the computer with a problem to deal with—one that I never solved. Josh discovered, before I got very far, that he could use the Atari Home Filing Manager Program, which I use to file our recipes, to organize baseball cards by team, position and batting average. Nevertheless, Josh's imagination inspired me. I began to think of ways the computer could be used to store notes for my books and writing projects as well as organize research information that suited my very idiosyncratic working habits.

At the same time Erica, Tonia, and my wife Judy were also thinking about how to use the computer. Even though the machine sat unused it was an active presence in all of our minds. Judy, who is a weaver, had toyed with the idea of using the computer as a design tool for weaving. Now a year and half later, we are composing a weaving design program for the Atari Computer Camps. The specifications for the program come from Judy's weaving experience and knowledge—she has begun to see the halo that had faded for Josh and me. Tonia, my oldest daughter who is interested in painting, might begin to explore some of the marvelous graphics programs that have been published recently, such as Reston's *Paint* program and Datasoft's *Graphics Master*. Erica, who is interested in biology and brain physiology, has also said that she's interested in exploring the relationship between computer programs and brain functioning.

If the computer no longer has a halo for Josh and me, it *has* begun to assume a more comfortable and ordinary role in our lives, much like that of pencils and telephones. I have begun to explore different languages and have become intrigued with the educational possibilities of Atari PILOT and LOGO. I've also become part of a computer telephone network and look forward to reading my electronic mail as much as I do to opening letters and packages. Josh has moved from playing games to making games. Recently we have been working together on developing a simple adventure game form using Atari PILOT that will make the structure of adventure games clear to beginning users and enable them to create their own games.

The Atari PILOT program listing on the next page is a version of the basic game. It involves a monster and two children and has a very short vocabulary list: Smile, Frown, Cry. The game can be changed, the vocabulary expanded, the graphics enhanced.

Our Atari 800 tends to be pretty busy these days. However, we still have times when it is little more than decoration. Once the initial aura wears off, the computer can become a part of your life but it's no substitute for all the other wonderful things one can do. Josh still plays baseball, Tonia paints, Erica likes to take photographs and Judy weaves. And despite all of the word-processing programs I've tried, there's nothing like the feel of writing with a fountain pen. ■

Herb's Adventure

To get started on *Herb's Adventure* simply type each program line into your computer exactly as it's listed. But, you'll find a funny-looking character on lines 230 and 590 that looks like this: }. This is a special *control character* that *CLEARs* the screen after you've completed each question-and-answer sequence in the program.

To make this special *CLEAR control character,* first press the [CTRL] button down, hold it down and then press down the [ESC] button. Now let go of those two buttons. Next press the SHIFT key, hold it down and press down the CLEAR button. You'll now see a funny-looking arrow pointing up and to your left on your screen. This CLEAR control character won't print on most printers; instead it's printed as an odd-looking bracket.

```
10 T:
20 T:
30 T:WHAT IS YOUR NAME?
40 A:$NAME
50 T:
60 T:$NAME DO YOU WANT AN ADVENTURE?
70 A:$ANSWER
80 M:YES
90 TY:WELL HERE WE GO...
100 JY:*MONSTER
110 TN:SORRY. MAYBE SOME OTHER TIME.
120 E:
130 *MONSTER
140 T:
150 T:    YOU ARE TAKING A WALK ONE DAY
160 T:AND MEET A MONSTER.  IT COMES UP TO
170 T:YOU AND BREATHES SMOKE IN YOUR FACE.
180 T:IT SAYS, "WHAT ARE YOU DOING
190 T:BOTHERING ME?"
200 T:
210 T:    WHAT DO YOU DO?
220 U:*VOCAB
230 T:}
240 M:CRY
250 JY:*CRY
260 M:SMILE
270 UY:*SMILE
280 M:FROWN
290 UY:*FROWN
300 J:*MONSTER
310 *SMILE
320 T:
330 T:
340 T:
350 T:    YOUR SMILE HAS MELTED MY
360 T:HEART. I WILL LET YOU START THE
370 T:ADVENTURE AGAIN.
380 E:
390 *FROWN
400 T:
410 T:
420 T:
430 T:    YOUR FROWN HAS ANGERED ME.
440 T:YOU BETTER RUN OR I'LL EAT YOU.
450 T:BUT I'M KINDER THAN YOU THINK,
460 T:SO YOU CAN START ALL OVER AGAIN.
470 E:
480 *CRY
490 T:
500 T:
510 T:
520 T:    I AM A FEROCIOUS MONSTER, BUT
530 T:I CAN'T STAND CRYING...HERE
540 T:TAKE MY TREASURE.
550 U:*TREASURE
560 T:TO START AGAIN, TYPE HELLO
570 A:$HELLO
580 M:HELLO
590 T:}
600 JY:*MONSTER
610 E:
620 *TREASURE
630 T:
640 T:
650 T:
660 T: R:  You can make any treasure
670 T: R:you want by using the Atari
680 T: R:Control Characters just like
690 T: R:you would use a set of
700 T: R:building bricks.
710 T:
720 T:
730 T:
740 T:
750 E:
760 *VOCAB
770 T:
780 T:HERE ARE THE WORDS YOU CAN USE:
790 T:
800 T:CRY,SMILE AND FROWN.
810 T:
820 T:WHAT WILL YOU DO?
830 A:
840 E:
```

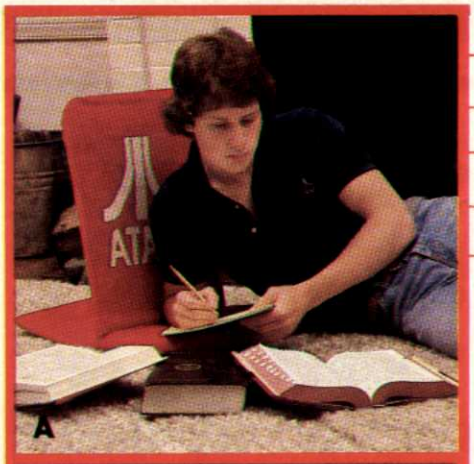

NEXT ISSUE:

Computer Education Special

- **Interview with Logo Creator, Seymour Papert**
- **Atari Logo and PILOT Programming Section**
- **A Look at Educational Software—What's Best for the Class and Home**

Plus

A Guide to *Free* Public-Domain Software
How to Design and Program Your Own Arcade-style Computer Game!
And More Computer Animation



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

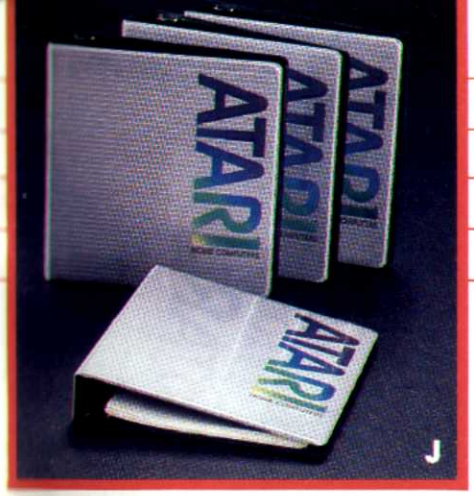
G **RUGBY SHIRT** (dark blue, maroon) S-M-L-XL. #AC119 $14.95

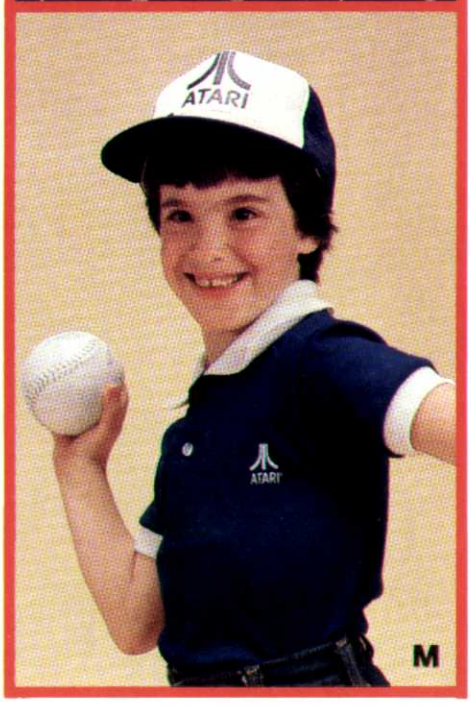
H

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. #ACI02 $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

THEY LAUGHED WHEN I SAT DOWN AT THE KEYBOARD

Squeals of derision rang through the room. "You, program a computer?" someone asked incredulously. "Now I've heard everything!"

"Enjoy your laugh, beetface," I thought. "You won't be chuckling for long." Little did they know I had MICROSOFT BASIC II, the powerful programming language that uses simple English commands.

I slipped the potent little cartridge into my ATARI Home Computer and closed the door with a confident slap. In a very short time, my friends were astounded at my programming prowess. Information, sounds, colors — even player-missile graphics — leapt across the screen. True, at one point I did have a little bug in a program, but MICROSOFT BASIC II's debugging features helped me correct it easily. I finished my *tour de force* by typing in a program written in another computer's MICROSOFT BASIC dialect.

Oohs and ahs filled the air. "Top drawer," snapped the Colonel. "What a man," Mimi cooed. MICROSOFT BASIC II and I had won the day.

ATARI®
A Warner Communications Company

Made in U.S.A.
Atari, Inc., Home Computer Division
P.O. Box 61657, Sunnyvale, CA 94086

C017668-10