Quality Computers ™

CELEBRATING THE APPLE II

II ALIVE

CELEBRATING THE APPLE II

Dear II Alive Subscriber,

We appreciate you as a customer, and as a fellow-supporter of the Apple II. To show our appreciation, we would like to offer you the following savings...

One coupon per order.

$3 OFF	$5 OFF
YOUR NEXT ORDER OVER $50	**YOUR NEXT ORDER OVER $100**
EXPIRES 8/31/94	EXPIRES 8/31/94

Thanks!

FOR YOUR CONTINUED SUPPORT

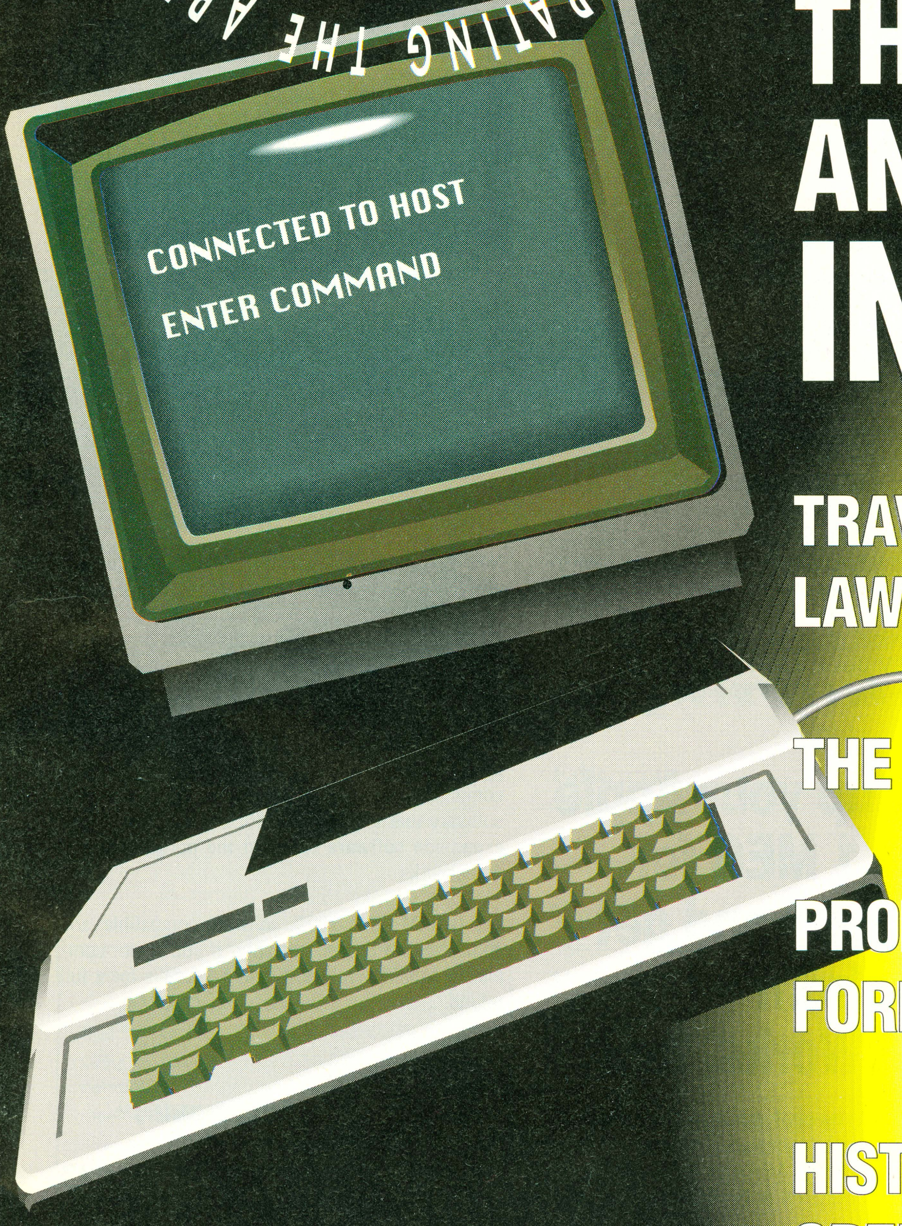
Quality Computers™

A QUALITY COMPUTERS PUBLICATION
VOL. 2 NO. 3 - JULY/AUGUST 94
PRICE $3.95

CELEBRATING THE APPLE II

II ALIVE

CELEBRATING THE APPLE II



THE APPLE II AND THE INTERNET

TRAVELING SALESMEN AND LAWN SPRINKLERS

THE APPLE II AND AUTISM

PRODOS BUILDING BLOCKS: FORKED FILES

HISTORY OF THE APPLE II OPERATING SYSTEM

Q-RAMS

30 Times Faster & Less Disk Swapping

ADD REAL POWER TO YOUR COMPUTER!

Power = Speed + Convenience

The only IIGS RAM card that gives you the power of a RAM disk... *automatically!*

FOR VALUE AND POWER YOU CAN'T BEAT THE Q-RAM GS2

When we could no longer find reliable, affordable hard drives, we made our own—the Q Drive. The Q Drive made Apples faster and easier to use. In fact, it was such a success that we had to wonder why we'd waited so long. Now we've done it again.

The Q-RAM GS2 is an economical way to add 4 MEG of memory to your Apple IIGS. It's fully compatible with all Apple IIGS hardware and software—including the IIGS RAM Disk and DMA peripherals like the Apple II High Speed SCSI Card.

The 4 MEG Q-RAM GS2 is not only affordable, but comes with FlashBoot free! FlashBoot lets you create a super fast, super convenient RAM disk on your Apple IIGS.

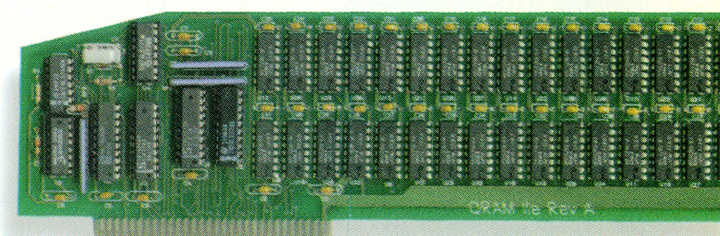
When you install the Q-RAM GS, life becomes just a lot easier. Programs load completely into memory at startup, eliminating disk swapping. Out-of-memory error messages disappear. You'll have memory to spare—memory to load desk accessories, or to set up a convenient RAM disk.

The Q-RAM GS2 installs easily, replacing your original 1 MEG. Apple memory board, or any other underachieving memory card you may own. But don't worry about getting stuck with a left over board that you can't use. Ask your sales rep about a RAM card trade-in. It is a terrific way to recycle your old card and save money at the same time!

Of course, since you're dealing with Quality Computers, you get an unconditional 30-day money-back guarantee and a five-year warranty. And the price is the best news of all—a 4 MEG Q-RAM GS2 costs about the same or even less than other IIGS memory cards in a 1 MEG configuration!

FLASHBOOT FREE WITH Q-RAM GS2

As an added bonus, when you buy a 4 MEG Q-RAM GS2, you get FlashBoot free. FlashBoot lets you quickly save and load the contents of a RAM Disk. What is a RAM Disk? Every Apple IIGS has a built-in RAM Disk capability that lets you reserve some of your computer's memory as a super-fast electronic disk drive. Set up your RAM Disk in the morning and you might not have to swap program disks all day! You can discover the speed and convenience of a RAM Disk with FlashBoot. FlashBoot offers several flexible options to boot the RAM Disk and the other drives attached to your computer, and makes loading your RAM disk easy.



4 MEG Only

$199^95

Beef up your Apple IIe to 1 MEG for only $99^95 and get more from AppleWorks, too!

Snap the Q-RAM IIe into your Apple IIe, and you'll create a whole new computing experience. Programs load completely. A lot less disk swapping. A lot more productivity. All of AppleWorks, plus your TimeOut applications will completely load into RAM. That means you can concentrate on working—not waiting for disk drives. Plus with a huge 1 MEG treasure trove of RAM, you can create huge documents, and not run into a single "Out of Memory" message.

Imagine students in your school's IIe lab doing more learning and less waiting. Imagine the joy of starting AppleWorks, completing a project, and never having to go back to the disk drive. It can happen with the Q-RAM IIe.

Now imagine all of this convenience and productivity at a price 1/2 of what other 1 MEG cards for the IIe cost. It's true, the new Q-RAM IIe costs only $99.95. At this price, the savings can really add up. You can upgrade your IIe lab and save $100 per computer. Now you can afford more software.

The Q-RAM IIe replaces your IIe's 80 Column Card or Extended 80 Column Card, and is 100% software compatible. The Q-RAM IIe comes with diagnostic software that test the card for peace of mind, and expansion software to boost the performance of AppleWorks.

The Q-RAM IIe is 100% software compatible, and comes with a 5 year warranty. If you're not completely satisfied, return it within 30 days for a full refund.

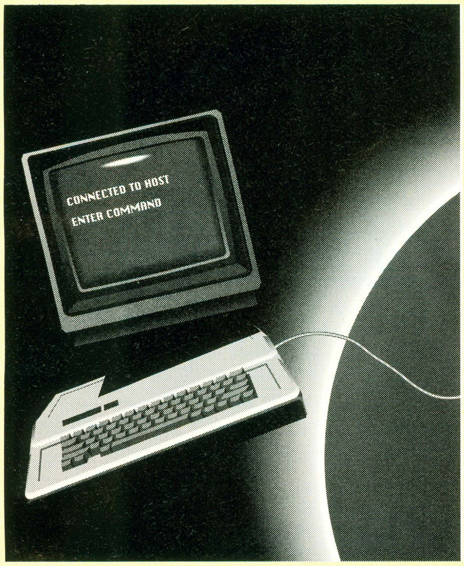
Quality Computers™

1-800-777-3642

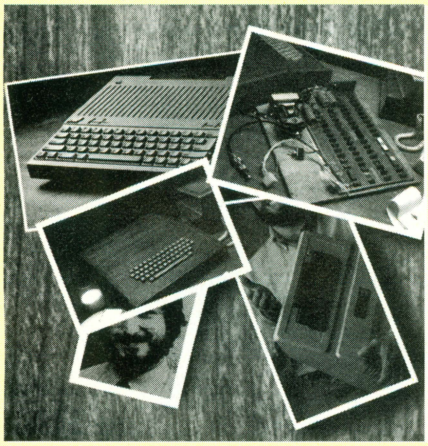
20200 Nine Mile Rd. • St. Clair Shores, MI 48080 • 810-774-7200 • FAX 810-774-2698

in this issue

features



Page 24



Page 40

II Alive is published every other month, for $19.95 per year by Quality Computers Inc., 20200 Nine Mile Rd., St. Clair Shores, MI 48080-1791. Application to mail at second class postage rates is pending at St. Clair Shores, MI and additional mailing offices. POSTMASTER: send address changes to II Alive Circulation, P.O. Box 349, St. Clair Shores, MI 48080-0349.

departments

NET OF DREAMS

JERRY KINDALL

EDITOR

You've probably heard a lot about the Internet lately. Whether it's referred to simply as "the Net" or the "information superhighway," there's nothing in recent memory that has so struck the public's fancy. The very term "information superhighway," a buzzword designed to capture the interest and imagination of the generally non-computer-literate masses, is adequate proof of that.

The National Information Infrastructure (NII) is the proposed superhighway. The Clinton administration envisions a high-speed fiber optic network which is as accessible to every U.S. citizen as a telephone line. Orders of magnitude faster than even the fastest modems, the NII would be capable of transmitting the entire text of this magazine in a matter of seconds—and of transmitting audio, video, computer software, and other forms of data.

But the NII is a delivery system—a medium. It doesn't provide content. Luckily, the Internet (which is actually a conglomeration of information resources, network systems, and communications services provided by over 13,000 independently-operated networks) has content to spare. While telephone and cable companies will undoubtedly participate in and utilize the NII to bring us communications and entertainment services, the most important content carried on the NII will undoubtedly be the Internet—or whatever it evolves into next week. Content created not by Hollywood studios, but by normal people like you and me.

As I've said many times in our Modem Nation column, telecommunications isn't about computers talking to other computers. It's about people talking to other people. And the Internet is, above all else, a whole lot of people. Millions of them. And more are joining every day. Some Internet service providers report a 10% per week growth rate. (The whole Internet isn't growing at this rate—just parts of it—but if it were, it would double in size every nine weeks!)

The Internet is such a vast resource that there's literally something for everyone. Even us Apple II users. When there are literally billions of megabytes of programs, files, and discussions available on the Net, even the fraction of a percent dedicated to the Apple II represents a real goldmine. Although the things you may have heard about the Internet—it's confusing, it's not user-friendly, it's not at all organized, it's full of heated emotional discussions with little worthwhile information content—are all true, they're also irrelevant. The good stuff, when you finally track it down, is as good as the bad stuff is bad.

That's why, in this issue, we asked M. G. Roberts to take us on a tour of the Internet. Tell us what it is, we said, tell us what's out there, how to get connected, and how to find what we're looking for. The resulting article was so large we had to split it into three parts. (Not too surprising, considering that I recently read a 700-page book which considered itself a "decent starter guide" to the Net.) You'll find the first part of our Internet series in this issue.

The time is rapidly approaching when having Net access will be as important as having a telephone, especially for those of us with interest in things like the Apple II. Even after the last dollar has been wrung from the commercial Apple II market, you can bet that you'll still be able to find Apple II owners on the Internet. So, if you don't have a modem, now's the time to get one. 14,400 BPS modems can be had for under $200, but if you can't afford that, 2400 is better than nothing.

While getting an account on a major online service such as GEnie or CompuServe will give you an Internet e-mail address, it won't get you full access to everything the Internet has to access. For that, you need an account on a "live" Internet site. I've been a GEnie user (and abuser) for more than six years, but Roberts' article—and the aforementioned 700-page book—has convinced me that it's time to supplement my GEnie account with a full Internet account. In the coming months, as Roberts' column unfolds, you'll find out why and how you should do the same.

And Another Thing...

Ellen Rosenberg, who has been acting as managing editor for the last several issues, will no longer be performing that service starting with the next issue. I'm still looking for a replacement—it's possible I'll end up editing the magazine myself again—but I do want to publicly thank Ellen for her outstanding efforts. Ellen will continue writing our Shareware Spy column, which had to take a break this issue but which will return next time. Thank you, Ellen! ■

NOISETRACKER REVISITED

Dear *II Alive,*

I feel you left out an incredible feature of *NoiseTracker* in the Music Composition section of the Sound Logic II article. It's true that it lacks the editing capability of *Soundsmith,* but it has one unique feature—RAM-based tracks. This was one of the first programs to implement DOC RAM swapping (storing sounds in normal RAM, and swapping them in and out of the sound chip's memory as needed, to allow sounds longer than 64K to be played). Using *NoiseTracker,* it's possible to create a song using sounds that completely fill your system's memory.

I've logged in over 1200 hours composing in *NoiseTracker,* and it's become a program I could not survive without. The brief mention of the program in the magazine did not do it justice.

Noah Mittman
@panix.com

APPLE II MEMORIES

Dear *II Alive,*

I read with interest the editorial on editor Jerry Kindall's first exposure to the Apple II. Back in 1983, at the age of 65, I decided I didn't want to be left out of the computer age, so I bought a Franklin Ace 1000, which was, as you know, the equivalent of an Apple II Plus—with 64K RAM and lower-case keyboard.

A few years later it looked like I was going to need a more powerful computer, so I bought an IBM clone called a Vendex. I have often compared learning MS-DOS to learning Greek—without a Greek dictionary! I struggled with the Vendex for about three months, and finally went to an Apple distributor in Reno and asked to see an Apple IIe. They told me that they no longer stocked the machine and showed me the IIGS. Within half an hour of bringing it home it was assembled and up and running! I gave the Vendex to my daughter who was back in college, after setting up some batch files that let her do simple things like run the word processor by typing WP.

This past Christmas I saw a Macintosh Performa on sale and became the owner of a second Apple computer. I'm learning about the Mac, but I still use the IIGS a lot. You can bet I agree that the Apple II will be around for a long time yet.

Roscoe "Bill" Murray, Jr.
Carson City, NV

Dear *II Alive,*

Jerry Kindall's description of "growing into" Apple computing and programming, especially regarding the effects of *Nibble* magazine, struck a chord in me. I came late into Apple computing, but had the immense good fortune to immediately discover Nibble. The *Nibble* experience was unique, and probably not replicable—no other publication has ever had that kind of effect on me (although I do like what I see in *II Alive.*)

I spent countless hours in a pleasant daze, learning, enjoying, adapting, modifying—hacking! The magazine served as my tutor, my confidante, and even an outlet for my own programming efforts ("HexAddition," published in the 9/90 issue).

Years later, I'm still enjoying and learning from my *Nibble* collection. I won't get maudlin and try to describe the effect of *Nibble's* demise on me. But the entire experience was literally mind-altering and absolutely unique. I think all of us who were exposed to the magazine are better and happier for it. Not a bad epitaph for anyone's tombstone.

David Hickey-Schiappa
Cheshire, CT

THE TRUE APPLENET AUTHOR

Dear *II Alive,*

I really appreciated the article on telecommunications in the March/April issue of *II Alive,* in which D. G. Harrison credited me with the authorship of the original *AppleNet* software. I only wish it were so.

AppleNet was actually written as a class project by Kevin D'Haeze, and was posted to the online services for everyone to enjoy. Since it was written in BASIC with Morgan Davis' *ModemWorks* extensions, it was simple for beginning programmers like me to "get into" the program and design modifications and patches for it.

Once Kevin had finished the class, he lost interest in *AppleNet.* Larry Edwards saw the ugly code being produced by us *AppleNet* hackers, and gently began enforcing some programming standards so that all our modifications could work together. Before his death in 1990, Larry had laid all the ground rules, and we (as a group) had released *AppleNet* v1.3f.

When I assumed the responsibility of riding herd on the programmers, the two most prolific were Mike Potter and Derek Fong. Between the three of us, we wrote most of the modules, patches, and other modifications that eventually became *AppleNet* v1.4. But even at v1.4, the core code was still Kevin's, not ours, so I don't feel justified in any claim that I "wrote" the BBS.

Eventually, events in my life required me to turn over "upkeep" of the software to Derek, who, by all accounts, has done an excellent job. It's a tribute to the skill of programmers like Derek, Larry, and Mike that Mr. Harrison didn't realize the BBS he was logged into was written in Applesoft BASIC. But even more, it's a tribute to the original author, Kevin D'Haeze, who realized that you can write a good BBS in BASIC.

Erik "Lurch" Kloeppel
Everett, WA

GLAD TO BE OF SERVICE

Dear *II Alive,*

Since my inquiry regarding Dollars and Sense in the March/April issue of II Alive, I've received six or seven replies. There are a lot of helpful Apple II promoters out there! I encourage other subscribers to use your publication to find sources of hard-to-find hardware or software. Thanks!

Elbert F. Hubbard
Gravette, AR

TURBOREZ OR NOT

Dear *II Alive,*

Recently I was looking through some of my old issues, and I noticed that about a year ago you mentioned a product called Turborez, a video resolution and color enhancement card for the IIGS, and said it was delayed four months. I haven't heard anything about it since then. Is it still going to be produced, or is it just going to be dumped? I'd certainly purchase one for my IIGS.

David J. Clark
Manlius, NY

David: We haven't seen it either. The product was being developed by a small (one-person) company, so some snags are to be expected, but at this late date I doubt it will ever be released.—Editor

Continued on page 60

II ALIVE

July/August
Volume 2 Number 3

A Quality Computers Publication

Publisher	**Joseph Gleason**
Editor-In-Chief	**Jerry Kindall**
Managing Editor	**Ellen Konowitz Rosenberg**
Contributing Editors	**Douglas Cuff** **Steve Miller** **Mike Westerfield**
Interview Editor	**Tara Dillinger**
Review Editor	**Jeff Hurlburt**
Art & Design	**Audrey Wolfe** **Carl Sperber** **Paul Sheppard** **Marcy Schwartzberg**
Advertising Sales	**Matthew Spatafora** **Tammy Lynn**

Our Address: *II Alive,* PO Box 349, St. Clair Shores, MI 48080 USA. (If shipping by UPS or some other carrier requiring a street address, use 20200 Nine Mile Rd., St. Clair Shores, MI 48080 USA.) Also check our Voice Mail, Fax, and Electronic Mail addresses (below) for faster ways of getting in touch.

Our Phone Number: Subscription-related inquiries *only* can be handled by our Sales Department at 1-800-777-3642 or (313) 774-7200. This number is not for editorial or technical inquiries; we prefer to receive those by mail or fax.

Our Fax: We accept correspondence via fax at 313-774-2698.

Voice Mail: Letters to the Editor *only* may be left on our Voice Mail system. Dial 1-800-777-3642 (US and Canada; other countries dial +1-313-774-7200). When our phone droid answers, dial 839 on your Touch-Tone phone, or wait for an operator and ask to be transferred to extension 839. All voice mail messages become the property of *II Alive.*

Electronic Mail: We accept electronic correspondence via our BBS at +1-313-774-2652, or via one of the following online services: GEnie (QUALITY), America Online (QualityCom), CompuServe (73477,3364), or the Internet (quality@genie.geis.com).

Disclaimer: Sorry, but due to the volume of mail we receive, we cannot promise replies to individual letters, faxes, or messages. All letters, faxes, and messages sent to *II Alive* become the property of *II Alive* and may be edited for publication or used for promotional purposes. We cannot return disks.

Letters to the Editor: Send your comments, questions, and concerns to *II Alive*—Letters at the address above.

Ask Mr. Tech: Send technical questions and answers to *II Alive*—Mr. Tech at the address above.

Shareware Spy: Send new freeware and shareware programs for review to *II Alive*—Shareware Spy at the address above.

Macro Exchange: All macro submissions become public-domain. If we print your macro, we'll send you $10. Send macro submissions to *II Alive*—Macro Exchange at the address above. Please include a disk if your submission is longer than half a printed page.

Art Gallery: You retain copyright on graphics we print in our Art Gallery department. Send graphics files (in a standard high-resolution, double-high-resolution, or Apple Preferred super-high-resolution format) to *II Alive*—Art Gallery at the address above.

Reviews: Send products for review to *II Alive*—Reviews, c/o Jeff Hurlburt, 7814 Santa Elena St., Houston, TX 77061. Also contact Jeff if you are interested in writing a review for *II Alive.*

Publication Schedule: *II Alive* is published on a bi-monthly schedule, with issues appearing in January, March, May, July, September, and November. Please allow until the 15th of the cover month for delivery (international subscribers allow slightly longer).

Subscription Information: U.S. subscriptions cost $19.95 (for a six-issue, one-year subscription). In Canada and Mexico, the annual rate is US $29.95. Outside North America, subscriptions are US $39.95 (for surface delivery) and US $49.95 (for airmail delivery). To subscribe, send check, money order, or credit card number (Visa, MasterCard, Discover, or American Express) to *II Alive*—Subscriptions at the address above. You may also subscribe by credit card by calling our Sales Department at the phone number above.

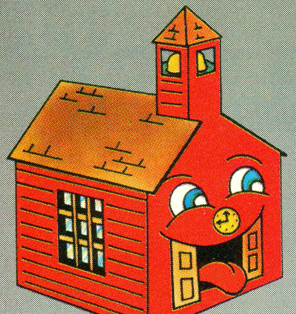
Damaged and Missing Issues: If you receive a damaged copy of *II Alive,* or do not receive a copy at all, contact Subscriber Services at the address above, or call our Sales Department at the phone number above.

Expiration Dates: Your subscription expiration date is printed on your mailing label. This date indicates the *last* issue you will receive—i.e., an expiration date of 3/94 means that the March/April 1994 issue will be the last you will receive. Renew at least a month before this date to ensure uninterrupted service.

Change of Address: Please mail a copy of your current mailing label, along with your new address, to *II Alive*—Subscriber Services at the address above.

Writing for II Alive: Send a self-addressed stamped envelope to *II Alive*—Editorial Department at the address above for our detailed Writer's Guidelines.

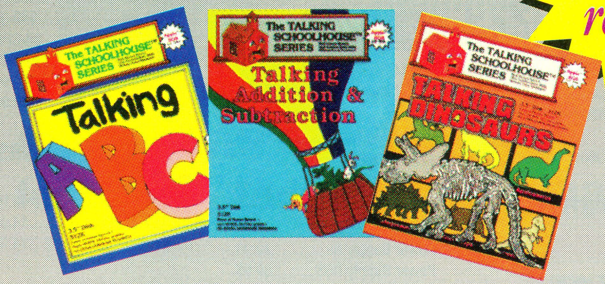
PRINTED IN THE UNITED STATES OF AMERICA



THE TALKING SCHOOLHOUSE SERIES

Choose any Talking Schoolhouse program for only $24.95 each!

50% off retail!



Imagine a "Talking Schoolhouse" inside your GS computer—software that uses perfect human voice and stunning graphics to teach basic skills to youngsters. It's the TALKING SCHOOLHOUSE™ SERIES from Orange Cherry Software. Software programs captivate children ages 3-9 as they explore basic concepts on their own:

- Talking ABC's
- Talking Alpha Chimp
- Talking Dinosaurs
- Talking Colors & Shapes
- Talking Animals Activity Set
- Talking School Bus
- Talking Classroom
- Talking First Reader
- Talking First Words
- Talking First Writer
- Talking First Dinosaur Reader
- Talking Reading Railroad
- Talking Speller
- Talking Speller II
- Talking Addition & Subtraction
- Talking Multiplication & Division
- Talking Numbers

This software brings the friendly and reassuring voice of a real person into the computer learning environment. Developers working with proven educational concepts have recorded actual human voice & electronically digitized the sound into the computer program. The result is perfect human speech—not a synthesized computer voice that only confuses youngsters.

Letters, numbers, words, concepts, and even dinosaurs come alive in thousands of colors with super high resolution graphics. These programs let the GS computer perform at its maximum level.

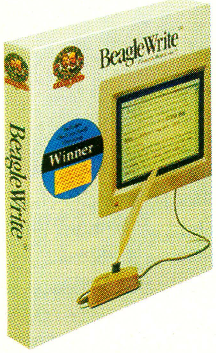
The Talking Schoolhouse Series from Orange Cherry Software. All programs are available on 3.5" disks for the Apple IIGS. To order, or for more information contact Quality Computers at 1-800-777-3642.

Call 1-800-777-3642 to place your order!

"Your people are always friendly and helpful." —Ray A. Oltman

Quality Computers™

SPECIAL • SPECIAL • SPECIAL



The finest in Apple II word processing

BeagleWrite for the IIc and IIe .... $48^95

SPECIAL • SPECIAL • SPECIAL

Save $19

Ultimate Apple IIGS Utilities

Signature/Six Pack Combo $39^95

Signature $29^95 • Six Pack $29^95

SPECIAL • SPECIAL • SPECIAL

Save $44

Creativity IIGS Style

Platinum Paint BeagleDraw Combo........... $79^95

Platinum Paint $59^95 • BeagleDraw $64^95

Platinum Paint
Activity Guide 32.95
90 IIGS Fonts 47.95

ROGER WAGNER
HyperStudio 109.95

TIMEWORKS **SPECIAL ▼**
Publish It! 4 ....29.95

UTILITIES

BEAGLE BROS
Program Writer 32.95
GS Desk Accessories 34.95

BYTEWORKS
Toolbox Pascal...............74.95
Toolbox C.......................74.95
Orca/M 2.0 ....69.95
Orca/C 79.95
Orca/C Bundle 119.95
Orca/Pascal 79.95
Orca/Pascal Bundle 119.95

QUALITY COMPUTERS
System 6.0.1 Upgrade
System 6.0.1 Upgrade.....................29.95
Original Bonus Pack ...59.95
New Bonus Pack 39.95
Six Pack 29.95
Signature.........................29.95
RepairWorks 34.95
SuperPatch 34.95
EasyDrive 39.95

SOFTSPOKEN
Cross-Works 69.95

VITESSE
Harmonie 39.95
HP Driver Set 24.95
PerfectImage...................19.95
Salvation Supreme 89.95
Bakkup 2.0 ....44.95

EARLY LEARNING

BRODERBUND
The Playroom SV 39.95
The Treehouse SV 39.95

DAVIDSON
Math and Me SV..............32.95
Math and Me GS SV........39.95
Reading and Me SV.........34.95
Reading and Me GS SV ...39.95

LAWRENCE PRODUCTIONS
The Lost Tribe 29.95
Katie's Farm GS...............29.95
McGee GS 29.95
McGee at the Fun Fair......29.95

MECC
Arithmetic Critters 39.95
Counting Critters 39.95
First-Letter Fun................39.95
Fun From A to Z 39.95

Paint with Words.............39.95

ORANGE CHERRY
Talking Alpha Chimp GS..39.95
Talking Animals GS 39.95
Talking ABC's GS 39.95
Talking Dinosaurs GS......39.95
Talking First Words GS ...39.95
Talking School Bus GS....39.95

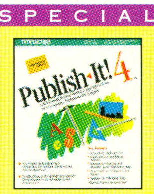
STICKYBEAR
ABC 24.95
Numbers 24.95
Opposites 24.95
Shapes GS 29.95
Opposites GS 29.95
Shapes 24.95
Alphabet GS 29.95

SOCIAL STUDIES

BRODERBUND
Carmen USA SV 29.95
Carmen World SV 29.95
Carmen World GS SV......29.95
Carmen Time SV 34.95
Carmen America's Past SV...34.95

MECC
Oregon Trail 24.95

SPECIAL



PUBLISH IT! 4
$29^95

LANGUAGE ARTS

DAVIDSON
Read 'n Roll SV 42.95
Word Attack Plus SV.......41.95
Grammar Gremlins SV 39.95

LEARNING COMPANY
Reader Rabbit 1 ....44.95
Reader Rabbit GS...........54.95
Magic Spells....................42.95
Children's Writing & Publishing Center SV..64.95

MECC
Word Munchers 24.95
Spellevator 39.95

ORANGE CHERRY
Talking First Reader GS...39.95
Talking Reading RR GS ...39.95
Talking Speller 1 GS........39.95
Talking Speller 2 GS........39.95

STICKYBEAR
Reading...........................24.95

Reading Comp 29.95
Reading Tutor 29.95
Spellgrabber...................29.95

MATH

DAVIDSON
Algeblaster Plus 39.95
Math Blaster Plus............39.95
Math Blaster Mystery 39.95

LEARNING COMPANY
Math Rabbit 24.95

MECC
Number Munchers 24.95
Fraction Munchers 39.95

ORANGE CHERRY
Talking Addition & Subtraction GS 39.95
Talking Money GS 39.95
Talking Clock GS 39.95

STICKYBEAR
Math...............................24.95
Math 2............................24.95
Math Tutor 29.95
Word Problems 29.95

BOOKS & VIDEOS

Apple II Guide 6.00
Print Shop Handbook......19.95
System 6 Book 12.95
Programmer's Reference to System 6 ....44.95
Q Drive Video 19.95
System 6 Video 19.95

GAME SPECIALS

FOR THE APPLE II
BUY ANY THREE FOR $19.95
PacMan 9.95
Ms. PacMan 9.95
Centipede 9.95
Defender 9.95
Donkey Kong 9.95
Robotron 9.95
StarGate 9.95
Jungle Hunt.......................9.95
Moon Patrol 9.95
Galaxian 9.95
Battlezone 9.95
Gremlins 9.95
Track & Field 9.95
Dig Dug 9.95
Joust 9.95
Tsushima 9.95
Scimmars...........................9.95
Planet Miners 9.95
Mines of Titan 9.95
Body Transparent..............9.95
Remember 9.95
Silent Service 9.95
Comics 9.95
Guderian 9.95

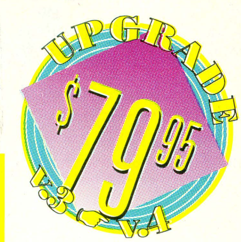
1-800-777-3642

We will meet or beat any nationally advertised price

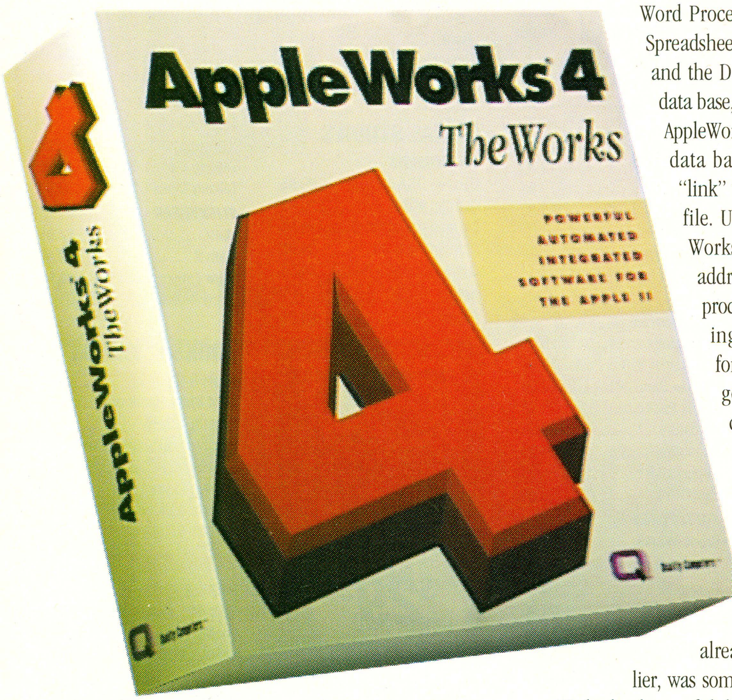
VISA MasterCard American Express Discover SCHOOL P.O.s

AppleWorks® 4.0

More power, more speed, easier to use!



If you haven't upgraded to AppleWorks 4.0 it's not too late! Upgrade to AppleWorks 4.0 from *any* Word Processing program for only $99.95!



In 1988, AppleWorks seemed to have reached its pinnacle with the release of version 3.0. But you weren't satisfied and either were we. That's why Quality Computers, and AppleWorks veterans Randy Brandt and Dan Verkade have developed AppleWorks 4.0—the most extensive upgrade in AppleWorks history.

ONE TOUCH COMPUTING

How would you like to automate AppleWorks? You can with AppleWorks 4.0. That's because AppleWorks 4.0 features dozens of built-in automatic functions that you can do with just a keypress. Imagine being able to set up and print an envelope automatically, or open a new data base with pre-formatted fields by pressing a key. There's lots of others just like it, and they're easy with AppleWorks 4.0!

TRUE INTEGRATION

Integration, always AppleWorks' strong suit, will become tighter than ever with new features to allow the Word Processor to access data base files, the Spreadsheet to access other spreadsheet files, and the Data Base to access word processor, data base, and spreadsheet files. For example, AppleWorks 4.0 will allow users to create a data base of names and addresses, then "link" the Data Base with a word processor file. Using the glossary function, AppleWorks 4.0 can look up and import an address directly into the current word processing document—without switching modules, copying, or manually formatting. You can also import categories from other data bases (and cells from spreadsheets) and to export information to other data bases, providing the Data Base module with relational capabilities.

EVEN EASIER TO USE

Making AppleWorks, an already friendly program, even friendlier, was something we thought about carefully. We had to be careful that what we were doing was REALLY making AppleWorks easier. We think we succeeded. For example, AppleWorks 4.0 can remember what order you used for each of your reports and will automatically sort the data base for you. The Spreadsheet now features a pop-up list of functions so users don't have to remember codes when entering formulas. The Word Processor uses distinctive symbols for formatting codes (instead of just carets) so boldface and underline can be recognized at a glance, instead of requiring the cursor to be on the formatting code to read it. The "Change Disk" menu allows users to display disk names by pressing OA-? instead of requiring them to know what slot and drive their data disk is in. "Add Files" displays text files and automatically loads them as word processor files instead of requiring users to go to a separate "New File" menu. The Word Processor lets you see and edit tab rulers right in the document. AppleWorks 4.0 even takes away the worry of saving your files with its Auto-Save function.

MORE COOL STUFF

Other major features include built-in support for HP's popular Deskjet printers, faster display and finds in the Data Base, split screen capability in the Word Processor, and data math functions in the Spreadsheet. The Data Base will have improved import and export facilities for exchanging data with other computers, and will feature spreadsheet-style formulas in calculated fields. A global auto-save feature, available in all AppleWorks modules, will protect users' work from power failures; and a Quick-Path menu will let users set up a menu of their most frequently-used directories.

Because AppleWorks 4.0 includes TimeOut, adding useful enhancements is easier than ever. In fact, you can even install most enhancements without leaving AppleWorks.

TRUE TO ITS HERITAGE

AppleWorks 4.0 will remain true to the AppleWorks spirit. Menus will remain easy to navigate; commands will continue to be simple-to-remember Apple-key combinations; help will still be available with a single keypress; all previous functions will remain the same.

APPLEWORKS 4.0

UPGRADES	
from version 3 5.25"	$79.95
from version 3 3.5"	79.95
Competitive Upgrade 5.25"	99.95
Competitive Upgrade 3.5"	99.95
10-Pack	499.95
50 User Site License	19.00/CPU
NEW INSTALLS	
Consumer	169.95
10-Pack	949.95
3.0 Network version	1,089.95
50 User Site License	39.00/CPU

Call for Site Licenses over 50 Users

1-800-777-3642

Quality Computers™

FEDERAL EXPRESS VISA MasterCard AMERICAN EXPRESS Card DISCOVER

School P.O.'s by phone, fax or mail

Spectrum, GNO/ME, Ancient Glory & MegaBuff

II ALIVE RATINGS

★★★★	Excellent
★★★	Very Good
★★	Good
★	Fair
☆	Poor

SPECTRUM

★★★

Seven Hills Software

$100.00

Requires Apple IIGS, 1.25 MB RAM, System 6.0 or later; 2 MB RAM and hard drive recommended

Reviewed by Charles L. Mudd, Jr.

Was *Spectrum* worth the wait? If state-of-the-art IIGS telecommunications capabilities are your criterion, the answer has to be Yes. Written by Ewen Wannop, the new Seven Hills communications package features flexible emulation and scripting, all in an easy-to-use Desktop environment.

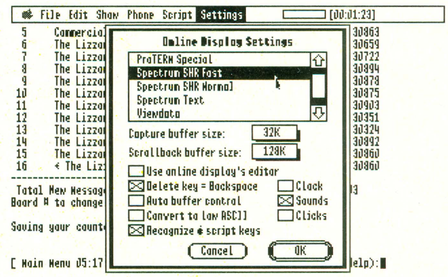
Emulation is the ability to mimic other software and hardware, so that your IIGS acts like a standard TTY, VT-100, or other terminal. (Before personal computers, people used to use stand-alone devices with screens and keyboards to communicate with larger computers, and although most people use personal computers for these tasks today, the word "terminal" has stuck.)

For instance, *Spectrum's* ANSI emulation lets remote systems display color text, highlighting, and graphics on your IIGS. Many IBM-based BBSs (bulletin board systems) offer this display mode, which is quite a bit more attractive than plain text. ProTERM Special is used on Apple BBSs; it can generate many ANSI-style screens, but uses a faster monochrome display. (As you might guess by its name, ProTERM Special emulation was first included in *ProTERM.*)

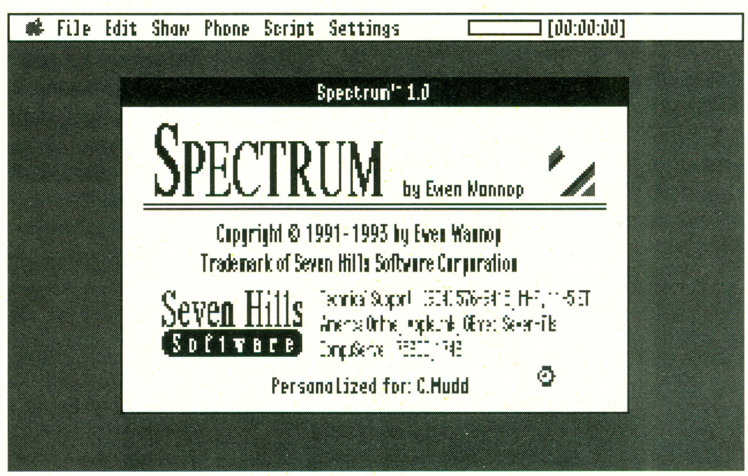
Unlike other programs, *Spectrum* lets you display your online session on either the IIGS super-hi-res graphics screen (for full color and ANSI compatibility) or on the text screen (for maximum possible speed). Most BBSs you call will work fine with *Spectrum's* "SHR" and "Fast SHR" displays. To the remote system, these act like Standard TTY, and present text on an attractive Desktop screen with standard pull-down menus. The package also includes the "Spectrum Text" display, a "Viewdata" terminal emulation, and an "Extended" version of ProTERM Special with enhanced color and graphics capabilities. These terminal emulations are modular; you can add new ones (when they become available) by dropping them into Spectrum's Add.Ons folder.

Scripting is a way to use automate your online activities. *Spectrum* lets you create scripts for custom emulations and for logging on, retrieving e-mail, logging off, and about anything else you do online. You don't need to understand scripting to use *Spectrum,* but scripting brings powerful new capabilities to the IIGS telecomm user. The package includes an indexed 95-page booklet, Scripting, which describes *Spectrum's* scripting language. (The English-like language, similar to that used by *Talk Is Cheap,* is simple to learn.)

Spectrum's main user manual, Getting Started and Reference, focuses on general everyday use of the software. It begins by outlining basic system requirements and installation for both 3.5" diskette and hard drive users. With the instructions provided, I installed Spectrum on my hard disk in less than five minutes. (For hard drive users, a diskette is provided with optional spoken messages for events such as "You have mail," "Sending mail," "Sending files," etc. These sounds can be activated from scripts or by *Spectrum* itself in the normal course of operation.)



The manual's Guided Tour gets you started in a hurry, while the "Reference" section explains potential pitfalls—such as possible causes of missing characters and how moving the mouse can slow data transmission. It also covers using high speed modems, modem cable wiring, saving a screen image to disk, and other telecommunications facts that are nice to have in one manual.



The manuals and in-program Help texts deliver concise and readable explanations for every *Spectrum* feature. Seven Hills also includes a handy Commands Card showing the program's major menus and "hot keys," and all the scripting commands are listed (along with manual page references) on the back. However, the manual doesn't have any tutorials, and is short on examples for some of the more sophisticated features.

Once *Spectrum* is installed, you need only double-click its icon to be on your way. Even when using ANSI or PTS online displays, you are just an Apple-W away from the familiar Desktop and easy access to File, Edit, Show, Phone, Script, and Settings menus.

The first time I used *Spectrum,* I went directly to the Phone menu and selected "Dial

Number." *Spectrum* allows you to accept the default port settings, or specify custom ones, for each entry into the phone book. You can also choose a login script for each number and select the number of redials the program will attempt upon encountering a BUSY signal.

My first attempt at a *Spectrum* connection was to my university's computer system. Because *Spectrum* and the university both support ANSI, I selected that emulation mode. After entering the phone number and other information, the program saved my entry in the phone book, where it is instantly accessible. Once the phone book entry has been created, logging on was straightforward and painless. The emulation performed perfectly. There were no improperly drawn characters or screen glitches. Out of the corner of my eye I noticed the online timer keeping track of the connect time—useful on pay systems like GEnie and CompuServe.

Besides the Desktop and online displays, the program also has Capture Buffer and Scrollback displays. Each buffer is adjustable in size from 0K to 3072K. A full-featured text editor is included, and is quite handy for dealing with selections lifted from the buffers, composing messages to be sent via the "Send Editor Text" command, and similar tasks.

Other windows and dialog boxes handle chores like setting editor formatting options (e.g., to strip linefeeds or other control characters), creating and editing scripts, editing character filter tables, and selecting of transfer protocols. At present, *Spectrum* offers several varieties of Xmodem, along with Ymodem, Zmodem, and CIS B+. (Seven Hills plans to add the Kermit protocol soon.) The file transfer window is attractive and displays an assortment of vital information.

If you value the freedom and friendliness of the Desktop interface, *Spectrum* is the supreme telecommunications application. If you want a flexible, easy-to-update package sure to stay on the leading edge of IIGS telecommunications, it's hard to imagine a better choice!

GNO/ME

★★★

Procyon, Inc.

$89.95

Requires Apple IIGS, 1.25 MB RAM, System 6.0 or later, and hard drive required; 4 MB RAM, accelerator, and modem recommended

Reviewed by Dan Brown

In the late 1960s, a small team of researchers at AT&T's Bell Labs began working on what would later be known as the Unix operating system. Unix became very popular because it had the ability to run more than one program at a time—and just as importantly, allowed these programs to communicate with each other. Furthermore, the operating system was "lean," used few system resources, and ran quickly. Today, Unix is the operating system of choice on most workstations and minicomputers, and is even found on mainframes and supercomputers. In 1991, Procyon released the closest thing to Unix for the Apple IIGS: *GNO/ME*, the GNO Multitasking Environment.

GNO/ME provides much of the flexibility and power found in Unix. Multitasking allows more than one program to run at a time. For instance, you could be compiling a program, reading your E-Mail, unshrinking files, and playing music in the background all at once. Unlike the "cooperative" multitasking used on the Mac (which only works when programs are written to periodically surrender control to the operating system), *GNO/ME*'s multitasking is interrupt-based, which allows much more accurate and reliable time-slicing among running programs.

GNO/ME's job control makes it easier to manage all the programs—also known as "processes"—that are running at any time by moving them into the foreground or the background, or by stopping them entirely.

The cost of this power is, of course, user interface. *GNO/ME,* like Unix, uses a command-line system, where you control the computer by typing your instructions at a prompt. Users who prefer using the Desktop may consider *GNO/ME* to be a throwback. Still, it hardly seems fair to knock *GNO/ME* for being exactly what it sets out to be—a Unix-like operating system and shell for the IIGS.

GNO/ME comes with a number of utility programs that can be run from the command line, and, due to the system's popularity among programmers, many more are available for downloading from the Internet and GEnie. If you've used Unix, you'll recognize many of *GNO/ME's* utilities—programs like *grep, less, more, vi, emacs,* and *tar.* Some of them aren't from the Unix world—*yankit,* which extracts ShrinkIT archives, is one of these.

Among other useful *GNO/ME*-compatible utility programs, you'll find Ian Schmidt's TMTerm, which provides an interface to *GNO/ME* from Desktop programs, and Dave Tribby's *Synthfile,* which plays and provides information on *MIDISynth* music files. These utilities, and more, are available free from various online services and the Internet.

And remember, it multitasks. *GNO/ME* lets you simultaneously run one Desktop program and any number of GS-specific text-based programs. (That one desktop program may be Procyon's *Switch-It,* which allows switching, but not multitasking, among as many Desktop programs as you have memory for. *GNO/ME* does not work with *The Manager.*) *GNO/ME* does not directly support more than one desktop program, or 8-bit programs.

We tested *GNO/ME* with a wide range of software. No conflicts showed up with Quality Computers' *Signature,* Seven Hills' *GraphicWriter III,* the *Finder, GS-ShrinkIT,* WestCode's *HardPressed,* Econ's *AutoArk,* and many other programs.

The documentation included with *GNO/ME* is very detailed, covering the system from both a user's and a programmer's perspective. For the user, it explains how the *GNO* shell, *gsh,* works; how commands are interpreted; and the system's job control features. For programmers, the new functions in the included C libraries are explained. Additional references are given if more information is available. Virtually anyone with a little Unix or IIGS experience should be able to set up the system without too much trouble.

Supplied on three diskettes with 297-page loose leaf manual, *GNO/ME* is a powerful, stable, and competent implementation of Unix for the Apple IIGS. Whether you're an advanced programmer or a novice, *GNO/ME* is a must for getting the most out of your IIGS.

ANCIENT GLORY

★★

Logical Design Works/Westwood

$20.00

Requires Apple IIGS, 1.25 MB RAM, System 5.0.4 or later

Reviewed by Colin Williamson

Perhaps it's just my imagination, but it does seem like Big Red has stepped up their IIGS game offerings since *Out of This World*'s phenomenal success. *Mazer II* is one recent example of this trend; and now, there's *Ancient Glory,* an arcade-adventure boasting myths, monsters, and gore galore. As Hercules, your challenge is to rid ancient Greece of Medusa, thus earning a place beside Zeus high atop Mount Olympus. As a bonus, you get to be immortal!

You soon discover that the land is knee-deep in giant scorpions, lions, cyclops, and other hostile nasties. Even worse, the goddess Hera is ever ready to create replacement monsters or hurl stone columns at you. Fortunately, you can get helpful hints from scrolls, and friendly gods (Hermes and Athena) offer powerful magical artifacts as rewards for services rendered. You must complete five quests to obtain the items needed to battle Medusa.

As in *Prince of Persia* and other such games, you can make your hero figure run, jump, and attack on a horizontally scrolling field of play. With joystick control, response is smooth and natural-feeling, but the optional keyboard controls are difficult and frustrating. The action area fills roughly a third of the screen. Below this are status bars to indicate your health, ammunition, and time remaining for your quest. The rest of the screen is filled by inventory icons that tell you what weapons, armor, and other items you possess.

Continued on page 19

QUESTION: I need more room on my System 6.0.1 System Disk for an application. (I really hate swapping disks.) Which tool files can I dispose of without causing problems?

Andrew Eisenman
Gilbert, WV

ANSWER: I wouldn't suggest deleting any tool files (i.e., files in the Tools folder, inside the System folder) unless you are positive you know which tools the program you're trying to put on the disk does and does not use. A program can use any tool file it wants to, although naturally many programs don't use every single tool. The best way to find out which tools a program uses is to write the program yourself (sorry, there's no easy way to find out which tools an existing program is using).

Instead of deleting tools, consider deleting some of the following files:

- P8 (inside System folder). Delete this if you are not using any ProDOS 8 (IIe/IIc) programs.
- BASIC.System (in main window). Delete this if you are not running any BASIC programs.
- AppleDisk5.25 (in Drivers folder). Delete this if you don't have (or can live without) your 5.25" drive.
- ControlPanel (in Desk.Accs folder). Delete this if you don't need to change your printer. (Hint: start up the disk, set up the printer as desired, then delete the ControlPanel file.) Make sure you use the Install disk to put a printer driver on the System disk—unless, of course, you're not going to be printing anything.
- Printer (in CDevs folder). What we said about ControlPanel applies here. Delete this once you've chosen your printer.
- Time (in CDevs folder). Nuke this. You can set the time using the text-based Control Panel (Apple-Control-Escape) if you ever find it necessary to do so.
- TS2 and TS3 (in System.Setup folder). These are tool patch files that update the ROM-based tools in the IIGS. Delete TS2 if you are using a ROM 03 IIGS. Delete TS3 if you are using a ROM 01 IIGS.
- Start (in System folder). This is the IIGS Finder. If you will be using this as a boot disk for a specific program, delete this file and place the SYS16 file (for example, MyProg.SYS16) in the disk's main window.

By deleting all the stuff we mentioned above (except for Start), you save at least 107K (a little more if you have a ROM 01 IIGS, since TS3 is a little bigger than TS2). Deleting the Finder (the Start file) will save you another 183K.

QUESTION: I have a hard drive on my system, but when I turn on my computer I have to reset and type PR#7 to get the drive to boot.

John Runyon
Middlebury, VT

QUESTION: I have networked my classroom's Apple IIs using the Digicard system. The Digicard interface is in slot 7. My Platinum-colored IIes boot up directly to the network when turned on, but with the older IIes we have to type PR#7 to boot the network.

Richard Schalhamer
Golden, CO

ANSWER: Older Apple IIes only know about Disk IIs (5.25" drives) and won't start up from any other kind of disk drive—hard drive, RAM disk, 3.5" drive, or network. In 1984 Apple made a ROM upgrade available to fix this (and the many other problems that were present in the original Apple IIe ROMs). This upgrade, called the Apple IIe Enhancement Kit, gives your older IIe the functionality of a newer IIe. It costs about $60 and can be obtained from your favorite Apple II supplier.

All Apple IIe computers bought during 1983 and early 1984 should be upgraded to remain compatible with current hardware and software. (Many of the products released in the past ten years requires this ROM upgrade.) If your computer says "Apple //e" at the top of the screen when you turn it on, it already has this upgrade; if it says "Apple][", you have the older version.

Having 128K in the machine is not the same thing as having an "enhanced" IIe. The IIe Enhancement Kit is a ROM upgrade, not a memory upgrade.

Enhancing your Apple IIes will solve your problems. (I'm frankly surprised you haven't found more things that wouldn't work in your un-enhanced machine in the past decade.)

QUESTION: I have been using the *BPI* accounting system since 1986. I've noticed that the program will become obsolete at the end of 1999, since it only stores two-digit year numbers. Is there any way to update the program so it will accept higher dates? Failing that, are there any business programs for the Apple II which will accept dates after 1999?

Carol Denning
Fremont, NE

QUESTION: Since its release in the mid 1980s, the Apple II version of *Quicken* has never been updated. My concern is that the program will only allow date entries up to and including the year 1999. Intuit's customer service tells me that they are no longer working on the Apple II version. Is this program to become obsolete at the turn of the century or is there something I can do to update the program?

John Forkes
Waukesha, WI

ANSWER: It's a shame that the programmers of *Quicken* and the *BPI* accounting system didn't realize there would still be people using their software at the turn of the century. Unfortunately, there's no way to upgrade the software yourself (unless you're a pretty decent programmer). In my own programming I've always handled dates in such a way that the program would be good for a whole century. In other words, if you typed in a two-digit year lower than, say, 80, the program would automatically assume you were talking about the 21st century, i.e., 2080. (If a program came out in 1988 you can be pretty sure users won't be typing in any checks dated "80" until 2080.)

I really don't know of any full-fledged bookkeeping software still being produced for the Apple II. You might be able to work up something in AppleWorks, especially if you have a relatively small number of customers (say, under a thousand) and know your accounting pretty well. You'd have to keep the year separate from the rest of the date, though, since AppleWorks also stores years with only two digits.

But even if you have to migrate to another computer (probably MS-DOS based) where more business software can be found, you can still look on the bright side. Having to change computer platforms once every fifteen years really isn't all that bad! One can only hope that your next machine will be useful for as along as your Apple.

QUESTION: In the November/December 1993 issue you answered a letter from William C. Roemer who said he has a "SyQuest 40 MB hard drive" connected to his Apple IIGS. But the SyQuest drive is strictly a Macintosh drive. So where did he get the drivers and other materials to run the drive on his IIGS?

Brian Terrill
Stockton, CA

ANSWER: Your characterization of the SyQuest drive as "strictly a Macintosh drive" is mistaken. The drive is a SCSI-compatible mass storage device, and thus can be used on any computer—Apple, Mac, IBM, Amiga, etc—just like other SCSI hard drives. No special drivers (other than the SCSI driver included with System 5.0 and later) are needed. All you need is an Apple II SCSI Card or a RamFast/SCSI card. The utilities included with either of these cards can format cartridges for use with ProDOS, or, if you're using System 6, you can also read and write Macintosh-format cartridges. ■

Guest Tech:
Turn On, Drop Out

by John J. Kennedy

Computers are very sensitive about the electricity they consume. Most of us have surge suppressors on our computers to deal with the minor voltage surges that are a part of any municipal power supply—not to mention the occasional "killer spikes" resulting from a lightning strike or crossed wires. Some even have uninterruptible power supplies (UPSs) which ensure that the computer keeps going even when the power goes off.

But what happens when you have a voltage dropout? Voltage dropouts are particularly common in the summer when everyone is running their air conditioner. In extreme cases they're called brownouts, but they can affect your computer long before they get that severe—especially if your computer has an accelerator or other sensitive component.

About a year ago, my Apple IIGS started playing tricks on me. Every now and then the outside border of the computer screen would flash different colors and the computer would freeze up! At about the same time, I started noticing that the lights in the house would sometimes flicker very, very slightly at random intervals. I thought my eyes were playing tricks on me at first, but it happened frequently enough that I eventually realized that something strange was going on.

I called Louisville Gas & Electric, who sent an electrician to my house to test the voltage in my home. He determined that, at times, I was experiencing a 3.5 volt drop. He traced the problem to two sections of neutral wire—one right outside the house and one up on the pole—and replaced both sections of weatherbeaten wires. After this, his measurements showed only an occasional half-volt drop, which is well within tolerance for residential service and solved both the flickering light problem and the computer problem. Since the problem was with the wiring outside my house, I paid nothing.

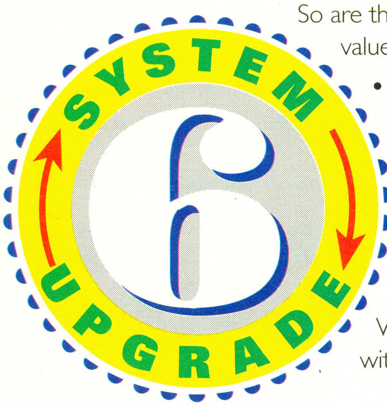
I later figured out that the problem with the wiring had been compounded by the fact that I sometimes run several major appliances at the same time. Under normal circumstances, the faulty wires could handle my house's load, but they couldn't handle the high demands of my air conditioner, dryer, and refrigerator all at once.

Not being one to leave everything to the power company, I bought an S.L. Waber surge and voltage dropout protector for my computer, which also has a telephone line surge protector for the modem. Unlike traditional surge protectors, this device has circuitry which can actually boost the voltage when it falls too low (as long as it remains within certain tolerances). It cost me about $100, but for complete peace of mind, it's well worth it. My computer hasn't hiccuped on its power feed since.

"QUALITY COMPUTERS' SYSTEM 6 BO WAY TO ADD VALUE AND FUN TO YOU

New Bundles!

Apple IIGS System 6.0.1 is BIG news!



HOT
New Features!

So are the new bundles that we are offering to make your upgrade more fun and value-packed! With System 6.0.1 you get:

- Easy to use Finder that navigates with keyboard or mouse.
- Ability to read IBM files directly into your IIGS (with special drive).
- Magic System Folder that knows just where to stash your fonts and DA's
- Add Alias to let you set up your desktop like a push-button menu selector.
- Boot ProDOS 8 quickly.
- Over 100 bug fixes.

With the QC System 6.0.1 Upgrade you get the System 6.0.1 disks plus our own System 6 Book with 6.0.1 Addendum, and a free program to customize your desktop—DeskPicture. All for just **$29.95**

1 THE ORIGINAL BONUS PACK

The Apple II Enhancement Guide: This book is packed with information to help you upgrade your Apple II. It covers RAM, hard drives, accelerators, the Finder, and more.

FlashBoot: What is faster than a speeding disk drive? A RAM disk. FlashBoot lets you automatically set up a super-fast, super-convenient RAM disk.

Clip Art: We're throwing in over 100 beautiful clip art images, perfect for desktop publishing or hypermedia applications.

Clip Sounds: Because the IIGS's sound capacity is so great, we're also including over 50 digitized sound files. Some are classics while some are the newest on the market. They go great in HyperStudio and other sound programs, or you can have music playing in the background while you work.

Icons: Over 50 fun, useful icons.

Desk Accessories: Just to give you more to choose from, we're giving you several handy desk accessories including: Enhanced Calculator, Scrapbook, Games, and more.

Fonts: Using the same font for everything is very uncool, so our Bonus Pack includes over 30 display and text fonts.

Video: Our 1-hour video takes you from installation of System 6 to moving through the Finder with speed and agility.

$59.95

THE NEW BONUS PACK 2

- 15 New Fonts
- 45 New Digitized Sounds
- 230 New Clip Art Images
- Icon Editor
- Art Gallery
- Animated Pointer Symbols
- Virus Detector
- DA Word Processor
- Sound Editor
- Window Hider
- Software Demos
- Utilities

$39.95

NUS PACK...THE MOST COST-EFFECTIVE R APPLE IIGS." –THE APPLEWORKS EDUCATOR

3 SIX PACK An incredible set of Finder Extensions including:

MoreInfo: Displays more information about your files, including their actual type and auxiliary type, access information, and more.

AlarmClock: Displays the time in a window or at the right of the menu bar. Plus it'll alert you for important occasions.

SuperDataPath: Remembers a default data directory for your programs so you don't have to change folders every time you launch an application.

HotKeys: Add dozens of functions to your function keys (on an extended keyboard) or to your numeric keypad.

CDEV Alias: Add any frequently-used Control Panel function (CDEV) directly to the Apple menu.

CPU & Memory Use: Shows visually how much of your computer's resources are being used at any time.

SelectIcons: Select icons in the front window based on partial filename, filetype, modification date, and more!

PrintCatalog: Prints and old-style directory listing on your printer, including filename, filetype, and more.

Encrypt: Save your data from prying eyes! Encrypt creates a password-protected copy of any file.

FilePeeker: Preview the contents of graphics, text, and sound files—and more—without having to launch a separate application!

WorkSet: Remembers a group of files and opens them all at once, with one double-click! Open a group of AppleWorks documents—or a group of folders—or a group of anything—all at once!

FileCompare: Check seemingly identical files for differences you might have missed comparing them visually.

ButtonBar: Places a button bar on the desktop with user programmable buttons.

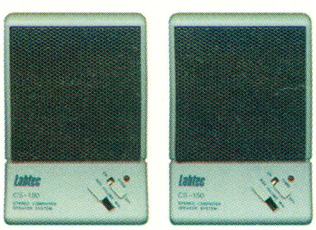
SizeUp: Calculates the size of the currently selected files, including files residing in selected folders, from within the Finder. SizeUp also tells you the total number of files or folders selected.

LaunchList: Provides a list of applications in a window when running the Finder. The LaunchList window can be configured to contain any application, or all of your applications.

XtraSounds: Adds over thirty new sound events to your system. Each of these new sounds is associated with a Finder function. Now you can have the Finder "talk" to you whenever it's doing something.

$29.95

VALUE-PACKED SOFTWARE BUNDLES



(Speakers sold separately for $19.95 a pair)

Buy any bundle with the purchase of the QC System 6.0.1 Upgrade and SAVE $10! Or buy any 3 bundles & SAVE $30, plus get a set of speakers for only $10!

WHILE SUPPLIES LAST!

30 Day Money Back Guarantee

4 Take advantage of your IIGS's sound capability with a sound input card and microphone, sound editor, and sound effects disk with twelve new sounds!

$49.95

Quality Computers™

20200 Nine Mile Rd. • St.Clair Shores, MI 48080
(810) 774-7200 • Fax: (810) 774-2698

SCHOOL P.O.s

To place your order call . . .

1-800-777-3642

GET YOUR
APPLE II CATALOG

Are you tired of searching for a complete line of Apple II products and quality support? Then ***STOP HERE*** *because Quality Computers and its Apple II Catalog provide* ***NEW EXCITEMENT*** *for these Classic Computers!* ***OUR APPLE II CATALOG*** *offers hardware and software solely for the Apple II. You'll find everything from AppleWorks 4.0, to utility programs, educational software, hard drives, memory, modems and much more. So if you want to know what's available for your Apple II, call Quality Computers*

YOUR 1 STOP APPLE II SHOP

$5.00

Call for your Apple II Catalog today

GUARANTEED LOWEST PRICES

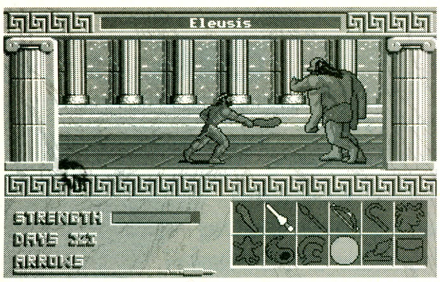
1-800-777-3642

Quality Computers™

20200 Nine Mile Rd., St. Clair Shores, MI 48080 • 1-810-774-7200 • Fax 1-810-774-2698

On the whole, the graphics are only decent. You can expect bright, crisp (if somewhat flat) monster pictures and scenery. The animation, however, is exceptional, especially for the major creatures—like the River God, Cyclops, and so on—you encounter at the end of each level. These are quite lifelike and fluid in their motion. As for sound—aside from a couple screams and clangs, there's little here to enhance the game's atmosphere. In fact, when a monster gets hit, a musical note chimes! Call me violent, but I would have much preferred a digitized "Aaargh!" or "Pfsplurt!"

Between action sequences, the game displays a detailed map that let you choose your



next quest. Most locations lead to new combat challenges, but a few, like Delphi, are home to verbose Oracles which supply advice and useful information about quests.

After playing a while, I discovered that the game is, for the most part, non-linear. You can go wherever you want to go, whenever you want to go there. The only catch is that Hercules has only seventeen days to obtain critical artifacts and dispatch Medusa, so you have to make sure you get everything you need in that time period. Run out of time, and the annoyed, trident-wielding Poseidon will confront you—and he isn't interested in discussing excuses.

Though combat requires only moderate arcade expertise, it's difficult to win chiefly because you have only one life. That's right—when Hercules bites the dust, you're back to square one. In a game like this, you should have an opportunity to save and restore games—say, at the map screen. Still, restarts are rapid, especially when running the game from a hard drive, and the action is addictive. If you enjoy hands-on hack-and-slash adventuring, go ahead and quest for immortality, fun, and *Ancient Glory!*

MEGABUFF PRINTER BUFFER CARD

★★★

Sequential Systems

$129.99

Requires ImageWriter II printer

Reviewed by Vern L. Mastel

The Apple Imagewriter II has been the workhorse printer of the Apple II community since 1985. It's still an excellent printer, but we all wish it would go faster—or at least, free up our computers while it prints out our documents. Software solutions like WaitLess and Express help, but hardware offers a better solution. Apple sells a 32K buffer for the Imagewriter. It helps, but, if you use *Platinum Paint* and other graphics-based productivity programs, or generate large *Appleworks 4.0* text documents—well, 32K is not nearly enough! Especially on the IIGS, where nearly every program print using the printer's graphics mode, which requires much more data than text mode.

A printer buffer is memory dedicated to holding the information sent to the printer by the computer. The buffer holds the information to be printed until the printer is ready for it, freeing your computer for more interesting tasks. Of course, if the print job is bigger than the buffer, the computer still has to wait for the printer, but otherwise, you can work while your printer pounds away.

MegaBuff boasts a full megabyte of onboard RAM—it's 32 times the size of Apple's buffer! It's big enough to hold several super-high-resolution pictures, or, of course, loads of AppleWorks text. Even better, it can be used two ways: as a standard serial interface fed by one computer, or as a buffered Localtalk network interface allowing multiple computers to feed one or more printers.

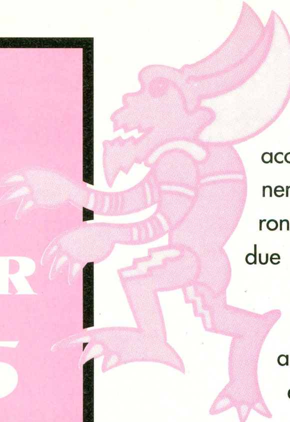
Installation is simple, but does require that you open the printer. There are four DIP switches on the card to configure network or serial mode, baud rate, and a command function (necessary when printing with a color ribbon or using ProDOS 8 software). The directions supplied by Sequential are adequate, but there are no illustrations to guide novices. Unless you are comfortable with taking apart your computer equipment, I recommend you get a capable local vendor or hardware-wise friend to help install the board.

Sequential Systems includes driver software for GS/OS applications like *GraphicWriter III.* There are two versions of the GS/OS driver: one for ProDOS 8 and GS/OS program users, another for those who use GS/OS exclusively. Installation of the driver is just a matter of copying the driver into the Drivers folder on your boot disk. Because I print from both ProDOS and GS/OS programs, my timing tests were done with the dual-mode drivers.

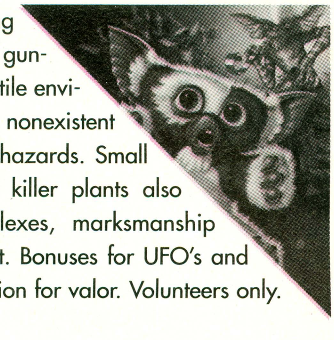
To check MegaBuff performance, I printed documents created with each of four popular applications—*Appleworks GS, Platinum Paint, Graphicwriter III,* and *BeagleDraw*—using various Imagewriter driver, port driver, and hardware setups. MegaBuff, with the Sequential Systems *SpeedPort* driver, significantly reduced wait Time when compared to standard hardware and printer port settings. For the average document, you can expect to

Continued on page 62

APPLE II GAME CLASSICS
$9.95 EACH OR 3 FOR $19.95

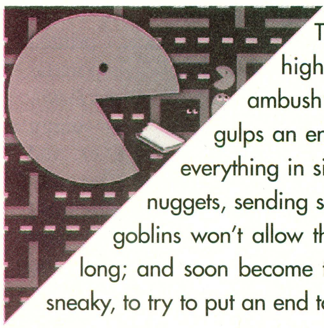


GREMLINS

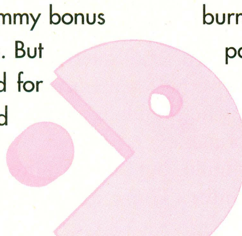


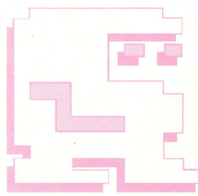
Applications being accepted for replacement gunners in high-risk job. Hostile environment. Road conditions nonexistent due to meteor and crater hazards. Small native population of killer plants also reported. Quick reflexes, marksmanship and diving skills a must. Bonuses for UFO's and enemy tanks. Recognition for valor. Volunteers only.

PAC-MAN

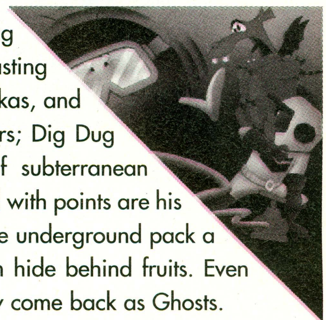


The refinements of dot chomping lead to high scores as hungry Pac-Man avoids ambush by voracious goblins. When Pac-Man gulps an energy dot he can turn the tables and eat everything in sight. . . that includes yummy bonus nuggets, sending scores into the thousands. But goblins won't allow themselves to be gobbled for long; and soon become their old selves, fast and sneaky, to try to put an end to Pac-Man's three lives.





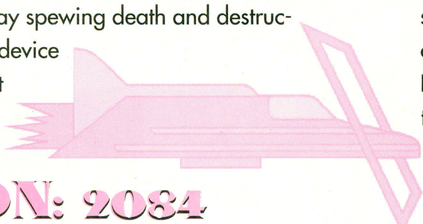
DIG DUG



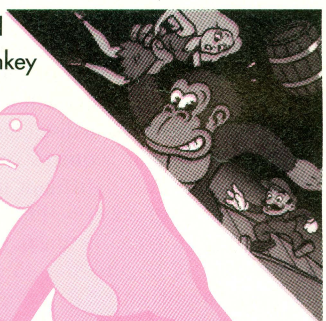
Dodging and blasting Pookas, and dropping rocks on fire-breathing Fygars; Dig Dug burrows his way through a maze of subterranean paths. Ripe fruits and veggies, loaded with points are his passion. But the evil denizens of the underground pack a potentially lethal wallop, and can hide behind fruits. Even when Dig Dug kills them they may come back as Ghosts.

STARGATE

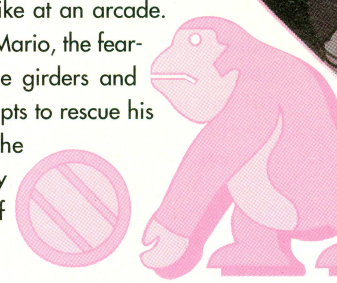
The entire universe is your enemy as you struggle to rescue humanoids stranded on the planet surface. To take them into a Warp you must reach the Stargate. But getting there isn't easy. Yllecian space guppies, Dynamos, Space hums, Phreds, Big reds, Munchies, landers, Baiters, Pods and Swarmers block the way spewing death and destruction. Will your cloaking device protect you from the threat within. . . Mutant humanoids?



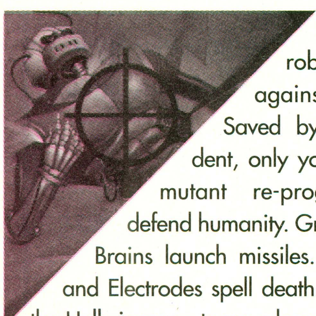
DONKEY KONG



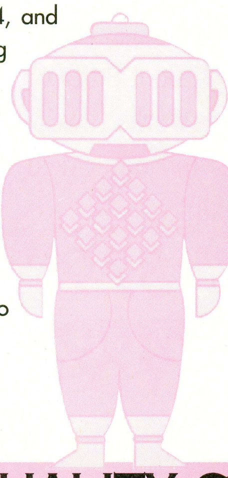
You can feel an excitement tingle up and down your spine when you play Donkey Kong at home, just like at an arcade. Your joystick guides Mario, the fearless carpenter, up the girders and elevators as he attempts to rescue his sweetheart from the clutches of Donkey Kong. All the thrills of the arcade game.



ROBOTRON: 2084



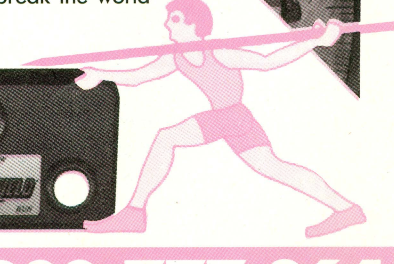
It's the year 2084, and robots are turning against their masters. Saved by a genetic accident, only you can resist their mutant re-programming and defend humanity. Grunts close in. The Brains launch missiles. Tanks, Sheroids and Electrodes spell death. And then there's the Hulk–immune to your laser. Your mission is to rescue, evade and destroy. Good Luck.



TRACK & FIELD

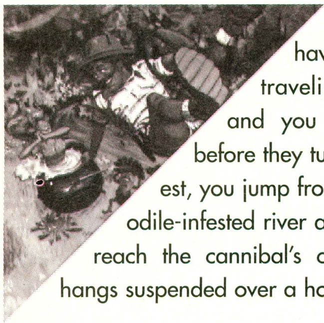
You've worked long and hard to make it this far. Now it's time for head-to-head competition in the 100 meter dash, long jump, javelin, 110 meter hurdles, hammer throw or high jump. You're out to beat the best times and distances on record. Included is a special arcade controller which gives you everything you need to break the world record in athletic competition.

COMPLETE WITH ARCADE CONTROLLER



APPLE II GAMES SPECIAL

JUNGLE HUNT

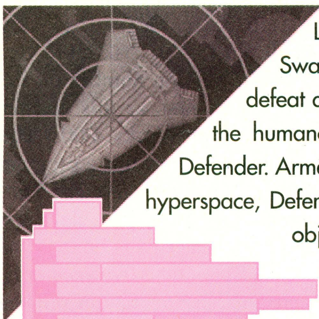


Savage cannibals have kidnapped your traveling companion, and you must rescue her before they turn her into stew! In the deep jungle forest, you jump from rope to rope. Then you brave a crocodile-infested river and a landslide of huge boulders. You reach the cannibal's campsite just in time–your sweetie hangs suspended over a hot cauldron of boiling goo!

GALAXIAN

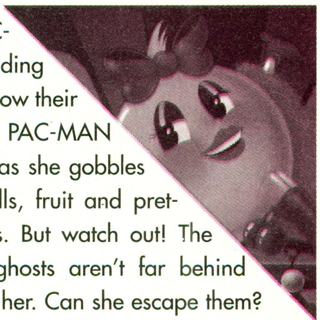
You feel that spine-tingling exhilaration every time you play GALAXIAN in an arcade. Now the same sensation is yours at home. Wave after wave of Drones, Emissaries, Hornets and Commanders come winging in from deep space. Skillfully you slide your ship right and left with your joystick, dodging their fire and blasting them out of the universe.

DEFENDER



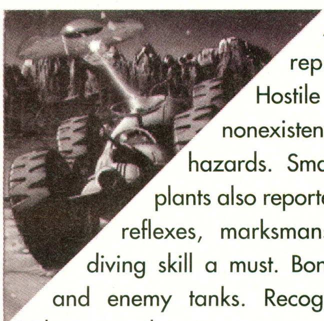
Landers, Bombers, Baiters, Pods, and Swarmers. The alien attack has come, and defeat at the hands of crazed invaders threatens the humanoids. Their only hope is the spaceship, Defender. Armed with smart bombs and able to shift into hyperspace, Defender evens the score only to become the object of another foul attack: kidnapped humanoids transformed into killer mutants.

MS. PAC-MAN



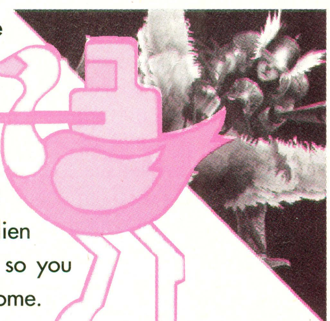
It was the love match of the century, PAC-MAN, star of the arcade, and his leading lady the unforgettable MS.PAC-MAN. Now their romance continues. You guide MS. PAC-MAN through four different mazes as she gobbles up dots, energy pills, fruit and pretzels. But watch out! The ghosts aren't far behind her. Can she escape them?

MOON PATROL



Applications being accepted for replacement gunners in high-risk job. Hostile environment. Road conditions nonexistent due to meteor and crater hazards. Small native population of killer plants also reported. Wuick reflexes, marksmanship and diving skill a must. Bonuses for UFO's and enemy tanks. Recognition for valor. Volunteers only.

JOUST

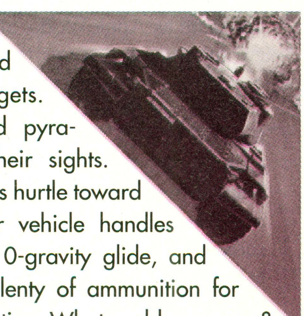


In days of olde, when knights were bolde—they never saw anything like this! You don your helmet, hoist your lance and mount your ostrich to do battle with the evil Buzzard Riders in deep space! Pterodactyls to the right of you, alien eggs to the left–learn to fly so you won't die so very far from home.

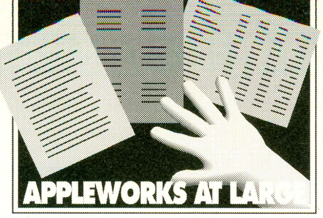
CENTIPEDE

An insidious invasion of multiplying insects (centipedes, jumping spiders, poisonous scorpions, and frenzied fleas) pose different perils to the mushroom patch. You must repeatedly blast enraged creepers and stubborn obstacles or lose your enchanted fungus. Remember to listen for distinctive sounds of the attacking bugs; and watch out for blasted centipede segments, each one grows a new head.

BATTLEZONE



You have full directional control through an entire landscape filled with hazards and targets. Tanks maneuver around pyramids to get you in their sights. Guided missiles hurtle toward you. But your vehicle handles like a dream on 0-gravity glide, and you've got plenty of ammunition for saucer hunting. What could go wrong?



MasterWorks Part 1: Mastering AppleWorks 4

by Beverly Cadieux

Beverly Cadieux writes the AppleWorks 4-related newsletter, TEXAS II. This series of articles is based on issues of TEXAS II which ran from July 1993 through early 1994, and adds new material based on what we've learned since AppleWorks 4 was released.

AppleWorks 4 is a new and impressive upgrade to the remarkably durable integrated program. Programmed by Randy Brandt and Dan Verkade, who wrote many of the Beagle Bros and JEM Software enhancements for AppleWorks, this upgrade has TimeOut and UltraMacros built-in, and includes many features from other products that were previously available only as external add-ons before.

Although 256K is recommended, AppleWorks 4 still runs on a 128K Apple II with a single 5.25" disk (Desktop storage is limited to about 21k in this minimal configuration). It comes on two 3.5" disks, or four 5.25" disks. The 3.5" disk set includes more sample files, most notably the source code for the built-in macros included with the AppleWorks 4 package. If you can use 3.5" disks, get the 3.5" disk set. (New owners receive both disk sizes; updaters receive either 3.5" or 5.25" disks).

The startup process is the same as for earlier versions of AppleWorks—allowing preloading and loading of inits and TimeOut accessories—but ending at the familiar Main Menu. At first, AppleWorks 4 looks and feels much the same as AppleWorks always has, but there are many new features that are perfectly integrated into the program.

There are three desktops—which can hold a total of 36 files—and three clipboards, one for each module of AppleWorks. There is a screen blanker, mouse control, and clock support; and an OA-H Hardcopy command that allows screens to be captured to the clipboard as well as to the printer. Options such as dictionary copying and periodic automatic saving of files are fully configurable. Many of the TimeOut accessories that you already own will work "as is" with AppleWorks 4; others can be updated to AppleWorks 4 compatible versions using the included Updater. More TimeOut accessory updates will become available as time permits the programmers to prepare them.

Here's a summary of what has happened since AppleWorks 4.0 was released in late October, 1993:

November, 1993: An update to AppleWorks 4.01 became available within a few weeks of the original release. This update fixed some bugs having to do with the clipboard and data base, and properly enabled the auto-save timer. Quality Computers shipped a free copy of version 4.01 to everyone who ordered AppleWorks 4. To find out what version of AppleWorks you now have, go to the bottom of the Main Menu help screen. It should say, "Copyright Quality Computers 1993 v4.01"—or later. If it still says "v4.0" after the copyright message, look at your master disks to be sure that you really have the old version, then check with Quality about obtaining a more up-to-date version.

January, 1994: AppleWorks 4 author Randy Brandt publicly released a free patching program, called *RFP,* to online services and user groups. *RFP* applies popular utility and cosmetic patches formerly found on the Beagle Bros patching program, *Companion Plus,* to AppleWorks 4. Patches include canceling the OA-H (Hardcopy) form feed, having the strikeover cursor active on startup, using check marks instead of "—>" in menu selections, and changing the Word Processor Return symbol to a MouseText bent arrow. *RFP* lets you switch OA-S with OA-Ctrl-S so that OA-S always saves the file to the original directory, and reverse Yes/No questions so that Yes is the default. These, and many of Randy's other patches, are now available for the price of downloading on online services, or through your local Apple user group.

February, 1994: A second update, AppleWorks 4.02, became available. A complete set of AppleWorks 4.02 disks can be ordered directly from Quality Computers. Those who prefer to perform the update themselves (it'll save a few bucks) can obtain a patching program called *Make402,* available on the online services and through your Apple user group. Written in Applesoft Basic, *Make402* quickly updates your AppleWorks 4.01 disk set to version 4.02. (Check the version number on the Main Menu help screen first, as this updater does not work with AppleWorks 4.00.) The *Make402* updater disk also contains a program to let AppleWorks copy the *TimeOut Thesaurus* dictionary to a RAM Disk. A new version of the TimeOut Updater fixes bugs in the original updates to *TimeOut Measurements, SuperForms,* and *SideSpread.* If you ran the TimeOut updater when you first got AppleWorks 4, you should run it again on your original TimeOut applications to be sure you have the latest updates. If you have a version of *UltraMacros 4* older than v4.3, you can enable macro recording and Debug by running the TimeOut Updater with the old *UltraMacros 4* accessories in the TimeOut directory.

A persistent problem with interrupts causing a "Restart System" message while UltraMacros is active is not fixed with the version 4.02 update. This problem is confined to the IIGS and occurs only if an external screen blanker or on-screen clock (not the AppleWorks clock) updates itself at the moment UltraMacros is accessing auxiliary memory. If a fix for this conflict ever does come, it will involve major changes in the UltraMacros program code. There are many good workarounds for this problem now; most users feel that the advantages of having UltraMacros are worth taking precautionary steps. Many users have reported that ProSel-16-launched AppleWorks 4 runs free from system crashes caused by interrupts. Other workarounds include setting programs that cause interrupts to avoid text-based programs, or simply starting AppleWorks from ProDOS 8.

April/May 1994: A new set of TimeOut updates are made available. These updates include most of the programs from the old *DeskTools, DeskTools II,* and *PowerPack* collections from Beagle Bros, now marketed under the title *DeskTools IV.* If you already owned one of the *DeskTools* package, or *PowerPack,* patches to update *TimeOut Directree,* *Calculator Plus, File Librarian, Help Screens, Program Selector, Line Sorter,* and others can be obtained (upon release) through online services and Apple user groups.

A new version of AppleWorks, containing minor bug fixes, is tentatively scheduled for the summer of 1994. To stay informed about current version numbers and upcoming releases, subscribe to an AppleWorks 4-specific newsletter like *TEXAS II. TEXAS II* keeps in touch with AppleWorks developers, and tests and reviews new products carefully—and was the only publication reporting on the progress of AppleWorks 4 as it was being developed.

In this series, we'll discuss AppleWorks 4 in detail, looking at menus, commands, and techniques in a way that will help you make the most of its new features. We'll even highlight some "little-known" aspects, including some that the manual doesn't point out. Many of these suggestions are real time-savers, so try them out, use them often, and don't forget they are there!

SHORTCUTS AT THE MAIN MENU: ADDING FILES

As we all know, Return on "1. From the current disk" goes to the list of files, which now shows up to 255 files, instead of 170 under AppleWorks 3.0.

OA-Return on "1. From the current disk" gives a prompt, "Enter Filename." Here, you may type the name of the file you want, and the file is added to the Desktop without going to the file list. This shortcut may be of little value if you use long filenames, or if you can't remember the name of the file you want, but it does save time, and is especially useful in macros.

Return - on "2. From a different disk" gives the change disk menu. Once in the change disk menu, you can press Return to select a device, or <OA-Return> to list all the subdirectories on the current disk or in the current directory. Some other shortcuts in the add files menu include:

OA-A or OA-. also allow you to select a subdirectory.

OA-D or OA-, drops a directory.

OA-? shows the volume names in each device.

OA-P lists your personal pathname list, which is configured under Standard Settings.

OA-1 to OA-8 changes to one of your pathnames, without seeing the OA-P list first.

Once you are in a list of files, ready to add a file to the Desktop, the number of files already on the active desktop is displayed. If the first desktop already has close to twelve files, you can switch to the next desktop with OA-D. Each file's locked and backup status is shown with an appropriate "*" or "+" symbol. In addition to AppleWorks Word Processor, Data Base, Spreadsheet, and Subdirectory files, the file listing now shows Text files, which can be selected and loaded as though they were regular Word Processor files. (The original AppleWorks 4 release marked text files loaded in this way as "New," forcing you to make a conscious decision to either save them or remove them from the Desktop even if the file wasn't changed. Starting with version 4.02, they're marked the same as regular word processor files added this way: "Unchanged.")

While in any file list (not just Add Files):

OA-Y ejects the current disk if it's a 3.5" disk.

OA-A sorts the list by date, file type, size, or alphabetically.

AppleWorks files from earlier versions load "as is," and are ready to use with AppleWorks 4—with a few minor exceptions: if you have been using AppleWorks 2.0, 2.1, or an even earlier version, word processor files may load with the cursor on a unreasonable line number, such as line 7842. To make sure AppleWorks 2.0/2.1 word processor files load with the cursor at the top of the file, set "Reset files when loading" under AppleWorks 4's Miscellaneous Standard Settings to "Yes" (the default is "No"). This setting forces the cursor to appear in the proper position at the top left of the file. If you never have to load AppleWorks 2.x word processor files, you may set "Reset Files when loading" to "No," and the cursor will appear in the same position and zoom mode as when the file was last saved.

The other exception in loading older AppleWorks files is record selection rules attached to data base reports, which are reset to "Selection: All records." (Throughout much of beta testing, data base report formats from earlier files were not retained at all! It was near the end of development that the programmers made the time in their already pressing schedule to add the conversion older report formats.) Calculated categories in reports must be redefined under Review/Add/Change, which now allows formulas and calculations, including date math, in the data base itself. With these two minor exceptions, files from earlier versions of AppleWorks load and work perfectly in AppleWorks 4.

AppleWorks 4 word processor and spreadsheet files are backward-compatible with AppleWorks 3.0. Once a data base file is resaved under AppleWorks 4, however, it becomes an AppleWorks 4-specific file and cannot be loaded by earlier versions of AppleWorks. (In some cases, it will appear to load, but will not display.) After a few weeks of using AppleWorks 4, you won't feel the need to go back to an earlier version. If you must load an AppleWorks 4 data base into AppleWorks 3.0 or earlier, you can go by way of a text (ASCII) file printed to disk.

SHORTCUTS AT THE DESKTOP INDEX

If you are loading only one file, AppleWorks displays the file on the screen. If you are loading more than one file, AppleWorks displays the Desktop Index, or OA-Q "QuickSwitch" menu, so named because this list of desktop files is accessible anywhere in AppleWorks simply by pressing OA-Q. The OA-Q index is in itself a shortcut for the Main Menu selection, "2. Work with one of the files on the Desktop."

```
          Desktop Index 3
 ------------------------------
 1.  Students                DB
 2.  TEXAS.II.v4.2           WP
 3.  E.Mail                  WP
```

While the OA-Q index is on the screen, a number of new open-Apple commands are available. These Open-Apple commands allow movement throughout the AppleWorks program, bypassing the Other Activities and other menus:

Tab displays the OA-Q desktop index for the next desktop. Tab also works to advance to the next desktop in the Save Files menu.

OA-1, OA-2, or OA-3 at the OA-Q index pops up the OA-Q index for the appropriate desktop, allowing you to jump to any Desktop instantly.

The sequence OA-Q OA-V allows you to view a list of the files on all 3 desktops at once. (This list doesn't permit you to select a file from the displayed Desktops, but a macro

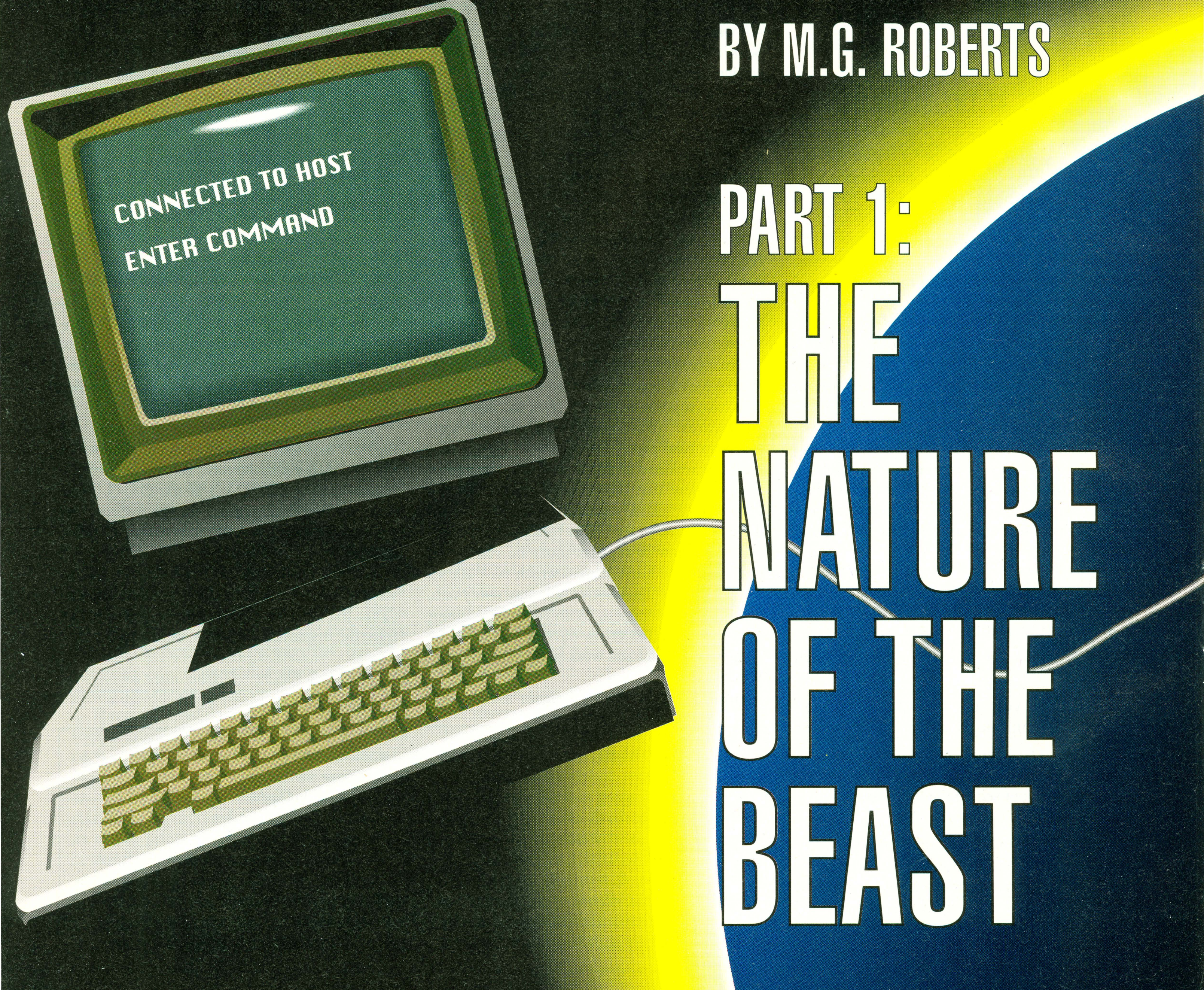
Continued on page 27

```
Disk Activities       <— Tab —>       File Activities

1.  Copy a disk                       1.  List all files
2.  Erase a disk                      2.  Copy files
3.  Verify a disk                     3.  Move files
4.  Format a disk                     4.  Install Inits or TimeOut files
5.  Compare disks                     5.  Rename files
6.  Rename a disk                     6.  Delete files
7.  Copy a subdirectory               7.  Lock files
8.  Create a subdirectory             8.  Unlock files
```

THE INTERNET:
THE CREATURE THAT HAD TO BE

BY M.G. ROBERTS



PART 1: THE NATURE OF THE BEAST

O, this is the animal that never was
They hadn't seen one;
but just the same, they loved
its graceful movements, and the way it stood
looking at them calmly, with clear eyes.

It had not been. But for them, it appeared
in all its purity. They left space enough.
And in the space hollowed out by their love
it stood up all at once and didn't need

existence. They nourished it, not with grain,
but with the mere possibility of being.
And finally this gave it so much power

that from its forehead a horn grew. One horn.
It drew near to a virgin, white, gleaming
and was, inside the mirror and in her.[1]

The information age was nearly 70 years away when Austrian poet Rainer Maria Rilke penned these enigmatic lines about the unicorn, a fictitious animal which nonetheless carved out an existence in our psyches through our sheer will to believe that it could exist. And now, 70 years later, we've nurtured the dream of an information society, not knowing just what form it would take, but certain that it would come to exist and that we would love it. Slowly that dream has become something more tangible. If ever there grew a real golden horn from the forehead of a dream, it was the Internet, the creature that had to be.

Internet History

A product of U.S. Government demands for scientific research in the 1960s, the Internet was originally the ARPAnet (named for the Department of Defense's Advanced Research Projects group), tended in spare offices and closets with spare time and spare funds. It was a researcher's dream come true. Databases, software libraries, news, idea exchanges, communications, education, Library of Congress catalogues, information on anything known to mankind—all available from any computer terminal. The intellectual freedom was addictive. So while the ARPAnet was serving its official functions, its users were dreaming of making that same power available to everyone—and not at the whim of some government agency.

One by one, smaller campus networks at universities and research facilities began to connect to each other, offering access to more and more people. The word "internet," which refers to a group of smaller independent networks connected together, gained a capital letter and came to refer to the worldwide community of corporations, universities, government agencies, and other institutions who were participating in this informal project. By the mid-1980s, the Internet was clearly evolving into a prototype for the Clinton administration's much-publicized "Information Superhighway."

The Internet is a hybrid creature, in reality more like a camel than a unicorn, with the trappings of committee origin. While there have, in fact, been official committees to shape it, develop its standards, and govern it, they have not been very successful ones. The Internet was built largely by people with their shirt sleeves rolled up toting power drills and unrolling cable—connecting their offices, laboratories, and classrooms and coaxing their computers to talk to each other, regardless of size, model, or format. The Internet's credo is simple—"information wants to be free"—and the architects of the Internet have devoted their lives to providing the world with ready access to information.

Unlike commercial information services, the Internet is not one computer. It's not even one network.

And so the Internet grew without plan but with a clear sense of purpose. Its history is one of trial and error, bridges and patches, arguments and compromise. But in the end, progress always won, for the dream is too strong to be ignored for long.

The Federal bureaucracy, even before the Clinton administration, participated in the Internet's growth—albeit reluctantly—by granting funds to pay for wide area network links and routing hardware. The National Science Foundation funded and oversaw much of the Internet's development since the mid-1980s, because it needed to connect its new supercomputer centers with a reliable network. Under the NSF, the advance of the Internet has been more smoothly orchestrated than before, but the actual force behind the growth has largely been small networks of computers across the country whose users demanded connection with other networks.

The development of Internet standards underscores the dynamics of the network's bureaucratic top-down "management" and its grass-roots bottom-up evolution. An example: the Organization for International Standardization—an administrative body based in Switzerland—worked for years to develop standards and protocols for computer connections and file transfers, but were almost secretive about their work until Carl Malamud, a well-known technical writer and networking expert, talked them into publishing their work. But shortly afterward, they withdrew their publication, partly out of fear of losing control of the profit potential. They believed the standards they developed would generate huge amounts of money as companies around the world paid a king's ransom for the information. But the pioneers would have their Internet even if they had to develop their own standards—which, thumbing their noses at the royalties demanded by the official standard, is exactly what they did.

As quickly as the bureaucrats came up with ways to make money from the Internet, the Internet community sidestepped their schemes. Information wants to be free, and free it would be. Local network administrators—usually volunteer university or research staff with full-time responsibilities elsewhere—discussed their problems with each other, and with phone companies and computer companies, then wrote the programs that made the Internet work. Although not always elegant at first, the programs gained strength through evolution—failures in day-to-day use sent them back to their designers for improvement.

Now the official standards are more or less complete, created largely in an ad-hoc manner. The Internet has "grown up" with a strong sense of independence and disregard for central authority. So if the Internet seems a little disorganized, decentralized, and difficult to figure out, that's only because it is. That information well is out there waiting to be tapped, but it doesn't just knock on your door and invite itself in.

Unlike commercial information services, the Internet is not one computer. It's not even one network. It's thousands of separate networks, and tens of thousands of computers. Each network and computer you "visit" while touring the Internet is owned and operated by a different group of people. There is really no centralized authority for the network, no 800 number you can call when you have problems. But the Internet more than makes up for its lack of organization with its sheer volume and variety of information resources.

The Apple II and the Internet

The Internet was built mostly for mainframes and supercomputers, but the dream of an information society would not be satisfied until the network's tentacles stretched to desktops. First came workstations—desktop machines with the power of yesterday's minicomputers—and now, it is entirely possible to

[1]The Selected Poetry of Rainer Maria Rilke, translated by Stephen Mitchell. New York: Vintage International, 1989. p 241.

connect even lowly personal computers directly to the Internet, although the Apple II does not currently have this capability.

But even though you can't connect directly to the Internet (a process which requires specialized hardware and software), you can always use your Apple as a terminal and connect, via modem, to a computer which is on the Internet. Or, failing that, to a bulletin board system which has an Internet mail gateway. Even though a mail gateway is far removed from true Internet access, there are still plenty of resources available even through this limited connection.

Who is the Internet for?

Is the Internet really for everyone? No—or, at least, not yet. Some efforts have recently been undertaken to make the Internet easier to use for the uninitiated, but most of it is still uncensored, disorganized, and lacking in standardization.

Is the Internet for everyone who shares the dream of instant information access—and is willing to expend the effort required to make the dream a reality? The answer to that is an unqualified *yes.*

What do you really want out of your computer? Is telecomputing and computer communication foremost on your list of priorities? Is it a major part of your life? Or would you be content with a chat line, a bulletin board, and some file libraries? If the latter is the case, then the Internet might be interesting to watch, but it might also be a waste of time. To wring useful information out of this ever-changing web of millions of computers—spread across more than thirteen thousand individual networks—frequently requires a hard-won virtuosity and determination the casual user may not muster.

On the other hand, there is much for the casual user, most of it accessible with moderate effort. Usenet newsgroups, which resemble forums on commercial online networks, carry discussions on literally thousands of topics. Some, like the computer discussion groups, are immediately and obviously useful; some, like the Urban Folklore newsgroup, are just plain fascinating to browse. Other newsgroups are full of frivolous discussion, and still others push the boundaries of good taste. (Of course, tastes vary—no two people will consider the same newsgroups frivolous or tasteless.)

Internet e-mail is one of the crown jewels of the Information Age. Virtually anyone who is connected to a wide-area network of any kind can be reached via the Internet, usually within seconds. There are even gateways which allow you to reach people on other networks, such as commercial information services and smaller bulletin board systems. *You can even send faxes through Internet e-mail!*

And if you like chat lines, Internet Relay Chat (IRC) carries not just a few channels, but more than a thousand channels where users can meet others with similar interests in "real time." As with newsgroups, discussion ranges from the serious to the ridiculous.

Since the Internet is not run by any one organization, there are very few rules about what sort of material is acceptable and what is not. (There are, for example, newsgroups for S&M and other non-mainstream lifestyles in the "alternate" newsgroup hierarchy—and that's only one of the things on the Internet which may shock or amaze you.) There are rules of etiquette that keep discussions on-topic, but these rules are enforced by the community itself, not by any central authority. Those who violate the rules are shunned or ridiculed by others, and, in extreme cases, can even have their Internet access revoked if enough users complain to the offender's service provider.

If you're a casual telecommunications user, your needs may easily be met by one of the commercial online services like GEnie, CompuServe, or America Online. Delphi, one of the smaller online services, actually offers inexpensive direct Internet access in addition to its usual services, giving you the best of both worlds; America Online promises to follow their lead soon.

Using the Internet is like a trip to another country

And prices on basic Internet connections are falling rapidly. Those who just want e-mail, newsgroups, and Internet Relay Chat can often get an account on an Internet-connected system for less than the cost of an account on a commercial online service! And many of these systems are paying more attention than ever to user interface, making it easier for neophytes to get their feet wet.

So, while the Internet is a vast and often confusing computing resource, don't let that stop you. The Internet is getting easier to use all the time, and you shouldn't let yourself be discouraged from getting an account just because you don't have the need, patience, or ability to use all of its services. Casual usage is quite acceptable to Internet denizens; you don't have to be a computer scientist to cruise the Information Superhighway.

Getting Serious

Using the Internet is like a trip to another country—it requires study and preparation. When planning to visit another country, you might first pick up a phrase book, buy a travel guide, and maybe ask people who had been there for suggestions and tips. The same is true of your journey into the Internet. While this series attempts to give you an idea of what to expect on the Internet, it is by no means complete!

When you connect to the Internet, you leave the security of your Apple and the operating system you call home. You are now in a world of mainframes, workstations, PCs, telephone links, satellites, routers, bureaucracy, inefficiency, pleasant and unpleasant surprises, friendly people, hostile people, and people who don't even speak your language (whether that's English or Applesoft). If you think you can venture into this world without a guide, you err. It would be irresponsible, arrogant, and short-sighted to tell you otherwise. There are dozens of books about the Internet. Some of them are written by the people who created it. They are not just good—they are essential. Don't leave AppleWorks without them.

There are very few "roadmaps" written specifically for Apple II users. The only one I know of is the "Frequently Asked Questions" (FAQ) file for comp.sys.apple2, the Apple II users newsgroup. Naturally, since this FAQ is floating around in the Internet, you have to be fairly adept at navigating the network before you can even find it.

Therefore, general Internet books are essential to understanding how to use the available resources. The Internet doesn't care what computer you're using. You'll probably want to pick up two or three books (see sidebar), then get a connection as soon as possible so you can apply what you read. The books, though excellent roadmaps, are frequently hard to follow unless you can associate some landmarks with the jargon. Don't expect to be able to learn all about the Internet from books before you even log on—that would be like trying to learn all about France from a street map of Paris. Learn what you can from the books, then explore a little, then read a little more.

Finally, if you're terrified of asking questions for fear of appearing uninformed, get over it. The Internet requires you talk to people. You will need some support—even if you consider yourself a computer expert. You may even need to call or meet with your system administrator if you will be part of a local network and need to know things unique to that network. Find a user group. If you live, say, in the very center of Wyoming and have to travel 50 miles for groceries, you will probably be making some long-distance calls. The Internet is built on individualism, but everyone's initiation involves looking like a complete novice until you make some contacts and read the books.

More To Come

So far we've told you what the Internet is and how to find out more about it. In the next installment of this series we'll discuss how to get yourself onto the Internet and start using it, then, in the final installment, we'll take a look at the tools you'll use to explore the Internet. Stay tuned! ■

Essential Guides

The best bookstores carry dozens of books about the Internet. You certainly don't need all of them, but some of them are virtual necessities. All the books below will increase your understanding of this new medium, though some are optional.

The Whole Internet User's Guide & Catalog
Ed Krol
O'Reilly & Associates (1992)

First on anyone's Internet information shopping list should be The Whole Internet User's Guide & Catalog. Krol goes back a long way with the Internet, and is currently Assistant Director for LAN Deployment in the Computing and Communications Service Office at the University of Illinois, Urbana. He was an insider during the creation of the NSFnet—part of the current incarnation of the Internet. Another trustworthy—and free—guide by the same author is *The Hitchhiker's Guide to the Internet,* which can be found online at various FTP sites. Like the book, it's excellent reading for the novice and expert alike.

Krol's text combines a clear writing voice and an obvious knowledge of his subject matter. When you don't understand something on the Internet, you can count on Krol to explain it in a way that makes it seem obvious but is never patronizing. This book will be your friend for a long time.

Doing Business on the Internet
Mary J. Cronin
Van Nostrand Reinhold (1994)

Doing Business on the Internet is an excellent guide for people who want to know what the Internet can do for their company—or for themselves. Cronin is University Librarian at Boston College, and teaches information management at the Boston College School of Management. With this kind of information background and experience, it's no coincidence that Cronin's book is wonderfully organized, well-indexed, and informative in the extreme. To answer questions on using the Internet for practical business purposes, she first had to find those questions and ask them—something no one else had bothered to do. She leads the field in actually putting the Internet to use. If you have ever asked yourself why you might find the Internet useful—even profitable—this is the book for you.

Internet: Getting Started
April Marine, Susan Kirkpatrick,
Vivian Neou, Carol Ward
PTR Prentice Hall (1994)

This book is chock full of basic information, including lists of Internet service providers throughout the US and around the world. It also has indexes of RFC's, or Requests for Comments—like FAQ's in that they address specific information on specific subjects, and among the most useful sources of information on the Internet. There are thousands of RFCs; indexes like this one are most helpful. While you might find most of this information in other books, this one is superbly organized.

The Internet Directory
Eric Braun
Ballantine Books (1994)

As you might expect, this is a book of lists. FTP sites, mailing lists, Usenet Newsgroups, library catalogs, Archie servers, Gopher sites, and more, more, more. You'll never use the Internet to the fullest if you don't know what's out there or how to get it. This book makes it work.

Exploring the Internet
Carl Malamud
PTR Prentice Hall (1993)

Carl Malamud did an amazing thing: he traveled around the world three times in search of the Internet. He interviewed the people who rolled up their sleeves and laid the cable. He interviewed techno-hermits who rarely came out of their offices to speak to normal people. He documented how the Internet is used in various parts of the world. He got the ISO to publish their Internet standards (hundreds of thousands of pages) so that the world could begin to use them. Exploring the Internet reads like a combination travelogue, spy thriller, and propeller-head dictionary of acronyms. (Malamud is a fun guy to go exploring with, but I wouldn't want him for a roommate.)

For anyone who has the patience to read page after page about things you've probably never heard of in your life, like INRIA, X.400, MAVROS, GIPSI, CNET, X.25, SMTP, and ANS.1, the ultimate reward will be a deeper understanding of the Internet than is available from any other book.

How the Internet Works
Joshua Eddings
Ziff-Davis Press (1994)

Once in a while you come across a book that cannot be described adequately with mere words. Appropriately, *How the Internet Works* is graphically oriented. You might say it is the Internet Picture Book. But far from being a children's book, How the Internet Works examines each concept of the Internet's structure, mechanics, and function and explains them all in crisp, readable prose, enhanced by informative full-page graphics. If you really want to know how TCP/IP works, or what a T-3 line does, how files are compressed, how the Internet is connected together, and any of a hundred other things, this book will make you an Internet expert beyond your wildest dreams—and make it seem fun and easy in the process. I don't think I ever really understood the Internet until I got this book. Highly recommended.

MasterWorks
Continued from page 23

included with the built-in default macro set provides exactly that function).

OA-Q OA-D and OA-Q OA-F move directly into the disk or file utilities menus.

DISK AND FILE ACTIVITIES

Some of the best new features of AppleWorks 4 are hidden under the Disk and File Activities menus. Disk and File Activities can be found in the Other Activities menu, where "2. List all files on the current disk" and "3. Create a subdirectory" used to be. At first, you may think that having to go all the way into File Activities to list files is an inconvenient extra step. Remember, though, that the <OA-Q OA-D> and <OA-Q OA-F> shortcuts quickly put you into disk and file activities. Once there, you can Tab between the two.

Under Disk Activities, there's an undocumented feature: the AppleWorks 4 disk copy function will copy disks in older Apple II formats such as DOS 3.3 and Pascal—in addition to all your ProDOS disks. Disk copying uses the desktop k available at the time, so if memory is tight, remove as many files as you can to make copying take fewer passes. If you have a large desktop, an entire disk may be copied in one pass. When making more than one copy of a disk, you need only to read the source disk once, then write the data out to as many destination disks as you want. You can then jump into "Compare" mode by pressing OA-N (or OA-Return on "No") at the question "Copy this disk again?"

Also under Disk Activities is a true disk formatter. Unlike earlier versions of AppleWorks, which only created data disks, AppleWorks 4 creates genuine bootable ProDOS disks—just add PRODOS and a .System file. You will also find what just happens to be one of the best disk verification programs available for the Apple II, the "Verify a disk" option on the Disk Activities menu. (After verifying a disk, an undocumented OA-Return keypress allows you to verify another disk in the same drive without having to re-select.)

In the File Activities menu, file lists appear in actual disk order instead by type, as they do in the Add Files listing. However, using OA-A (arrange), any AppleWorks file list may be sorted by type, size, reversed chronological date, and alphabetical name. Directory sorting is for the current display only, and does not save the new directory order to disk. You'll have to sort it again every time if you always want to view files by name, size, date, or type. Instead of the old, generic AppleWorks file type, "Other," File Activities is more specific about listing the file types you need: Basic, Binary, System, Text, Word Processor, Data Base, Spreadsheet, and Subdirectory. Only less-used files, such as IIGS fonts (used with

Continued on page 30

AN INTERVIEW WITH JAWAID BAZYAR

by Tara Dillinger

Jawaid Bazyar is the president of Procyon Enterprises, Inc., an Apple IIGS software developer and publisher best known for its Unix-like GNO/ME operating system—and also a programmer for Sequential Systems, publishers of the disc-Quest CD-ROM access software.

II ALIVE: Let's start with your name. It's very unusual. How do you pronounce it?

BAZYAR: I hear that a lot. It's *Jah-waid Bahz-yar.* Said just like it's spelled, since it's a phonetic spelling of a Farsi name (the origin is Afghanistan).

II ALIVE: Are your parents from Afghanistan?

BAZYAR: Just my father. My mother was born in Southern Illinois, and is mostly of German heritage. I've been told "Jawaid" means "forever", and "Bazyar" means "farmer". My father came to the States as an agricultural student, and since surnames are not common in Islamic culture, he had to choose one when he immigrated. I try not to take the implications of my names to their logical conclusion!

II ALIVE: How and when did you first get interested in computers?

BAZYAR: I first ran into an Apple II+ in 7th grade—that was 1981, I believe. I remember that the computer store still had DOS 3.2 applications on the shelves. My parents always left me to do pretty much whatever I wanted, and I got into all sorts of mischief before I found the computer, but when I did, I knew right then and there I could spend the rest of my life at it. So far, that's exactly what I've done!

II ALIVE: When did you get your own computer?

BAZYAR: In 1983, I think. It was an Apple IIe, of course. I'd begged my parents for several years to get me a computer, any computer, and finally Mom caved in and let me spend some of my college money. I was 14 or 15. I did the same thing everyone else my age was doing with the IIe—playing lots of games.

II ALIVE: How did you get started in programming?

BAZYAR: Around the time I got my computer, I met someone who was running a computer bulletin board system. I set out to write my own BBS, and for several years I was sysop of "The Caverns of QYV." I wrote the BBS software, which I called *ProBBS,* from the ground up. It was the first program I ever wrote that I sold to someone else. I only sold two copies, but at the time I had no idea just how huge the computer market was, so I never tried to get it published anywhere else.

II ALIVE: How much did you sell it for?

BAZYAR: I think I sold one for $100 to a friend of mine, a doctor who was as interested in computers as I was, and who was largely responsible for egging me on and keeping me enthusiastic about the "beast." The other copy I sold to another kid for about the same amount, and I spent many nights at his house trying to get the BBS to work right with his new external 1200 baud "high speed" modem. This was about a year or two after I first got my first computer.

II ALIVE: What was your next project?

BAZYAR: I'd done a lot of BASIC programs. The Beagle Bros stuff was a great inspiration for some of my experiments. I wrote duplicates of some of their software, to improve it and flex my programming muscles. One of my favorites, from a user-interface point of view, was a *Wizardy* character and game editor, which I've never released. Mt.

Vernon, Illinois was a small town, and I didn't have any way to connect to the online services without long distance fees, so I didn't distribute much of my stuff at all.

I guess my next big project wasn't until I got into college and discovered the Internet. The Internet showed me just how vast the computer community, and the world in general, was. I was going to the University of Illinois at Urbana-Champaign, in the middle of nowhere in Central Illinois. No offense to any alumni, it's a good school, but too darn windy. My major was in Computer Engineering, which I got a bachelor's degree in. I was lucky enough to take an "experimental" Computer Science course, a prototype for changes to the Department a professor wanted to make. The language used in the course was Scheme, an AI hopeful at the time. So, I picked up *ORCA/C* and wrote a Scheme for the IIGS. I used it to play a text adventure game I'd written for the class.

II ALIVE: Were you still using the IIe at all?

BAZYAR: Oh, no. In June '87 I put the IIe in a closet and set up my fiery new IIGS. I eventually sold the IIe for $250 (I'd paid $2000, but that's the biz). I was one of the first people I knew with a IIGS—it's a Woz machine. The original dealer demo totally blew me away, and—poof—there went more college money.

II ALIVE: So when did you start Procyon?

BAZYAR: Sometime in 1985. It was just me, with a DBA, but it was my business and I liked it.

II ALIVE: Where did you get the name?

BAZYAR: I was involved with a group of programmers as early as 1982. We had a rather dull name—SICS, Southern Illinois Computer Software. Being teenagers, we didn't accomplish much, and the group fell apart quickly. I'd always disliked that name, but I did like "Sirius" and the like, and so I pored through star charts in an encyclopedia until I found "Procyon". It's a Greek word that means literally "Before the Dog," or alternately "The Dog in the Sky." It's the brightest star in Canis Minor, and rises at night before Sirius, the Dog Star, hence the literal translation. "Sirius" was of course already in use by Sirius Software, one of those great companies from the early days of the Apple II, but I liked the idea of being "before" them even if I started my company later! Some of my older programs floating around have the name "Procyon Software" on them, and all that's from that period.

II ALIVE: So what was your first major product?

BAZYAR: During my senior year (in 1991), Derek Taubert and I embarked on a project we called "Shell Stuff," to bring our favorite Unix utilities to the IIGS under the ORCA shell. We quickly discovered that ORCA couldn't handle a straight port of most of the Unix stuff, because of the differences in the operating systems. We ended up rewriting the first "Shell Stuff" programs completely from scratch, but that would have taken forever.

So I became intrigued with the possibility of porting a small Unix system called *Xinu* to the IIGS. Derek and I spent many late nights working out the fundamental multitasking code—I believe it was all of 2K on disk—and finally got it working. At that point, I realized how useful a IIGS Unix could be, and thought it might even have some measure of commercial success, so a friend and I incorporated as Procyon, Inc. and started working on GNO (which stands for GNO's Not ORCA, sort of a tip of the hat to the GNU people).

II ALIVE: What is so appealing about Unix?

BAZYAR: Unix started as a philosophy, more or less. It grew out of research done at AT&T Bell Labs, and was intended mainly for internal use at AT&T. At the time Unix was created in the 70s, computer operating systems tended to be gargantuan, monolithic, and insanely complex things. This, of course, led to hard-to-fix bugs, user dissatisfaction with the restrictions placed on them by the operating system, and other bad things. Also, since computer hardware back then wasn't particularly speedy, these fat operating systems were slow as well. DEC's VMS, IBM's mainframe operating systems, and the OS that Unix itself was a reaction to—Multics—all suffered from this deficiency of design.

The philosophy of Unix is simple. Instead of the operating system doing everything conceivable, the operating system does very little other than provide basic access to peripherals, and, more importantly, channel communication between programs that are running in the system. This allowed Unix programmers to easily combine small, manageable programs to solve more complex tasks. And if they needed a capability that wasn't provided as part of Unix, they could add their own, because the system was very easily extensible.

II ALIVE: How does that work? Give us an example.

BAZYAR: Take the simple task of viewing information one screenful at a time, a feature useful when looking at text files, or viewing a directory listing—or anything, really. The creators of Unix realized that this feature (and many other features) would be useful in a variety of situations, so they invented the concept of pipes.

So in Unix, if you want to page through the output of any program, all you have to do is pipe it "into" a program called more. So, instead of writing code to handle output paging in dozens of different programs, they only did it once. To see a directory listing a page at a time, you just type *ls | more.* To page through a text file, you type *cat file | more.* It works with any Unix program or command.

This feature was was missing from the IIGS operating system, and adding it opened the door to bringing Unix applications to the IIGS. The program that interprets user commands like ls | more is called a shell, and under Unix, you even have your choice of different shells. All of today's shells (or command line interfaces) can be traced back to UNIX. MS-DOS, ORCA, GNO, AmigaDOS, and on and on.

II ALIVE: MS-DOS is a shell? I never thought of it that way

BAZYAR: MS-DOS has a shell (COMMAND.COM). Compared to Unix, MS-DOS is an abomination, but that's another topic for another time.

II ALIVE: How about ProDOS?

BAZYAR: ProDOS is a little like Unix in that you can use different shells with it. BASIC.System is one shell for ProDOS. There are others, including MicroDot, ECP8, and, my favorite, Davex. For GS/OS there's also ProSel 16, ECP16, and ORCA. Davex is a nifty program—it's fast, it has lots of features, and it's sensible. Dave Lyons, who wrote it, eventually got hired by Apple and was responsible for writing large parts of Systems 5.0.4 through 6.0.1. Dave was also the person who helped me get started doing full-fledged IIGS desktop programming, so his name appears as a "Thanks to" in every Procyon product

II ALIVE: How has GNO fared?

BAZYAR: We first showed GNO at the 1991 KansasFest. We were a little disappointed that we sold a total of four copies. But those sales were to people who were very enthusiastic about the product, and word spread quickly.

That fall, I signed on to GEnie to do GNO product support and promotion. GEnie helped a lot—it provided a lot of press, and put me in contact with knowledgeable Apple II folks like Matt Deatherage, who was a great help in finishing GNO. The GNO we showed at KansasFest was an unfinished version, and we'd seriously underestimated the scope of the project.

I met Dave Lyons for the first time at KansasFest; he sat down at the computer, crashed GNO, and found the problem within five minutes.

II ALIVE: What else are you involved with, besides Procyon?

BAZYAR: I've been doing programming and technical support for Sequential Systems since April, 1993. I've written drivers for their MegaBuff 1 MB buffer card for the ImageWriter II, some other minor programs, and, most recently, *discQuest,* which is a CD-ROM access program. Sequential also publishes Procyon's *Switch-It!*—a multiple-application switcher. And, I might add, my association with Sequential has also resulted in my picture appearing in one of their ads in *inCider/A+!*

II ALIVE: How did you get the job?

BAZYAR: I'd contacted Sequential about becoming a dealer for their RAM-GS memory card, which I wanted to sell with Switch-It!, because it's a real memory-hungry program. After a hectic few months of negotiating I ended up with a publishing contract and a steady source of income from my consulting, which was a nice change after working several months at 7-11!

II ALIVE: How did Switch-It! come about? A tie-in to the Unix idea of multitasking?

BAZYAR: Actually, Switch-It! was pretty much anti-Unix. After GNO was shipping, we noticed that half the people who called for information on GNO thought it was some sort of "MultiFinder" for the IIGS. Obviously, there was a market there, so I started working on *Switch-It!* I wrote the basis of the program in about a week with some help from Bill Gulstad, who had done an experimental "MultiFinder" called *Leapfrog* a few years earlier. His help got me over a few hurdles that would have taken a long time without his assistance.

II ALIVE: With *Switch-It!* you went head to head with Seven Hills' *The Manager*. Who won?

BAZYAR: Head-to-head is a pretty accurate description, yes. I found it exhilarating. All of a sudden, people were talking about competition in the Apple II market! I'll refrain from comments on who won, other than to say that *The Manager* was obviously released before its time to avoid complete market domination by *Switch-It!*. My favorite point in favor of *Switch-It!* is that it works flawlessly under GNO, which is very convenient.

II ALIVE: Does *Switch-It!* do true multitasking?

BAZYAR: No, GNO is the only IIGS product which does "true" (pre-emptive) multitasking in the sense we computer science types use the word. Switch-It! is "just" a switcher, but in real life, that's just almost as useful (and simpler to do) than the "cooperative" multitasking that the Macintosh and *The Manager* do. Without the powerful process-to-process communication that Unix has, multitasking is pretty limited.

II ALIVE: Let's talk about your latest achievement.

BAZYAR: That would be cleaning my apartment. Oh, you must mean *discQuest*. That project started shortly after Sequential acquired the RamFAST from the now-defunct CV Technologies. Joe Yandrofski, the owner of Sequential, was interested in a way to market the card in a flashier "bundled" way to boost sales, and figured that CD-ROM was the way to go.

He gave me a copy of *The Family Doctor* CD and I started digging into it to see how the information was stored on the disc, and to see if I could present it in a way that would be useful to IIGS users. Our concern was that every CD-ROM used different encoding techniques, and that each would require a whole new program to use it, but then we discovered that all the discs in CMC's product line use the same databases for storing and retrieving information. So, when I finished *discQuest,* IIGS users had twenty new CD-ROM titles for the Apple IIGS—not just one.

II ALIVE: What exactly does *discQuest* do?

BAZYAR: A lot! It was very challenging to write. I had to do things I'd never done before: graphics processing, database retrieval, all sorts of stuff. The CMC CD-ROM titles (and some from other publishers) come with an application called *discPassage* that takes a standardized multimedia database and presents the information to the user in a way that allows the user to search hundreds of megabytes of data for words and phrases in just a few seconds. It also has a basic hypertext facility which allows pictures and other related information to be viewed with a click of the mouse. *discQuest* is, basically, a version of *discPassage* for the IIGS. CMC has Macintosh and PC versions of the software. The IIGS version is far superior to the PC version (which uses a plain text interface and gruesome keyboard commands), and stands up well next to the Mac version.

My favorite part of the project was the hacking. CMC wasn't about to tell us how their database worked, so I had to reverse-engineer just about everything. Some of the tricks I'd learned years ago cracking copy-protected Apple II software came in handy.

Once the IIGS market had had a taste of the great new possibility of CD-ROM, they wanted even more. Specifically, an encyclopedia. And that's what I'm working on now—a IIGS front end for the Compton's Interactive Encyclopedia CD-ROM. It should ship sometime in June, and at that point no serious Apple IIGS user will have a good excuse not to have a CD-ROM drive.

II ALIVE: Anything you'd like to say to wrap this up?

BAZYAR: Just one—Apple II Forever! ■

MasterWorks

Continued from page 27

TimeOut SuperFonts), are listed as "Other" in File Activities lists.

Pressing OA-Return on a file list invokes File Activities' "Expert Mode," allowing you to change to another disk or path, and copy, move, rename, delete, lock, and unlock files—without being asked to insert source and destination disks. OA-Return also bypasses the question "Do you really want to do this?"—so be sure that you mean it when you hit OA-Return.

AppleWorks 4's File Copy function allows copying and moving files between two disks of the same volume name. Some other 8-bit utility programs get confused if the source and destination disks have the same name, as they would if you made a backup copy of a disk. Since source and destination disks may be selected by device (rather than name) in AppleWorks 4, disk names pose no problem. Moving files, which is nothing more than copying, then deleting, the source files, is not restricted to movement between subdirectories on the same disk; you can move files to another disk as well. An undocumented OA-N keystroke (or pressing OA-Return on "No") at the question, "Is this correct?" to confirm the file copy or move switches the source and destination paths—useful for "synchronizing" two disks by copying the latest versions of files to both directories.

The File Copy function may also be your most effective opportunity to use the OA-P Pathnames list, which is configured under Standard Settings. Using Pathnames, you can press OA-1 through OA-8 to select source and destination paths with these easy steps:

OA-Q OA-F	Get to File Activities menu
Down Rtn	Select the "Copy Files" function
OA-1	Copy files from source: Pathname 1
OA-2	To destination: Pathname 2
Y	Say Yes to confirm the pathnames
Select files	Choose the files to copy
OA-Rtn	Copy them - with no questions asked.

CONFIGURING PATHNAMES

In order to use the OA-P pathname list, and the hot-keys OA-1 to OA-8 to quickly items from this list, you must first set up your pathnames. Go to the Standard Settings menu (use OA-Q OA-S to get there fast). Select menu item 8, Pathnames. You will be prompted to enter the eight pathnames you use most. After they are configured, you can pull up this list any time you need to change to a different path:

On pressing Escape or OA-Q to exit File Activities, the AppleWorks disk or path is always reset to whatever it was before you entered. This is because your file copy destination, or the directory you just worked on, probably had little to do with your regular AppleWorks data path. If you do not want the pathname reset to what it was, but instead want to work with the last path you were working with in the File Activities, you can exit by pressing OA-Escape Escape to exit File Activities. This will put you momentarily in the TimeOut menu, then at the Main Menu. This unusual (and undocumented) exit method may not have been intended as a "feature"—actually, the fact that the path isn't reset when you leave via OA-Escape is probably a bug—but it's an easy way to handle what might otherwise be an occasional inconvenience.

```
Disk: Ext. Disk 3.5 #1        PATHNAMES      Escape: Standard Settings
________________________________________________________________________
    __________________________
   |     Main Menu            |_____________________________
   |   ________________________                             |
   |  |     Other Activities     |__________________________|_
   |  |   ________________________                            |
   |  |  |     Standard Settings    |_________________________|_
   |  |  |   ________________________                           |
   |  |  |  |     Pathnames            |________________________|_
   |  |  |  |                                                      |
   |  |  |  |     1.  /ram/                                        |
   |  |  |  |     2.  /ram/timeout/                                |
   |  |  |  |     3.  /ram/temp/                                   |
   |  |  |  |     4.  /dl2/                                        |
   |  |  |  |     5.  /sf/fonts/                                   |
   |  |  |  |     6.  /TEXAS.II/                                   |
   |__|  |  |     7.  /TEXAS II/On.MACROS/                         |
      |  |  |     8.  /AppleWorks/                                 |
      |__|  |                                                      |
         |__|______________________________________________________|
________________________________________________________________________
Type number, or use arrows, then press Return               125K Avail.
```

MORE KEYBOARD SHORTCUTS

Back at the OA-Q menu (Desktop Index), OA-Q OA-C views and edits the last used clipboard. Clipboard editing is quite versatile, allowing you do almost everything you can do with an AppleWorks file, including deleting text, inserting printer options, zooming in and out, and—amazingly—even copying to yet another clipboard. (The secret to how the clipboard works while you're editing it: when you edit the clipboard, a dummy file is created, freeing up the real clipboard for use while you are editing the file. When you exit edit mode, all the data is moved from the dummy file into the real clipboard.)

You can't save the clipboard to a file, but you can print it, and, of course, copy or move it into a regular desktop file. To copy or move data from a clipboard other than the one related to the current file, press OA-F (or OA-Rtn on the horizontal menu selection, "From clipboard"). This will present a menu of clipboards from which you can select:

```
Clipboard to use?
Word Processor
Data Base
Spreadsheet
```

Copying or moving to the clipboard always goes to the current application's clipboard. You can mix all three types of data (word processor, data base and spreadsheet) by printing to the clipboard, which always goes to the word processor clipboard—there's an "append" option there to allow more than one file's data to be stored at once.

Appending to the clipboard may be one of AppleWorks 4's best new features; once you try it, you'll find that you use it again and again. The option to append to the clipboard from a file can be found at the far right of the Copy or Move menu. As always, pressing the first letter of a horizontal menu item selects it, so next time you copy to the clipboard, try OA-C, then A to append.

```
Copy Text?
Within document
To clipboard
From clipboard
Append to clipboard
```

AppleWorks 4 contains many other places where pressing OA-Rtn, OA-?, or another Open-Apple key yields an alternative to "regular" keypresses. As you move through the program checking out all the new Open-Apple commands, be sure to stop at the Main Menu and press OA-? for Help. Here, you'll find the AppleWorks 4.0 "Easter Egg," a little-known feature revealed when you scroll to the bottom of the Help screen and press OA-A.

After printing a file, AppleWorks no longer marks it as "Changed," as it did under previous versions. Even quitting AppleWorks 4 is faster than before. Usually, you are asked to save or throw out changed files on the first, second, and third desktops before you quit. To quit without saving any files, press OA-Y (or OA-Return on "Yes") in answer to the question, "Do you really want to do this?" You will exit AppleWorks immediately with no further questions asked.

While AppleWorks includes many shortcuts for moving through its menus and files, there is no substitute for practical experience at the computer. Although we're doing our best to explain AppleWorks 4, you can't fully appreciate the program's finer points by merely reading about them in a book or an article. Now is the time for you to go to the computer, and explore AppleWorks 4 for yourself! ■

Resources:

AppleWorks 4
List: $169.95
Updates: $79.95-$119.95
Quality Computers
20200 Nine Mile Rd.
St. Clair Shores, MI 48080
1-800-777-3642

TEXAS II, TEXAS II on Disk
6 issues, $15.00
6 issues + 3 disks, $39.00
Kingwood Micro Software
2018 Oak Dew
San Antonio, TX 78232-5471
210-490-6373

ProSel-16
List: $89.95
(Discounts available)
Charlie's AppleSeeds
9081 Hadley Pl.
San Diego, CA 92212
619-566-1297

DeskTools IV
List $29.95
Updates available
Quality Computers
20200 Nine Mile Rd.
St. Clair Shores, MI 48080
1-800-777-3642

THE APPLE II AND AUTISM

How Computers Are Helping Special-Needs Children

by Phil Shapiro

Since the early days of the Apple II, the computer has been meeting the special needs of mentally and physically handicapped people—especially children. Using such assistive add-ons as the Adaptive Firmware (AFC) Card and the Echo II speech synthesizer, special needs children have been benefitting from intellectual enrichment opportunities unavailable through any other means.

While the Apple II can serve a broad range of special needs, I happen to have a particular interest in helping children with autism. Autism is an enigmatic neurological disorder that affects the functioning of the brain in diverse ways, mainly involving communications and social-skills deficits. Autism manifests itself in several well-defined behavior patterns: compulsions to perform certain tasks over and over again, difficulty in expressing needs, avoidance of eye contact, apparent insensitivity to pain, lack of understanding of impending dangers, inappropriate or no response to loud sounds, extreme distress for no discernible reason, and an aversion to being cuddled or hugged. An apparent fascination with spinning objects is also common.

The general public's awareness of autism is based on Dustin Hoffman's portrayal of an autistic in Rain Man. But there is a whole gradient of autistic behaviors. The "autistic savant" portrayed in Rain Man is not typical of all autistic individuals.

Milder forms of autism can respond to early intervention. When an autistic child is diagnosed properly, parents and teachers can plan appropriate strategies to help compensate for the child's particular deficits. Determining the exact nature of those deficits is where the Apple II comes into play.

Last week I had the chance to visit some developmental psychologists doing interesting work with the Apple IIGS at the Children's National Medical Center in Washington DC. Dr. Donna Vaught and Dr. Deborah (Debbie) Custer work with young children diagnosed with "pervasive developmental disorder," a form of autism. They use the Apple IIGS to help draw these children out, to help diagnose the child's cognitive capabilities and deficits, and to help show parents how computer-assisted instruction might be of help with their special needs child.

The children that visit the hospital range in age from two to seven years old. Dr. Vaught explains that many of these children have physical coordination problems, but otherwise are cognitively quite capable. These children often show enthusiasm for computers for the simple reason that interacting with computers does not call for the complex social skills the rest of us take for granted.

The computer also helps the developmental psychologists better ascertain what mental tasks the child can and cannot do. The computer assessment of cognitive abilities serves as a useful adjunct to the "pen and paper" cognitive tests that are given to these children. Debbie Custer mentions that she'd love to see some of these "pen and paper" tests transported over onto the computer. With its motivating effect, the computer can help children concentrate longer on the tests. And the longer that the children spend performing the tests, the more these tests reveal the child's particular strengths and weaknesses.

The software used to diagnose children's abilities starts out with the very simplest of computer tasks. I had a chance to try out some "switch" software, where children press a single large red switch when a particular animal passed in front of a hand on the screen. This simplified form of arcade game can tell the psychologists a lot about children's hand-eye coordination as well as whether the child has a good sense of cause and effect. Dr. Vaught mentioned that one autistic child who was previously rather non-verbal showed a greater inclination to communicate when prompted to describe the animals he saw on the computer's screen.

Debbie Custer went on to explain how the children that visit their office have varying developmental abilities. The task of the psychologists is to diagnose as best as possible the location a particular child occupies in the spectrum, and then to prescribe the most appropriate course of action to meet that child's needs.

I was surprised to learn that much of the software being used with these children is public domain, freeware, or shareware. Apparently some really useful software for special needs purposes has been generously distributed in this fashion. (Parents or teachers interested in acquiring copies of public domain or shareware Apple II special needs software can contact Joan Tanenhaus at Technology for Language and Learning. The address and phone number of this organization are given at the end of this article.)

I had a chance to try my reflexes with some of the public domain and shareware software used at the hospital. Persons who use Apple II's with special needs populations might recognize the names of the programs I examined: *Functional Academics, Step by Step, The Scanning Game, Single Switch Games, Dot to Dot, Clown, Dancing Man, Fireworks, Number Switches: Animals, Number Switches: Shapes.* None of these programs are particularly fancy. But most do a good job of establishing cause-effect relationships between the child and the computer.

Donna Vaught also mentioned that she had heard that one of the best software publishers of special needs software is Laureate Learning Systems, in Winsooki, Vermont. I, too, have heard the acclaim and accolades this company has earned. (See sidebar article for further information about Laureate.)

I can't wait to go back and visit the hospital to see how these special needs kids are using the computer. It's a moment of special magic watching how an Apple II computer can bring such joy and excitement into their lives. ■

Autism Resources

Author: The author takes a keen interest in the use of Apple II computers with special needs populations. He can be reached at: Balloons Software, 5201 Chevy Chase Parkway, NW, Washington DC 20015. Or via e-mail on GEnie at: P.SHAPIRO1; on America Online at: pshapiro; on Internet at: pshapiro@aol.com. Electronic mail communication preferred.

Contacts

Dr. Donna Vaught
Dr. Deborah Custer
Children's National Medical Center
111 Michigan Ave., N.W.
Washington DC 20010-2970
(202) 884-2187

Publications

Two of the best publications that cover special needs computing software and hardware are the *ConnSENSE Bulletin* and *Closing the Gap*. The *ConnSENSE Bulletin* is a highly respected publication assembled by Chauncey Rucker, at the University of Connecticut. Both *ConnSENSE* and *Closing the Gap* sponsor annual conferences on special needs computing. The *ConnSENSE Bulletin* is more informal, more along the lines of a well-organized user group publication. They tend to ferret out more info on products from smaller developers. *Closing the Gap* articles tend to be longer. Both publications have something important to offer.

ConnSENSE Bulletin (3 issues per year)
US Individual ($15), Schools & Institutions ($26), Individuals outside US ($20).
Multi-year subscription discounts available, too.

UCONN
Chauncy N. Rucker, Ph.D.
A. J. Pappanikou Technology Lab
249 Glenbrook Rd, U-64
Storrs, CT 06269-2064
(203) 486-0172
Internet:RUCKER%UCONNVM.
bitnet@yalevm.ycc.yale.edu
Make checks payable to: A.J. Pappanikou Center

Closing the Gap
Annual subscription: $26
Two year subscription: $42

Closing The Gap
PO Box 68
Henderson, MN 56044
(612) 248-3294
America Online: ctg
Internet: ctg@aol.com
Make checks payable to: Closing The Gap

Apple II Special Needs Computing

Joan Tanenhaus
Technology for Language and Learning, Inc.
P.O. Box 327
East Rockaway, NY 11518
(516) 625-4550

This non-profit organization was started in 1988 to help advance the use of technology and computers with children and adults with special needs. The organization's principal work has been to collect and create high quality public domain and shareware Apple II software for special needs users. A 46-page catalog listing the collected programs is available for $10.

Joan Tanenhaus, the founder and principal operator of this organization, is nationally recognized for her expertise in assistive computing. She writes regularly for Closing the Gap and other national publications, and was asked to be one of the judges of the 1991 Johns Hopkins National Search for Assistive Technology contest. Most of her time is spent giving workshops and doing consulting work with schools and organizations in the New York City metropolitan area.

Street Electronics Corporation
6420 Via Real
Carpinteria, CA 93013
(805) 684-4593

Manufacturer of Echo and Cricket speech synthesizers. Echo works with the Apple II+, IIe, and IIGS. Cricket works with the Apple IIc and IIc+. The TextTalker program that makes these speech synthesizers work is written in Applesoft BASIC. It's possible for persons with a knowledge of Applesoft BASIC to create their own talking software, or modify existing public domain programs to add speech synthesis features.

Don Johnston Development Co.
1000 N. Rand Rd., Bldg. 15
Wauconda, IL 60084
(800) 999-4660
Contact: Barbara Sistak
America Online: DJDE
CompuServe: 72350,3143
Internet: DJDE@aol.com

Manufacturer of Adaptive Firmware Card. As of January, 1994, the price of the Adaptive Firmware Card is $520, plus $15 shipping. It can be used in either the Apple IIe or IIGS.

Edmark
P.O. Box 3218
Redmond, WA 98073-3218
(800) 426-0856

Manufacturer of TouchWindow touch screen. Apple II TouchWindow list price is $275. Available at discounted street price from Quality Computers and Educational Resources.

Organizations

The Disabled Children's Computer Group is a community-based resource center that allows parents and teachers to examine software and hardware for special needs users. Nationally recognized, and well worth a visit if you're in the San Francisco area.

Disabled Children's Computer Group (DCCG)
2547 8th St.
Berkeley, CA 94710
(510) 841-DCCG
Internet: DCCG@applelink.apple.com

The Trace Center distributes an inexpensive CD-ROM disc, titled "Co-Net CD-ROM" that includes several databases with information about 18,000 assistive and rehabilitation devices. This CD-ROM is available for Macintosh or IBM computers, and sells for $27 (including postage).

Trace Center
University of Wisconsin
S-151 Waisman Center
1500 Highland Ave.
Madison, WI 53705
608/262-6966
608/262-8848 FAX
608/263-5408 TDD
Internet: info@trace.waisman.wisc.edu

The Autism Society of America promotes information and public awareness on autism, and currently operates through a network of 186 state and local chapters working with parents, professionals, and the general public to educate and inform about autism. The Society publishes a quarterly newsletter, The Advocate; distributes free information packages; and offers referral information.

The information packet the Autism Society of America sent was quite impressive. This organization should be the first stop for anyone wanting to learn more about autism. Apparently the organization conducts most of its information referral work by phone. Annual dues are a reasonable $20 for individuals, and $30 for families.

Autism Society of America National Headquarters
7910 Woodmont Ave, Suite 650
Bethesda, MD 20814
(301) 565-0433
(800) 3-AUTISM
(301) 657-0869 (fax)

Books

Grandin, Temple, *Emergence: Labeled Autistic*
(A moving autobiography by a college professor who tells about her life and experiences with autism. See also New Yorker article by Oliver Sacks, listed below.)

Barron, Judy and Barron, Sean, *There's a Boy in Here,* New York : Simon & Schuster, 1992. (A riveting first-person account of a mother's attempt to reach her autistic son.)

Hart, Charles, A.. *A Parent's Guide to Autism,* New York, NY : Pocket Books, 1993. (A book with useful practical advice.)

Videos

Autism: Learning to Live. Presented by the Institute for the Study of Developmental
Disabilities, Indiana Resource Center for Autism ; produced by WTIU/Indiana University Radio and Television Center, Bloomington, IN : Indiana Resource Center for Autism, 1991. 1 videocassette (58 min., 46 sec.).

Autism—A World Apart. University of Southern California School of Broadcasting-Journalism; Boston, MA: Fanlight Productions, [1988]. 1 videocassette (30 min.)

Laureate Learning Systems

Laureate Learning Systems was founded in 1982 by two speech-language therapists interested in how computers can assist children and adults with special language needs. The company's software spans across a linguistic hierarchy, ranging from toddler age on up. Their software can also be helpful to adults suffering from traumatic head injury and aphasia.

Laureate's software has won all sorts of awards and recognitions in the assistive computing field. I was impressed to note that a list of their awards takes up half a page in their company profile flyer.

Currently they have 16 Apple IIe software titles and 10 Apple IIGS titles. The IIGS programs run under System 5.0, and usually require 512K or 1MB of RAM memory. Hard drive users will be pleased to note that this GS software is hard drive installable, although you'll need the original disk handy for the program to operate.

Many of the company's "Apple IIe" software can be run on the older Apple II+ computer as well. Bernie Fox, the company's vice-president, explains that the catalog refers to the software as "Apple IIe" software to differentiate it from Apple IIGS software. Almost all the "Apple IIe" software titles will run on any Apple II+, IIe, IIc, or IIc+, as well as the IIGS.

A generous preview policy allows parents and teachers to try out the software for 30 days before purchasing it. (Just call with a credit card number or purchase order.) Prices range from $95 to $225 per disk—which seems high for a traditional publisher, but is quite fair for software with such a small market.

Steep discounts are offered for schools or learning centers wishing to purchase several copies of a program. You can buy a lab pack of five disks for double the price of a single disk, or you can buy ten disks for three times the cost of a single program.

Also, Laureate has an innovative offer which allows customers to trade in programs they have outgrown and receive a credit of 50 percent of the original purchase price toward their next purchase.

Finally, the company recently set up an electronic mail address to help communicate with existing and prospective customers. You can reach Laureate on CompuServe at: 74040,73, or via Internet at: 74040.73@compuserve.com

While visiting the Children's National Medical Center in Washington D.C., I had a chance to see two Laureate programs in action. "First Words" trains and tests early development nouns. Words are presented in various categories, including animals, household items, body parts, outside things, clothing, toys, common objects, utensils, food items, and vehicles. I couldn't help but smile at the cute animation and sounds in this program. Correct answers are rewarded with zany humorous antics on the screen.

Donna Vaught mentioned how show liked using this program with Edmark's TouchWindow. She went on to explain how the software gives some important customizing options: you can choose to have the Touchwindow accept only responses that are right on the location of the word/graphic being tested, or you can choose to have the TouchWindow accept responses that are in the general vicinity of the word/graphic being tested. By allowing such choice, the software can still be beneficially used with children who have motor coordination deficits.

Another program I examined was Micro-Labs. As with Laureate's other software, this program gives many choices to customize the activities for a particular student or hardware set-up. For example, higher functioning children might opt to use a mouse rather than the Touchscreen input device.

A free catalog of Laureate Learning Systems products is available upon request. The company also has an 8-minute videotape, which is also free upon request.

Laureate Learning Systems
110 East Spring St.
Winooski, VT 05404-1837
1-800-562-6801

Growing an Apple Orchard

by David Lewis

What can a teacher do to get more computers in your school? I'm a computer teacher and administrative assistant, though when I started my quest I was a seventh grade teacher of many subjects. This is my tale.

PLANTING THE SEEDS

Our school is a small (150 students) re-kindergarten to grade 11 school in the isolated north of Quebec, Canada. The students are natives and speak English as their second language.

I got my Apple IIGS in 1987. I'd looked at IBM clones, but couldn't see myself dealing with the MS-DOS prompt. I also looked at the Mac (which the dealer was pushing), but when I saw there was no color monitor or printer for the machine, I chose the IIGS. Any teacher will tell you how quickly they get tired of coloring signs, posters, banners, etc. by hand. When the dealer mentioned a program that would keep my marks, I was hooked!

For the next couple of years, I used the GS for my classes. My students appreciated my efforts—often, my GS gave a little extra something to my teaching. Other teachers took note, too, and I often heard, "Boy, I wish those lab computers could print in color!" and "Your marks are done already?"

Meanwhile, the school board discovered computers. (A mere ten years after the microcomputer was introduced.) They were seen as "the wave of the future" and it was decided that they would be available to all students. First it was announced that every student would have his or her own computer. When the cost was finally counted, that proclamation was revised to "a computer for every classroom." When that cost was calculated, there were murmurs of a computer lab in each school, maybe.

Which computers would be best? What it finally came down to was not which system would be the easiest for students to learn to use, or which would enhance teaching the most, but which was the cheapest. Because in most school boards, educators don't make the decisions. It's the bean counters. Those in charge of the purse strings often have the final say. It's not what's best for the students and teachers in the long run. It's the best deal, the lowest bid. And that translated into IBM clones.

So eventually our school received a small lab of eight IBM clones. As time went on, they were upgraded and, later, replaced with more up-to-date clones. The computer teacher taught BASIC programming, along with spreadsheets, databases, and of course, *WordPerfect.*

BEARING FRUIT

When the computer teacher left, I was offered the position. After much thought, I decided I could do more to help the students than my predecessor, so I accepted.

I believed then, and still do, that application is what counts. In other words, how you use a computer to make your life easier, more productive—and yes, more fun—is more important than what computer you use. If we want our students to graduate ready for work, they must have computer knowledge and application skills of a general, not specific, type. With computer systems and software changing so rapidly, it does no service to train students to use a particular software program or operating system. It's far better to give students the background they need to figure out things on their own.

So I worked with the students in many diverse areas. Everything from keyboarding to graphics to desktop publishing. Most of all, I encouraged exploring. I wanted to see students think and apply themselves to all sorts of computer situations.

There's nothing worse than a school where students go to the lab once a week—or, worse, once or twice a month. What can the computer teacher (assuming the school even has one) expect the students to retain from week to week? For best reinforcement, students should be on the computers every day. And students should never be taken to the computer lab so they can play. That's a waste of time, and only reinforces the idea of a computers as game machines.

As I started my new job, I realized that I wanted to see at least one computer in each grade level for use by both teachers and students. I wanted to see teachers using that computer as both a presentation and an administrative tool—and I also wanted to see students working individually cooperatively, using the computer for teacher-assigned projects.

When you mention computers in the classroom, many teachers think of a student sitting glassy-eyed before a monitor, playing war games. It does happen sometimes—but it doesn't have to! Teachers can use a computer with a large screen monitor or television to involve the whole class in discussions, problem-solving activities, and group decisions. And not only did I want a computer for every classroom, I also wanted to see the existing computer lab used in a way that would complement and reinforce the use of the computers in the classrooms.

EXPANDING THE ORCHARD

The first time I inquired about getting more computers in the school, I was told gently but firmly that, based on the number of students we had, we already had our quota of computers—and we were lucky to have as many as we did! No more computers would be forthcoming. I could have folded my tent and vanished in the dark, but I couldn't leave my students. I kept thinking about options, and so I made a plan of my own.

Picking the type of computer was no problem—since I already knew how easy the Apple IIGS is to operate, that was my obvious choice. Other factors involved included the low cost of second-hand IIGS computers (inexpensive, with the seller usually tossing in lots of software) and maintenance (I never had a problem with my GS, while there was not one day when all our clones were in working order). I figured we'd buy second-hand systems from money raised from various means. But before I could even start thinking of a way to raise the necessary pennies, opportunity came knocking.

I was walking through the office when I overheard a teacher talking to someone on the phone. The words were something like, "I don't know where you could get rid of those computers." With a single leap I was over the desk and managed to get the teacher's attention. As soon as he hung up, I grilled him on what he'd been talking about.

At a nearby school (the only other school in our small town, actually), the principal had a unique problem. Once upon a time, they had had a computer teacher who ordered computers for the school. The computers came, but they'd barely been taken out of the boxes when the teacher quit. No one else knew how to operate them, and nobody was interested enough to learn. So for two years they sat at the back of their "computer lab," unwanted and unloved.

The school had just received a grant to—get this—buy computers for the school. They'd decided to get the most powerful multimedia PC clones they could afford. Their problem was what to do with those dusty, pesty, out-of-date Apple IIGS computers sitting at the back of their lab. I phoned that school's principal the next day, and semi-calmly told I might be interested in those out-of-date computers. He invited me to drop by after school and take a look.

When I walked into their lab, I wanted to dance the dance of joy, but I restrained myself. I looked over the computers, ran my fingers over the dust, and told him I might be interested in buying all six of them if the price was right. His only question was, "How much?"

"Wellllll, they are outdated, but I suppose we could use them in the classrooms. If you throw in the printer, how about $3500.00?"

"Sold!"

As you can imagine, I left happy that night. Of course, there was a minor problem. Somehow, I had to come up with the money. But I was wired with success, and I certainly wasn't going to let a little thing like money stop me from taking advantage of a great deal like this.

MAKING IT HAPPEN

I went to our principal and described the deal and my plan for the school. I also told him that if he didn't go along with the loan, I would have no choice but to buy them myself (via a bank loan), and re-sell them somewhere else. I was certain I could make a bundle in short order on such a deal. The principal came around to my way of thinking and lent me the money—I'd have to pay it back before school ended that school year.

Ahhh, what a feeling of accomplishment! I picked up the computers and installed them at our school. The students were thrilled, but many of the teachers were apprehensive. And I had incurred a huge debt, and still had no software for the computers. So it began.

I acquired a partner—the first grade teacher—who believed in the plan, was enthused about the computers, and wanted to get software right away. You name it, we did it. We had bingos, caribou spaghetti suppers, raffles, draws, rummage sales, and other ventures too numerous to mention. But it paid off. We paid back the money in plenty of time and had enough money left over to start our software library.

The best way to become aware of the best software is to read everything you can get your hands on. I subscribed to numerous computer magazines and joined a user group. Usually, I looked for software that was open-ended enough to be used for many different subjects (AppleWorks, *The Childrens' Writing and Publishing Center,* and *HyperStudio* are perfect examples). However, when a teacher expressed an interest in software for their particular subject, I was happy to oblige.

Besides actually acquiring and installing the computers and software, I also had to help teachers overcome their fear of the technology. Most teachers are way too busy to sit down and really learn a new program—or even to read manuals. The fastest way to learn is still by doing, with someone there to help you if you get stuck. I took on this role, and subscribed to even more magazines, joined more user groups, and even got a modem and accounts on a few online services so I could continue to serve as the expert.

From this auspicious start, our Apple orchard has continued to grow. Each year we've added more computers and bought more software. We've bought second-hand computers from all over: California, Ontario, British Columbia. If it's a good deal, we go for it—and there's now an Apple IIGS in every single grade.

Where will it end? It won't. There are always new programs to buy. There's hardware to buy: accelerators, memory boards, hard drives, scanners, laserdisk players, and so forth. And yes, we still want more computers! One per classroom is great, but two or more in each classroom would be heaven!

RECIPE FOR SUCCESS

So that's the story. What's the "boiled down" recipe for getting computers—or more computers—into your school? Here, in summary, is what worked for us.

- There must to be one person in charge—someone with fire in their eye. He or she must have the drive, determination, faith—and the time—to work not only on the plan, but also on involving the teachers and students. There are talkers and doers. Don't whine and complain about how bad things are and how you wish somebody would fix things. Do it!

- There must be a plan. You wouldn't think of going to a strange city without a map. So it is with computers. You will also get more people (like the principal) on your side when you can describe exactly where you plan to take the school.

- You have to have a partner (at least one). No matter how dedicated and driven you are, it will be a tough road to travel by yourself. You need someone to bounce ideas off of and to take over when you're unavailable.

- While it's good to have the school authorities and the teachers behind you, the most important crowd to have in your corner are the students. Do demonstrations, help teachers use the computer in their subject, and yes, have some "fun" (but educational) programs for the students too. Prove that learning can be fun with computers, and the kids' enthusiasm will win over even the most technophobic teacher.

- Look for ways to raise money. Brainstorm with fellow teachers, students, and friends. Explore all the possibilities. Last year we received a small grant for a multimedia project, to be produced by the students on the history of their tribe. Not only did we receive funding for this project, but the "official" stamp of approval gave students extra enthusiasm and motivation—and pride. Read all you can in as many different places as you can. Subscribe to catalogs and look for the best deals. Take out a classified ad out if you have to! There are people out there who would donate their systems to you, if you only ask.

In the end, it comes down to you. If you want change the world, start by doing something. Make your mark on your school, your students, and their lives. Show your students that one person can make a difference. Don't wait for opportunity—make opportunity!

By the way—if you have an Apple IIGS system, software, or hardware want to sell, or even donate (it's tax deductible!), please contact:

David Lewis
P.O. Box 520
Schefferville, Quebec
G0G 2T0 CANADA

(418) 585-3811 day phone
(418) 585-3433 evening phone
(418) 585-3347 Fax

Yes, even writing articles for magazines and newsletters is a valid way to spread the word about your plan. ■

Standing on the Shoulders of Giants

by Michael L. Rogers

STATEMENT OF THE PROBLEM

I teach an introductory chemistry course at the freshman college level which is taken by a variety of nonscience majors. My course is the only opportunity many of these students have to put into practice the scientific method as a way of discovering truth about the substances and processes that make up our world. Because most of my students go on to careers which require a high degree of human interaction (e.g., nursing, health management, and sports medicine), I have implemented collaborative learning activities in all parts of the course where the scientific method is employed.

As I searched for appropriate lab experiments, texts, and supplemental materials, I soon came to realize that textbooks and teachers alike usually present the scientific method in ways that emphasize the resulting scientific laws and theories, and not the process by which those laws and theories are generated. Moreover, laws and theories are all too frequently presented as the finished product of an isolated application of the scientific method. Such an approach is at cross purposes to group learning, so I began to develop an approach to the scientific method which would emphasize the collaborative nature of science.

Isaac Newton was my inspiration. When Newton was asked how he had come to achieve so much as a scientist, he answered, "If I have seen further, it is by standing on the shoulders of giants." Newton clearly understood his relationship to the scientists and experimenters who preceded him, and I saw in his words a way to structure my course to model the actual process of scientific discovery—a process in which experimental data comes at the beginning, not the end, and in which both data and conclusions are communicated to other scientists across both time and distance. The data and conclusions of one scientist become the starting point for another scientist, making both scientists partners in a common enterprise: the pursuit of scientific knowledge. Collaboration comes in the communication; in my course, science is as much about communication as it is about natural phenomena, laws, and theories.

To make my students into scientific collaborators, I assigned some of the historically significant experiments which led to the chemical principles covered in our course. The experiments included:

- Volta's Electric Pile
- Guericke's Sulfur Sphere Electric Generator
- Toricelli's Vacuum and Pressure Experiments
- The Magdeburg Spheres
- Priestley's Experiments on Gases
- The Hallwachs Photoelectric Effect
- Joule's Mechanical Equivalent of Heat

After researching the assigned experiment, each student group developed a modification of the experiment that could be carried out in the lab. Within the limitations of safety and available equipment, the experiments were to be as faithful to the historical precedent as possible.

ENTER THE COMPUTER

Since my goal was to get my students to actually develop the important theories and laws on which chemistry is based, the groups were required to present their actual data in a report. I taught the groups to use Roger Wagner's *HyperStudio* on our Apple IIGS computers to produce and present their reports, with ThunderScan and ComputerEyes digitizers providing images from reference texts and still pictures of the experiments.

We have *HyperStudio* disks available in our computer lab. However, I found it helpful to set up a "Production Room," available to students on a sign-up basis, in which an Apple IIGS is permanently connected to an Imagewriter printer (for Thunderscan) and an old U-Matic VCR (which produces nice still images for ComputerEyes). Of course, the production room Apple IIGS is also fitted with a *HyperStudio* microphone. I also connected a surplus composite color monitor to allow the VCR's image to be viewed while ComputerEyes completes a scan.

Since my course is already quite demanding, even the *HyperStudio* tutorial book, as good as it is, represents an intimidating additional workload to students. I therefore developed a half hour training session in the use of *HyperStudio*, then completed training group-by-group, on a need-to-know basis. All groups quickly learned to construct stacks that included clip art, scanned images, sound bites, and text. A few of the groups used *HyperStudio's* SimpleScript programming language, mainly to do calculations (such as Boyle's Law problems) with viewer-supplied input data.

STRUCTURE IN SCIENTIFIC COMMUNICATION

Elsewhere in my course, students discover why written scientific reports follow a definite format: Title, Abstract, Introduction, Procedure, Results & Discussion, Conclusion. A consistent structure helps the reader to focus on the information, rather than the writing style. I also developed a standard format for *HyperStudio* presentations, so as to focus the viewer on the main themes of the experiment, which are defined by the following questions:

1. How was the experiment set up?
2. What previous work made this experiment possible (and why)?
3. What results were obtained?
4. How did the experiment change society's thinking about our world?

Students generally set up the stack around a description of the experimental design, then used buttons placed on the appropriate cards to explore the other three questions. To maintain uniformity, I asked all students to use the same names for buttons serving the same purposes:

buttons addressing question 2) were named "History"; buttons for 3) were named "Results", and buttons for 4) were named "Changes". For a stack to be complete, each kind of button had to be used.

PRACTICALITIES

Until very recently, the Apple IIGS, running software such as *HyperStudio*, was the most economical way to do multimedia in the classroom. Although recent reductions in hardware costs and new capabilities in software have changed the multimedia landscape dramatically, *HyperStudio* on the Apple IIGS remains a great way to get into multimedia. Students learn to use the Apple IIGS very quickly, and *HyperStudio* has just the right mix of simplicity and capability for classroom use. I expect to use our multimedia hardware and software for at least four more years.

Our computer stations do not have hard drives; consequently, each team preparing a stack is required to fit the entire stack, including the run-time version of *HyperStudio*, on a single 800K 3.5 inch disk. This length restriction actually fits nicely into my instructional strategy: conciseness is an asset in multimedia just as it is in written work. Students soon learn how to maximize the informational content of graphics and audio clips, and I frequently find that students are weighing very carefully the relative importance of the various media and pieces of information available to them.

One of the more useful multimedia capabilities of the Apple IIGS is the ability to videotape a stack directly to a VCR, through the composite video out jack. Although composite resolution and color on the Apple IIGS leaves something to be desired, judicious selection of text font size and color can do much to improve the view, and students seem not to mind the color artifacts anyway. All stacks are videotaped and run on a continuous loop VCR next to the constructed experiment in the lab during our Lab Fair. Stacks are then made available to students on a checkout basis as they prepare for the final exam, which covers the chemical principles revealed by the lab experiments.

Because the buttons in all the stacks are standardized, I can combine several stacks to show the progression of an idea, such as the idea of electrical charge, through its complete historical development. I've also been able to edit student-produced stacks for use in lectures; the composite video out of the Apple IIGS facilitates this mode of presentation in our classroom, which is equipped with a projection TV.

It should be no surprise to *HyperStudio* owners that students react enthusiastically to learning tools which unleash their own creativity and give them a sense of ownership of the content of the stack. The ComputerEyes images, which are always pictures of the students' own work, seem to be especially useful in conferring ownership on the students. My students have a much richer sense of their place in the process of scientific inquiry than they could ever have doing a stand-alone "cookbook" lab experiment in which data was communicated solely by pencil and paper. ■

Making Your Own MOD Files

by Tony Morales

In the first part of this article, we looked at what MODs were and how to put together the samples and the header for your own MOD, using only a few simple tools. In this final installment, we'll look at the more complicated process of putting the notes together.

CONSTRUCTING NOTES

Each note's data includes the note value (pitch) to be played, the instrument to be used for the note, and the effects and other parameters for the note, if any. The note value (pitch) requires 12 bits, the instrument number takes 8, the effect takes 4, and the effect parameter fills 8.

The way all this information fits together into four bytes is a little confusing. (See Figure 1.) The first byte of the 4-byte note consists of the upper 4 bits of the instrument number in the high-order nibble, along with the upper four bits of the note value in the low-order nibble. (The high nibble will always be \$0 or \$1, since instrument numbers range from \$00 to \$1F.) The low-order nibble of the instrument number is stored in the high-order nibble of byte 3 of the 4-byte note.

The low-order nibble of byte 1, and all of byte 2, combine together to be the note value. This is actually a numerical representation of the frequency of the note—not the note's position on a piano keyboard or the musical staff. (See Table 1 for allowable note values.) There are 3 octaves possible, with 12 notes in each octave. While a three octave limit may seem restrictive, this restriction is easy to get around—simply create another instrument sample with a higher or lower pitch and use this instrument to play the lower or higher notes.

The low-order nibble of byte 3 contains the effect number, \$0 to \$F. (Effects are used to add musical flourishes to your MOD.) The last byte contains the parameter for the effect. If no effect is desired, both the parameter byte and the effect nibble should be set to 0. A list of the available effects, with their required parameters, is located in table 2.

The following example demonstrates the 4 bytes needed to play a note C in the 3rd octave, with instrument #1, and a fine slide up value of 2:

```
00 D6 1E 12
```

This can be broken down as follows:

- Instrument number: first nibble (digit) of first byte and first nibble of third byte: 01
- Note value: last nibble of first byte and all of second byte: 0D6
- Effect: last nibble of third byte: E.
- Since the effect is E, the last byte is broken down into an effect number (1) and a parameter (2).

BUILDING A PATTERN

A pattern is a 1024-byte chunk of the MOD, which is made up of 64 positions. Each position consists of 16 bytes, which store four notes and their associated "effects." (Individual notes, then, require four bytes of storage.) A MOD has four "tracks," each of which can contain an independent melody or accompaniment part; however, on each track, only one

Figure 1—A Protracker note (four bytes)

```
_____ byte 1 _____    byte 2    \  _____byte 3_____       byte 4
/                    \/          \ /                 \      /        \
0000                 0000-00000000  0000             0000   00000000
```

Upper four bits of sample number.	12 bits for note period.	Lower four bits of sample number.	Effect number	Effect parameter

note can be playing at a time.

Now, how do you put the individual note values together into a pattern? The MOD player steps through the pattern, with a short delay between each division. (The delay is based on the tempo stored with the MOD file.) At each "tick" of the clock, the MOD player examines the current note data in each pattern.

When the MOD player sees a note in a pattern, it begins playing the note. But how does the MOD player determine when to stop playing the note? The note stops playing automatically when the end of the instrument sample is reached, or when another note is encountered on the same track. (Remember, each track can only play one note at a time.) Note values in which the bytes are all zero tell the MOD player not to start a new note, but to continue playing the current note (or silence) while this tick of the clock passes.

If an instrument is looped, it will continue playing until another note is encountered, since the sample end is never reached. To turn off such a note, or to stop any note before its time, you can use the Set Volume effect (effect number C) to start another note with a volume of zero. (To start the next note, you should use the same effect to set the volume back to an audible level.)

Another way to look at it: there is no length parameter—the length of each note is determined by the space between that note and the next one. If you consider each pattern to be a measure of 4/4 time, then a quarter note rhythm (four notes per measure) would be produced by separating the notes you enter by sixteen spaces. More complicated rhythms are possible, although it is difficult to create triplet (and other odd value) rhythms because the number of patterns (64) is not evenly divisible by three.

It will take some experimentation to get the hang of creating patterns. To get started, zero 1024-bytes of memory, like this:

```
00<1000.13FFZ
```

Then begin entering your notes. Remember, each "tick" of the clock takes 16 bytes, so simply add one to the "sixteens" digit to move to the next position within the pattern. When the pattern is done, save it to disk:

```
BSAVE P1,A$1000,L1024
```

MAKING A SONG

Make a couple more patterns, repeating the above steps, each time saving the new pattern with a higher number (P2, P3, etc.). When this is done, reload the MOD header and define the pattern play list. Type the following to reload the header:

```
BLOAD HDR,A$1000
```

Then, you must decide the order in which you would like your patterns to play back. You can play them in any order and repeat them as often as you wish (perfect for verse/chorus arrangements of songs). Count the total number of patterns you want to play. Suppose you wanted to play your patterns in the order 1, 2, 3, 3, 2, 1. This is a total of six patterns to be played, even though you only have three patterns. At offset 950 ($3B6) in the header, store this total:

```
13B6:06
```

Then, starting at offset 952, enter the patterns to be played, subtracting one from each pattern number:

```
13B8:00 01 02 02 01 00
```

You can have up to 127 patterns in the play list, which play back up to 64 actual patterns. Resave the header as follows:

```
BSAVE HDR,A$1000,L1084
```

LINKING THE MOD

The patterns (in order) are stored right after the header. The sample data is stored after the patterns. Combine the files together (using Davex or another similar utility) in the following order:

```
HDR, P1, P2, P3, INST
```

Naturally, if you have more patterns, they should follow after P3 but before the instrument file. Now load your MOD into one of the IIGS MOD players and test the results!

It may take some time to fully understand the concepts presented in this article. If you are serious about creating your own MODs, the best thing to do is to look at other MOD files. Compare the data you see with the sounds you hear to find out exactly what each effect sounds like. Happy MODing! ■

Table 1—Legal note values in Protracker MOD files

Note	Decimal	Hex
C1	856	358
C#1	808	328
D1	762	2FA
D#1	720	2D0
E1	678	2A6
F1	640	280
F#1	604	25C
G1	570	23A
G#1	538	21A
A1	508	1FC
A#1	480	1E0
B1	453	1C5
C2	428	1AC
C#2	404	194
D2	381	17D
D#2	360	168
E2	339	153
F2	320	140
F#2	302	12E
G2	285	11D
G#2	269	10D
A2	254	0FE
A#2	240	0F0
B2	226	0E2
C3	214	0D6
C#3	202	0CA
D3	190	0BE
D#3	180	0B4
E3	170	0AA
F3	160	0A0
F#3	151	097
G3	143	08F
G#3	135	087
A3	127	07F
A#3	120	078
B3	113	071

Table 2—Effects available in Protracker MOD files

Note: Not every MOD player supports every effect.

Effect	Command	Effect	Command
0	None/Arpeggio	8	* NOT USED *
1	Portamento Up	9	SampleOffset
2	Portamento Down	A	VolumeSlide
3	TonePortamento	B	PositionJump
4	Vibrato	C	Set Volume
5	ToneP + VolSlide	D	PatternBreak
6	Vibra + VolSlide	E	* SEE BELOW *
7	Tremolo	F	Set Speed

For the E command, the parameter only uses the lower 4 bits of the parameter byte. The higher 4 bits are used for additional commands.

Effect	Command	Effect	Command
E0	* DON'T USE *	E8	* NOT USED *
E1	Fineslide Up	E9	Retrig Note
E2	Fineslide Down	EA	FineVol Up
E3	Glissando Control	EB	FineVol Down
E4	Vibrato Control	EC	NoteCut
E5	Set Finetune	ED	NoteDelay
E6	Patternloop	EE	PatternDelay
E7	Tremolo Control	EF	Invert Loop

ChronOS

The life and times of the Apple II operating system

Part 1: The Early Days

by Nate Trost

In a stage play depicting the history of computing in the 1990s, the Operating System Wars would undoubtedly be center stage. The operating system (OS) is a vital mediator between the applications that run on a computer and the machine's actual hardware. In many ways, good hardware is useless without a good operating system to make the power of the machine available to application programs.

Operating systems have grown from their humble beginnings to serve many purposes. Most people think an operating system's main purpose is managing files and disks, but modern microcomputer operating systems also control access to other hardware resources (such as serial ports), handle networking, provide a consistent user interface, manage memory allocation, direct multitasking, and provide communication protocols that allow programs running at the same time to communicate with each other. In many cases, the operating system is designed so that application programs do not even need to know details about the hardware they're running on. (Observe, for example, how most Macintosh programs will run properly regardless of how large the monitor is or how many colors it is displaying.)

As mainstream computing moves towards standardized hardware, software companies battle to proclaim their OS as the wave of the future. "Bigger!" and "Better!" are the rallying cries as vendors attempt to win market share in an ever increasing base of computers. In many ways, the question users ask when buying software is changing from "Which computer will this program run on?" to "What operating system does this program run under?"

The stakes are high, and the players are numerous—Apple has System 7, is developing System 8, and owns a large part of a company called Taligent which is busy creating a new OS code-named Pink. Microsoft has MS-DOS, Windows, Windows NT, and is developing Chicago (Windows 4), Cairo, Daytona, and others. IBM has OS/2, and, along with Apple, part of Taligent. NeXT has NeXTStep, Sun has Solaris, Novell has NetWare and, now, Unix (which a lot of other companies have, too).

It seems like every other week there's a new "strategic alliance" that will eventually result in a new operating system choice for users. IBM's PowerPC-based PReP machines will reportedly be offered to the public without any standard operating system—instead, IBM will offer five different operating systems and let the market decide which one will become the standard! The operating systems are likely to be Windows NT, OS/2, Workplace, AIX (IBM's version of Unix), and, eventually, Taligent's Pink.

While none of these new developments are directly relevant to the Apple II user, operating systems have always played an important role in the growth and success of the Apple II. In this article, we'll begin a fascinating trip through the development of Apple II operating systems.

The Beginning (June 1977)

Interestingly enough, when the Apple II was first released in 1977, it had basically no disk operating system—there wasn't a need for one, since the first disk drive for the Apple II wouldn't ship for another year.

Without disks, the only storage medium available was the audio cassette tape. Early Apples had a cassette port on the back of the case which allowed a standard cassette recorder to be connected to the computer. The computer's firmware (the software built into the computer's read-only memory) had software which allowed the user to store and retrieve data on the tape (in the form of the Monitor's R and W commands, and the BASIC LOAD and SAVE commands).

The concept of a file was a fuzzy one. To load a program, you had to insert a cassette and fast-forward or rewind it to the point on the tape where the program had been saved. As you can imagine, the aggravation of finding the right spot—coupled with the reliability problems and slow loading speed of tape—made cassettes a less-than-ideal storage medium.

The Disk Drive (June/July 1978)

In June of 1978, Apple unveiled the Disk II 5.25" disk drive and a corresponding disk controller for the Apple II. The cost of one drive and a card was around $500. In 1978 the Disk II cost hundreds of dollars less than disk drive systems for other microcomputer platforms, when they were available at all—and, furthermore, could store more data than other such systems. The arrival of the disk drive in both physical and financial terms was one of the major reasons for the growth and success of the Apple II platform.

Now the Apple II needed a standard way to load and save files on disks. Apple filled this gap by introducing Apple DOS (Disk Operating System) with the Disk II drive. After a month-long holdup due to some serious bugs in DOS 3.0, Apple started shipping DOS 3.1 in July of 1978.

Compared with the operating systems of today, DOS 3.1 and its progeny look like something out of the Stone Age. As its name suggests, the scope of DOS was limited to disk input and output, unlike later operating systems, which handled user interface and access to other types of peripherals. DOS was even simplistic in its handling of files, using a "flat file" directory system which stores all the files on the disk in one master list. Given the capacity of the Disk II, the number of files that could fit on a disk wasn't so overwhelming to deal simultaneously, so this was a reasonable approach.

In most ways, the simplicity of DOS was an advantage. Since the documentation Apple initially shipped with DOS was less than complete, the main topic of user group meetings and newsletters was, for a while, how to use DOS. (Apple later remedied the documentation deficiency.)

Another advantage of DOS's simplicity was that programmers could learn to use it easily. This facilitated the development of important programs, such as VisiCalc, which helped build the Apple II into the dominant business computer platform.

The nature of DOS also let programmers and hobbyists tear into it, to customize DOS and create utilities that let users customize or improve DOS on their systems. Prime examples include:

- Beagle Bros' DOS Boss, a DOS customization utility that allowed changing command names and error messages, among other things
- Diversified Software Research's DiversiDOS and Beagle Bros' ProntoDOS, modifications of DOS that made disk access substantially faster
- MicroSparc's UniDOS, a modified version of DOS that could use Apple's 3.5" UniDisk drive

Refining the Original (1979-1980)

DOS underwent two more major upgrades before reaching maturity. Apple released DOS 3.2 in February of 1979. Several DOS commands were altered slightly, and new utility programs were included on the System Master disk. A minor upgrade, DOS 3.2.1, was released in July, 1979 to corrected a few minor bugs in DOS 3.2.

The last major revision of DOS, version 3.3, was released in August of 1980. DOS 3.3 would become the standard Apple II operating system for the next three and a half years. Even after the release of ProDOS, DOS 3.3 remained popular with developers. The complexity of ProDOS was unnecessary for games and educational software, and publishers continued to release software using DOS 3.3 for years.

DOS 3.3's main enhancement was the ability to read 16-sector disks. (DOS 3.2 disks were 13-sector disks.) New chips for the Disk II controller card allowed denser data storage using the same disk and drives. The controller card could also be used with DOS 3.2, but DOS 3.3 could not read disks formatted with DOS 3.2. It was necessary to start up a DOS 3.2 disk (using a special program called BOOT13) to read DOS 3.2 disks. A conversion program, known as MUFFIN, was also included, to allow users to move files from DOS 3.2 disks to DOS 3.3 disks.

A final maintenance version of DOS 3.3 was released in January, 1983 with the Apple IIe computer. Among other things, this release included a "fast loader" for Integer BASIC (the older version of BASIC which could be loaded into a memory bank in 64K Apple IIs, such as the IIe) and a couple of bug fixes.

Apple Pascal (1979)

Although DOS 3.3 was the primary Apple II operating system in the early '80s, it was not the only one. In August, 1979, Apple released Apple Pascal, an integrated software development system (and Pascal language compiler) based on UCSD Pascal. Apple Pascal had its own operating system and file structure; special utilities were needed to convert files to and from DOS 3.3 disks. A FORTRAN compiler which used the same operating system was also released. Since it was specific to Apple Pascal, this operating system was not widely used, although a few commercially released programs did use the format.

Like DOS 3.3, Apple Pascal required the new disk controller chips, since it used 16-sector disks. It also required a memory expansion card which, since it was originally bundled with the Apple Pascal Language System, came to be known as the Language Card. This card became the standard way to expand a 48K Apple II to 64K.

CP/M

There were many different microcomputers on the market in the late '70s and early '80s. Although these computers often differed substantially on a hardware level, many shared a common operating system—CP/M. CP/M (Control Program for Microprocessors) was developed by Digital Research in the mid-'70s.

A fairly simple command line operating system, it quickly caught on and because a standard for computers based on the Z-80 or 8088 microprocessor. Because of its popularity, it was inevitable that CP/M would eventually come to the Apple. Several manufacturers made Z-80 peripheral cards to make the Apple CP/M-compatible. A number of companies (including Applied Engineering and Microsoft) marketed these cards and enjoyed healthy sales until CP/M began to die out in the mid-1980s. Ironically, it was supplanted by MS-DOS, which looks an awful lot like CP/M!

ProDOS (January, 1984)

Although the highlight of January, 1984 was probably the introduction of the Macintosh, the month also marked the arrival of ProDOS, a new operating system designed to replace DOS 3.3. ProDOS was not an enhanced DOS 3.3, but rather a "smaller" version of SOS (Sophisticated Operating System), the operating system used by the the Apple III. As such, ProDOS could neither read nor write DOS 3.3 disks, though it could read and write Apple III SOS disks.

Unlike DOS 3.3, ProDOS could support mass storage devices such as 3.5 inch drives and hard disks, without dividing them into smaller, floppy-disk sized "volumes." ProDOS also replaced the old flat-file directory system of DOS 3.3 with a hierarchical system providing directories and subdirectories to allow mass storage to be managed more effectively. ProDOS also extended the reach of the operating system, handling memory allocation and system interrupts as well as disk and file access.

DOS 3.3 remained popular with many users, but eventually ProDOS became the king of the hill. Apple started shipping ProDOS with new computers, and developers of productivity and business software switched over to ProDOS rapidly, taking advantage of the capabilities of the new operating system to create programs like AppleWorks (which actually started life an Apple III program called III E-Z Pieces).

A Chronological Leap

In the next installment we'll jump ahead to 1986, where the introduction of the Apple IIGS resulted in a completely new and powerful operating system. We'll also see how Apple continued to enhance the original ProDOS operating system. ■

TIMELINE

Jun 1978	Disk II and DOS 3
Jul 1979	DOS 3.1
Feb 1979	DOS 3.2
Jul 1979	DOS 3.2.1
Aug 1979	Apple Pascal
Aug 1980	DOS 3.3
Jan 1983	DOS 3.3 (IIe version)
Jan 1984	ProDOS
Sep 1986	ProDOS 8 v1.2 and ProDOS 16 1.0 (Apple IIGS System Software 1.0)
May 1987	Apple IIGS System Software 2.0
Sep 1987	Apple IIGS System Software 3.1 (First IIGS Finder)
May 1988	Apple IIGS System Software 3.2
Sep 1988	Apple IIGS System Software 4.0 (First version of GS/OS)
July 1989	Apple IIGS System Software 5.0
Dec 1989	Apple IIGS System Software 5.0.2
Dec 1990	Apple IIGS System Software 5.0.3
Feb 1991	Apple IIGS System Software 5.0.4
April 1992	Apple IIGS System Software 6.0
Dec 1992	Apple II System Software 4.0.1 (ProDOS 8 2.0.2)
May 1993	Apple IIGS System Software 6.0.1

PRODOS BUILDING BLOCKS: PART TWO

Editor's Note: We take a break from the Blockhead program I started last month to bring you an article which explains a file format common on the IIGS: the forked, or extended, file. This article not only tells you how these files are stored, but provides a means to access them from ProDOS 8.

BY R. FORREST HARDMAN

Have any of these things ever happened to you?

- You want to view a documentation file for some new piece of shareware, but it's unreadable. The file looks like a TXT file, but it sure doesn't act like one.

- You're forced to use *ClarisWorks* on a Macintosh to type in a report of some kind. You cleverly save it to a ProDOS disk in AppleWorks 2.0 format—but when you get back to your Apple IIe you find that something went wrong and the file won't show up in the Add Files screen. You can't load it as a text file either. Come to that, you can't access it in any way with any of your programs! Ack! You can't even *delete* it!

- Someone gives you a diskful of sound samples that you want to play using Michael Mahon's nifty 5-bit-resolution Sound Edit program—but they turn out to be "sound resource" files that you can't use or load in any way whatsoever.

If you've ever been in one of these situations, you're a victim of forked files. Introduced first on the Macintosh and then implemented in the operating system of the Apple IIGS, these bizarre files contain two parts: a data fork and a resource fork. It's like having two files with one name. The 8-bit Apple IIs don't use resources, so ProDOS 8 doesn't deal with them. Accessing a forked file from ProDOS 8 is, according to traditional wisdom, impossible. Fortunately, the entry for the word "impossible" in the Apple II user's dictionary reads, "See possible."

About Forked Files

What do we mean by "unreadable" files? It's no harder to disk-copy a floppy full of forked files than to copy a floppy full of ordinary files, as a few minutes with your ProDOS System Utilities Disk (or Copy II Plus, or whatever) will demonstrate. Forked files are, in fact, fundamentally just like any other files: blocks of data on the disk. They're merely organized in a slightly different way.

Ordinary files come with an index—an ordered list of all the blocks that the data is stored in. Forked files, in contrast, have two indexes, and a "note" that tells the computer—among other things—where the two indexes are kept. ProDOS 8 just isn't equipped to deal with files that have two indexes—it gets confused when it came to the note telling where the two indexes were kept.

On the other hand, it's possible to write a program that will understand the note, find the real indices—and copy the normally unaccessible data into two plain, ordinary files. In fact, it can even be done in BASIC, with a little help from assembly language.

Understanding the program in this article requires some technical knowledge. A good ProDOS book, say Apple's, or Gary Little's Exploring GS/OS and ProDOS 8, published by Addison-Wesley, is also handy.

But you don't need to pay any attention to the commentary to use the program; to do that, all you need to know is how to type in a BASIC program without making mistakes. When you use the program you'll need to know the pathname of the forked file. Fork Split will ask you for this pathname, along with a target pathname the resulting files should be stored—and does the rest all by itself.

Fork Split will also ask you for the file type of the new files—i.e., how the new files should be saved. For example, if you have what *ClarisWorks* called an AppleWorks word-processor file, you can specify "AWP"; "Teach" and other documentation files fare well as "TXT"; if you have a sound-resource file, a better choice would be "BIN" or perhaps "$D8".

The Program

Fork Split reads forked files block by block; conventional "READ" and "BLOAD" commands won't work. Thus, the first part of Fork Split is a routine designed to read individual blocks directly from the disk. After translation into a form that can be POKEd in from Applesoft, we wind up with a subroutine that looks like this:

```
6000 Y = 768 : REM a common start loca-
  tion for storing code like this
6010 READ X : IF X>-1 THEN POKE Y, X :
  Y=Y+1 : GOTO 6010
6030 RETURN
6040 DATA 32, 190, 222, 32, 248, 230,
  142, 63, 3, 32, 190, 222
```

```
6050 DATA 32, 103, 221, 32, 82, 231, 165,
  80, 166, 81, 141, 64
6060 DATA 3, 142, 65, 3, 32, 190, 222,
  32, 103, 221, 32, 82, 231
6070 DATA 165, 80, 166, 81, 141, 66, 3,
  142, 67, 3, 32, 0, 191
6080 DATA 128, 62, 3, 176, 1, 96, 32,
  139, 190, 76, 9, 190, 3
6100 DATA 0, 0, 0, 0, 0
6110 DATA -1
```

If you type that in and then issue a GOSUB 6000, you'll end up with a small machine-language program that reads a block of data from disk, stored in memory at location 768. This program itself is invoked with the instruction

```
CALL 768, device specification, address,
block
```

Block is the number of the block to read; for instance, a block of zero will retrieve the first block from the disk—the block the computer reads when you first turn it on. Address is the starting memory address to read block to. The device specification is a "difficult" way to express "slot and drive", which we'll discuss in a moment.

Oh, yes, one more thing: since we're going to read and write files as 512-byte blocks, we need one more scrap of code that will let us make sure the files we write end up the right length (in bytes, not blocks)—so add these lines (note the new line 6110):

```
6110 DATA 32, 190, 222, 32, 248, 230,
  142, 102, 3, 165, 253, 166
6120 DATA 254, 164, 255, 141, 103, 3,
  142, 104, 3, 140, 105, 3, 32
6130 DATA 0, 191, 208, 101, 3, 176, 212,
  96, 2, 0, 0, 0, 0, -1
```

With the above routines (which are actually machine-language code that gets stored in the computer's memory) we can read blocks from the disk. We will use these routines and a little knowledge of the forked file format to read forked files. Next we need some code to find out what file to read:

```
2000 PRINT:INPUT "Enter pathname of
  forked file, or <RTN> to quit: ";F$
2010 IF F$ = "" THEN RETURN
2020 IF LEFT$(F$,1)<>"/" THEN 2000: REM
  ensure root directory named
2030 T = 0 : FOR X=LEN (F$) TO 1 STEP -1
2040 IF MID$(F$,X,1) = "/" THEN T = X : X
  = 1
2050 NEXT : IF NOT T THEN 2000 : REM make
  sure there's a file specified
2070 X$ = LEFT$(F$,T) : F$ =
  RIGHT$(F$,LEN(F$)-T): REM split off file-
  name
2090 INPUT "Put split files where? /";P$
2100 IF LEFT$(P$,1)<>"/" THEN P$="/"+P$ :
  REM make sure path starts
2110 IF RIGHT$(P$,1) <> "/" THEN
  P$=P$+"/" : REM and ends with slash
2120 INPUT "Save files as type: ";T$ : REM
```

Figure: assembler source code.

NOTE: You do not need to type in and run this code; it is only for reference purposes.

```
                * READ_BLOCK call source code.
                * This routine is invoked as follows:
                *      CALL 768,IO,ADDR,BLK
                * where IO is the device unit number, ADDR is the
                * memory address to read the block to, and BLK is
                * the block number.
                * The routine will merrily read blocks to just about
                * any specified address so Be Careful when using it.
                *
                * Zero page stuff:
                *
                LINNUM  =       $50
                *
                * Applesoft calls:
                *
                FRMNUM  =       $DD67
                CHKCOM  =       $DEBE
                GETBYT  =       $E6F8
                GETADR  =       $E752
                *
                * and the ProDOS interface:
                *
                BADCALL =       $BE8B
                ERROUT  =       $BE09
                MLI             =       $BF00
                *
                                ORG     $300
                * get slot/drive spec into the X register:
000300: 20 BE DE                JSR     CHKCOM ;skip first comma
000303: 20 F8 E6                JSR     GETBYT ;grab the value
000306: 8E 3F 03                STX     RBPARMS+1      ;put it
where we can use it
                * fetch the address to read the block to:
000309: 20 BE DE                JSR     CHKCOM ;skip second
comma
00030C: 20 67 DD                JSR     FRMNUM ;fetch a 16-bit
value
00030F: 20 52 E7                JSR     GETADR ;into LINNUM,
LINNUM+1
000312: A5 50                   LDA     LINNUM
000314: A6 51                   LDX     LINNUM+1
000316: 8D 40 03                STA     RBPARMS+2  ;and put it,
etc.
000319: 8E 41 03                STX     RBPARMS+3
                * and finally get the block # the same way
00031C: 20 BE DE                JSR     CHKCOM
00031F: 20 67 DD                JSR     FRMNUM
000322: 20 52 E7                JSR     GETADR
000325: A5 50                   LDA     LINNUM
000327: A6 51                   LDX     LINNUM+1
000329: 8D 42 03                STA     RBPARMS+4
00032C: 8E 43 03                STX     RBPARMS+5
                * Make the ProDOS call:
00032F: 20 00 BF                JSR     MLI
000332: 80                      DFB     #$80      ;READ_BLOCK
000333: 3E 03                   DA      RBPARMS
000335: B0 01                   BCS     ERROR
000337: 60                      RTS
000338: 20 8B BE      ERROR JSR BADCALL ;cause a BASIC error
since
00033B: 4C 09 BE                JMP     ERROUT ;we had a read
failure
```

```
          *
00033E: 03          RBPARMS DFB   #$3          ;3 Parameters
00033F: 00                        DS     1           ;device
unit number here
000340: 00 00                     DS     2           ;where
to put the block
000342: 00 00                     DS     2           ;block
number to read
          *
          *End-of-file correction routine.
          *This routine is invoked as follows:
          *      CALL 836,refnum
          * where refnum is the reference number of the
(OPEN!)
          * file whose EOF needs adjusting.  The desired EOF
          * value must be placed in memory locations 253-255
          * prior to calling since there are no 3-byte integer
          * conversion routines in Applesoft to my knowledge.
          *
          * get refnum specification into the X register:
000344: 20 BE DE                  JSR    CHKCOM ;skip first comma
000347: 20 F8 E6                  JSR    GETBYT ;grab the value
00034A: 8E 66 03                  STX    SPARMS+1      ;put it
where we can use it
00034D: A5 FD               LDA   $FD          ;and the EOF
value...
00034F: A6 FE                     LDX    $FE
000351: A4 FF                     LDY    $FF
000353: 8D 67 03                  STA    SPARMS+2
000356: 8E 68 03                  STX    SPARMS+3
000359: 8C 69 03                  STY    SPARMS+4
          * Make the ProDOS call:
00035C: 20 00 BF                  JSR    MLI
00035F: D0                        DFB    #$D0          ;SET_EOF
000360: 65 03                     DA     SPARMS
000362: B0 D4                     BCS    ERROR
000364: 60                        RTS
          *
000365: 02          SPARMS  DFB   #$2          ;2 parameters
000366: 00                        DS     1           ;file
reference number
000367: 00 00 00                  DS     3           ;EOF
value
```

```
no validity check...careful!
```

That should be fairly easy to follow; a few INPUTs, some simple string alteration (turning something like "/MY.DISK/MY.FOLDER/MY.FORKED.FILE" into "/MY.DISK/MY.FOLDER/" and "MY.FORKED.FILE" in separate string variables). Next, something more complicated:

```
2125 ONERR GOTO 2190
2130 PRINT CHR$(4)"PREFIX"X$ : SL =  PEEK
  (48944) : POKE 216,0
2140 IO = SL : REM the system likes it
  straight; human translation thus
2160 SL = INT (SL/16): DR = 1: IF SL > 7
  THEN SL = SL - 8: DR = 2
2170 PRINT "Reading blocks from slot "SL"
  drive "DR
2180 RETURN
2190 CALL -3288 : POKE 216,0 : PRINT "Bad
  pathname."; : GOTO 2000
```

If you stick a directory name (e.g. "/MY.DISK/") into X$ and GOSUB 2125, this code scrap will print the slot and drive that the disk containing that directory resides on. Since we are going to be reading blocks from the disk directly, this is vital—for this purpose, ProDOS identifies disks by device numbers, not by name.

The PREFIX statement makes sure the specified directory is currently available (i.e. that the disk is in a disk drive somewhere); we then extract the device number (slot and drive) of that directory. The ONERR statement is a bit underdeveloped, dismissing every error from a bad or missing disk to an illegal pathname as a "bad pathname." This just means that the pathname specified, for whatever reason, can't be accessed.

The key to this subroutine is PEEK (48944). After every disk operation, this memory location contains the slot and drive of the most recently used disk device in a slightly encoded format. Since the PREFIX command accesses the desired disk in the process of looking for it—and since it furthermore stops looking when it's found the disk we've asked for—this number is just what we need. The slot number is multiplied by 16, and to this number is added the drive number (minus one!) times 128. Our routine performs the reverse mathematical operations to extract the actual slot and drive number.

Next comes something really complex: a routine to see whether the file exists, and if it is in fact forked.

If you've ever catalogued a disk in BASIC or done a List All Files in AppleWorks, you've probably noticed that subdirectories are listed right along with all the other files. This is because subdirectories are files much like any other files, which means in turn that they can be read much like other files. So let's read through the directory file specified by X$ and see what's there. We use the BLOAD command so that we can get at the raw information, rather than the formatted information returned by OPENing and READing a directory.

```
3000 OFF = 0 : REM offset into directory
3010 ONERR GOTO 3140
3015 PRINT
  CHR$(4)"BLOAD"X$",TDIR,A$2000,L$200,B"OF
  F:BASE = 8196:POKE 216,0
3020 MTCH = 0 : REM keep track of whether
  we found the forked file we wanted
3030 X = PEEK (BASE)
3040 NL = X * 16: NL = (NL - INT (NL /
  256) * 256)/16
3050 ST = INT (X/16) : REM dig out Name-
  Len, Storage Type
3060 IF ST < > 5 OR NL < > LEN (F$) THEN
  3110: REM can't be what we want
3070 FOR X = BASE + 1 TO BASE + NL
3080 MTCH = MTCH + ( CHR$ ( PEEK (X)) =
  MID$ (F$,X - BASE,1))
3090 NEXT :MTCH = (MTCH = LEN (F$))
3100 IF MTCH GOTO 3130
3110 BASE = BASE + 39: IF BASE < 8703
  GOTO 3020
3120 OFF = OFF + 512: GOTO 3010
3130 KP = PEEK (BASE + 17) + PEEK (BASE +
  18) * 256: RETURN
3140 CALL -3288 : IF PEEK (222) = 5 THEN
  MTCH = 0 : RETURN
3150 PRINT"Error "PEEK(222) : POKE 216,0:
  STOP
```

After setting X$ and F$ properly, a GOSUB 3000 will read the first block (512 bytes) of directory file X$, then search for the file F$ in that block. On standard disks there are 13 file entries in each directory block, each 39 bytes long, so we'll look at them one at a time, starting at relative position 0, then 39, then 78, and so on.

The X value picked up in line 3030 contains both the length of the name of the particular file that's being kept track of as well as a note as to whether or not it's a forked file. Obviously, if the file isn't forked or its name is the wrong length we'll want to go on to the next; ditto if the name itself doesn't match. If none of the files are what we're looking for, the next 512 bytes of the directory are read, and we repeat the process until we find what we want or the end of the directory file is reached.

If there is a match, this is indicated by setting MTCH to something other than zero; further, the number of the block containing the note that tells where the forked file's indexes are located is returned in the KP ("Key Pointer") variable.

After a successful GOSUB 3000, we need to dig out the "note" that tells us where the two forks are:

```
4000 CALL 768, IO, 8192, KP : REM read
  key pointer block to $2000
4010 S1 = PEEK (8192) : S2 = PEEK (8448)
4020 K1 = PEEK (8193) + PEEK (8194)*256 :
  K2 = PEEK (8449) + PEEK (8450)* 256
4030 B1 = PEEK (8195) + PEEK (8196)*256 :
  B2 = PEEK (8451) + PEEK (8452)* 256
4040 D1 = PEEK (8197): D2 = PEEK (8198) :
  D3 = PEEK (8199)
4050 R1 = PEEK (8453): R2 = PEEK (8454) :
  R3 = PEEK (8455)
4060 RETURN
```

Lotta numbers there. What do they all mean? Block number KP contains information on the forks of the file, and the code above just reads block KP and peeks out that data. K1 and K2 end up containing the numbers of the first block (the "key" block) of the data fork and resource fork respectively; S1 and S2 tell us how to interpret those first blocks, as we'll see in a few paragraphs.

B1 and B2 are the size, in blocks, of the data and resource forks. D1 through D3 and R1 through R3 are the size of the data and resource forks in bytes. (These 3-byte values would usually be PEEKed out with a line like "X = D1 + (D2*256) + (D3*65536)", but for a technical reason we'll be leaving them as separate numbers.)

At any rate, with the information gleaned by a GOSUB 4000 we can begin the massive undertaking of reading the forked file!

Forked files, as mentioned earlier, come in two sections. These sections will end up being split into two files; one's name will begin with "D." and the other's with "R.". The letters stand for Data and Resource, which are the names for the two sections of the forked file. (Which file is the one you want? Well, that varies. Usually you'll find what you want in the "D." file, but not always. Load both and see!)

```
5000 PRINT "Copying data fork"; : FS$ =
  P$+"D."+LEFT$(F$,13)+",T"+T$
5005 ONERR GOTO 5015
5010 PRINT CHR$(4)"CREATE"FS$ : POKE
  216,0 : GOTO 5020
5015 CALL -3288 : IF PEEK (222) <> 19
  THEN PRINT"Error "PEEK(222) : STOP
5020 BY = 0
5030 IF S1 = 1 THEN IF K1 THEN DB = K1 :
  GOSUB 5400 : REM Seedling fork
5040 IF S1 = 2 THEN IB = K1 : BC = B1 :
  GOSUB 5300 : REM Sapling fork
5050 IF S1 = 3 THEN MB = K1 : BC = B1 :
  GOSUB 5200 : REM Tree fork
5060 POKE 253, D1 : POKE 254, D2 : POKE
  255, D3
5070 PRINT CHR$(4)"OPEN"FS$
5080 CALL 836, PEEK (48848) : PRINT
  CHR$(4)"CLOSE"
```

Some of that makes sense. Line 5000 makes a new pathname, used to save the "D." file; after that the new file is created (line 5015 says that it's OK if that filename already exists, but that any other error is fatal). Then comes some subroutine-calling that handles the actual file reading and writing.

As we mentioned earlier, there are different ways to interpret the data in block number K1. Suppose we're dealing with a word processing file. If the file is very small, block K1 would be actual data—512 or fewer bytes of text. If the file is larger than 512 bytes, a level of complexity is added and block K1 will contain an ordered list of the data blocks containing the file's text. And if the file is very large, so large that one block can't contain the entire list of data blocks, block K1 will contain an ordered list of ordered lists of data blocks! The value of S1 tells us which of these possibilities is the case, and each possibility is dealt with in lines 5030-5050.

Lastly we open the just-written "D.WHATEVER" file and make sure its length is set correctly. That was so much fun we'll do it again with the "R.WHATEVER" file:

```
5100 PRINT: PRINT"Copying resource
  fork";: FS$ =
  P$+"R."+LEFT$(F$,13)+",T"+T$
5105 ONERR GOTO 5115
5110 PRINT CHR$(4)"CREATE"FS$ : POKE 216,
  0 : GOTO 5120
5115 CALL -3288 : IF PEEK (222) <> 19
  THEN PRINT "Error "PEEK(222) : STOP
5120 BY = 0
5130 IF S2 = 1 THEN IF K2 THEN DB = K2 :
  GOSUB 5400 : REM see above
5140 IF S2 = 2 THEN IB = K2 : BC = B2 :
  GOSUB 5300
5150 IF S2 = 3 THEN MB = K2 : BC = B2 :
  GOSUB 5200
5160 PRINT CHR$(4)"OPEN"FS$
5170 POKE 253, R1: POKE 254, R2: POKE
  255, R3
5180 CALL 836, PEEK (48848) : PRINT
  CHR$(4)"CLOSE"
5190 RETURN : REM all done!
```

Now comes the hardest part of all: the subroutines to read the forked file. Actually, reading a single data block is quite easy:

```
5400 CALL 768,IO,9216,DB:REM read data
  block to $2400
5410 PRINT CHR$(4)"BSAVE"FS$",
  A9216,L512,B"BY
5420 BY=BY+512 : RETURN: REM increment
  where-to-write-next-bit pointer; exit
```

And if the fork is one block long that's it! That's all we do, just GOSUB 5400. It's read, written and we're outta here.

If the fork's size ranges from 2 to 256 blocks long—i.e., the fork is what is called a "sapling"—we'll read the list of blocks, and then GOSUB 5400 repeatedly by calling the following routine:

```
5300 PRINT ".";: CALL 768,IO,8704,IB:REM
  index to $2200
5310 FOR IX = 0 TO BC : DB = PEEK
  (8704+IX) + PEEK(8960+IX)*256
5320 IF DB THEN PRINT ".";: GOSUB 5400 :
  REM read all blocks with data
5330 IF NOT DB THEN BY=BY+512
5340 NEXT : RETURN
```

Line 5300 fetches the list of blocks to address 8704 ($2200); we then loop through the list, calling the line 5400 subroutine for each data block (DB) listed. (Note that 5400 is not called for data block entries of zero, although the "where we'll write in the output file" pointer is incremented as though these nonexistent blocks had been written. This is to handle "sparse" forks; see a good ProDOS reference for an explanation of sparse files.)

Forks of 257 blocks or more—"tree" forks—are even more complicated, since they have a list of lists of block numbers; we handle this in a manner similar to the above, by calling 5300 repeatedly for each list of blocks.

```
5200 PRINT ".";: CALL 768,IO,8192,MB:REM
  master to $2000
5210 MX = 0 : T = BC : REM to count down
  the blocks
5220 IB = PEEK (8192+MX) +
  PEEK(8448+MX)*256 : REM fetch index
  block
5230 BC = 255 : IF T < 256 THEN BC = T :
  REM do proper # of data blocks
5240 IF BC > 0 THEN GOSUB 5300 : T = T -
  256 : MX = MX+1 : GOTO 5220
5250 RETURN
```

Good news: that's the hard part! All we need now is to tack on a master control routine at the beginning of the program, like so:

```
1000 PRINT CHR$(4)"PR#3": GOSUB 6000
1020 GOSUB 2000 :  IF F$ = "" THEN END
1040 GOSUB 3000 : REM and go look for it
1050 IF MTCH THEN GOSUB 4000 : GOSUB 5000
1060 IF NOT MTCH THEN PRINT "File does
  not exist or is not an extended file."
1070 GOTO 1020
```

Type in the whole program, type RUN, and off you go. There are all kinds of forked files out there waiting for you! As for me, I've got to...split. ■

Traveling Salesmen and Lawn Sprinklers

by Mike Westerfield

SUMMER FUN

We're installing a sprinkler system in our front yard this summer to try to transform our little patch of desert into a patch of something that resembles somewhere grass might like to live. With pick and shovel in hand, I had been trenching the yard. After several days of back-breaking work, I noticed I wasn't getting very far very fast, and took a moment to measure my progress. I was doing about four feet an hour. The complete system needs about 130 feet of trench.

"OK," I thought. "Sitting in front of the computer all day for the past few years obviously has not been good for me, and I deserve and need all this exercise."

Still, cracking through hardened sand that was on its way to becoming sandstone in just a few years, rather than a few thousand, I had a lot of time to think about better ways of digging trenches. With my body engaged and my mind wandering, I thought of the classic traveling salesman problem.

THE TRAVELING SALESMAN PROBLEM

The problem, of course, is to find the best way to trench the yard—so I don't have to dig a single extra foot. It turns out that this is a pretty common problem, and lots of people have worked on it for a long time.

The classic traveling salesman problem is to find a route through a group of cities, setting things up so the salesman travels to each and every city exactly once. Of course, you don't want your salesman to waste time on the road (or, worse, get himself into trouble with farmer's daughters), so the shortest route is the best one. The same idea applies to any problem where you have a series of goals and a price for getting from one to the other. The traveling salesman problem applies to phone lines, computer networks, building roads, and, yes, even sprinkler systems!

THE SPRINKLER SYSTEM

Let's look at a concrete (or at least hard sand) example. Figure 1 shows a diagram of my front yard, complete with letters labeling the sprinkler heads. O is the origin—the first sprinkler head, the one closest (fifty feet) to where the valves are located. Since the trench from the valves to the origin must be dug regardless of how we connect the other sprinklers, I've left it out of my calculations.

The problem, then, is to start at position O, then move to any other sprinkler head. From there, you can move to any sprinkler head that has not already been visited. You keep going until all of the sprinkler heads have been visited, all the while adding up the total distance traveled.

Of course, there are a lot of different ways to visit all the sprinkler heads. Starting from position O, you have 7 choices. Once you move to one of those seven sprinkler heads, there are 6 left, so there are 7*6 or 42 possible ways to visit the first two sprinkler heads. All together, there are 7! (factorial), or 5040, different ways to visit them all. One of those ways is shorter than all of the rest—or at least as short as any others—and it's that route we're after. So how do we find it? Simple—by trying them all!

Well, I wouldn't want to try 5040 different combinations by hand (or even with a pocket calculator), but my computer is more patient and better organized than I am. A simple recursive subroutine does the trick. In Listing 1, Init sets up an array that holds the distances between any two sprinkler heads, as well as an array that tells us which sprinkler heads have been reached so far. *LayPipe* does all of the work. We call *LayPipe* from the main program, telling it that we want to start at location O, that no trenching has been done so far, and that no heads have been visited. That last value gives us a simple way to stop the recursion, since we know we'll be done as soon as all 7 heads have been visited. It's also used as an index into order, an array that holds the heads we've visited in the order we've visited them.

Inside *LayPipe,* the first step is to check to see if we're done. If so, we check to see if the

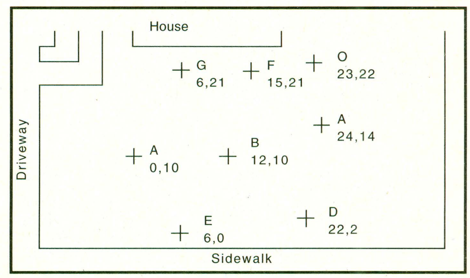


FIGURE 1

total length of the trenches is shorter than the best combination we've found so far. If it is, *LayPipe* prints the solution. Of course, a better solution might be found later, but I get a kick out of watching the program improve on its results. (Note that we've started the "best so far" variable, *minPipe,* at an absurdly high value of 10 to the 10th power—so the first solution the program finds will automatically be considered the best so far. Things will get better from there, though.)

If we're not done, *LayPipe* loops over all of the heads, visiting each head that has not been visited already, and calling itself to see what happens from there.

This is a pretty simple way to attack the problem, but it takes a while. On my Apple IIGS, which has an accelerator card, the program ran for 24 seconds, called *LayPipe* 13,700 times, and checked all of the 5040 possible solutions. Well, 24 seconds is no big deal—heck, it takes about half that just to compile and link the program. On the other hand, our problem only has 7 sprinkler heads. What if you're running cable for a network connecting 30 users? There are 30!, or 2.6E32, routes to check! The program would run for about 4E22 years! (That's 4 followed by 22 zeroes!)

In other words, the expansion is geometric. Each time you add a sprinkler head (or network node) you multiply the previous number of possible solutions by the new total number of heads (or nodes). The resulting number gets very large very quickly. There's gotta be a better way.

#1s Apple IIe System

CPU, 80-column card, monitor, disk drive, cables, Cyberlearn software.
($350 / $330 prepaid)

#1a Apple IIe System 128K

128K CPU, extended 80-col. card, monitor, disk drive, cables, Cyberlearn software.
($385 / $350 prepaid)

#1b Apple IIe Color System

128K CPU, extended 80-column card, Color Monitor (Amdek, Taxan, or similar), disk drive, cables, Cyberlearn software.
($545 / $520 prepaid)

#1c Apple IIe Deluxe Color System

128K CPU, extended 80-column card, hi-res Apple Color Monitor, two disk drives, cables, Cyberlearn software.
($675 / $650 prepaid)

#2a Apple IIc System

128K CPU, 80-column monitor, internal drive, printer & mouse ports, cables, Cyberlearn.
($325 / $295 prepaid)

The IIc is compatible with all IIe software!

#2b Apple IIc Color System

128K CPU, internal drive, Color Monitor (Amdek, Taxan, or similar), ports, cables, Cyberlearn.
($495 / $470 prepaid)

#2c Apple IIc Deluxe Color System

128K CPU, two disk drives, hi-res Apple Color Monitor, ports, cables, Cyberlearn.
($625 / $595 prepaid)

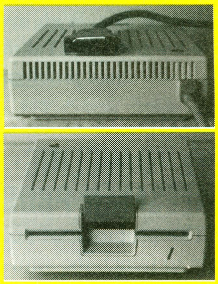
#3a Apple IIGS RGB System

1.25mb CPU, keyboard, mouse, hi-res RGB Color Monitor, 3.5 drive, cables, Cyberlearn.
($725 / $695 prepaid)

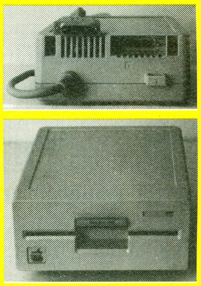
#3b Apple IIGS Deluxe RGB System

1.25mb CPU, keyboard, mouse, hi-res RGB Color Monitor, 3.5 drive, 5.25 drive, cables, Cyberlearn education software.
($820 / $795 prepaid)

#4a
Ribbon cable Drive for Apple IIe $59



#4b
19-pin cable Drive for IIe/c/GS $99



#4c
Daisy chain for IIe/c/GS, Mac LC $159

943003

Excess Inventory Specials!

Mail order and payment to:

Classroom Computer Co.
20 Sunnyside Avenue, M/S A-112
Mill Valley, CA 94941
Call 1-800-800-5115 or (415) 388-9000
ORDER BY FAX (415) 388-9090

Free printer with purchase of 5 or more computers!

(Free printer offer expires July 15, 1994)

All equipment comes standard with a 1-year warranty!

1-800-800-5115

Toll-free from anywhere in the U.S.!

Classrom Computer Co. offers Apple IIe, IIc, and IIGS computers, ImageWriter printers, and accessories at the lowest prices anywhere. CCC is run by experts in the Apple II business who have been dealing with them for over 12 years!

In order to bring you the lowest prices, most Apple II equipment is refurbished. We have a large inventory of computers which have been refurbished to original operating condition. All CCC products come with a free parts and labor warranty for 1-year. We can also provide inexpensive service of your existing computers and offer Extended Service plans for up to 5 years.

As a special bonus, CCC includes a free site license for Cyberlearn* educational software with every order for Apple IIe, IIc, or IIGS computers. Cyberlearn is the revolutionary $295 authoring system that received Electronic Learning Magazine's highest rating!

*Cyberlearn requires 128K. Limit 1 per customer.

#9a ImageWriter Printer

#9b ImageWriter II Printer

Printers include appropriate interface and cable when ordered with a computer.

√ **Check one:**

__ I have enclosed a check or money order.
__ I have enclosed a public school purchase order.

Orders from individuals and private schools must be prepaid. Net 30 terms are available to public schools only. Thank you!

Purchase Order Number: __________ (optional)
Name: __________
School: __________
Address: __________
City, State, Zip: __________
Telephone: (_____)__________

Quan.	#	Description	Terms	Unit	Total
	1s	Apple IIe System 64K		$350	
	1s	Apple IIe System 64K	PREPAID=	$330*	
	1a	Apple IIe System 128K		$385	
	1a	Apple IIe System 128K	PREPAID=	$350*	
	1b	Apple IIe Color System		$545	
	1b	Apple IIe Color System	PREPAID=	$520*	
	1c	Apple IIe Deluxe Color System		$675	
	1c	Apple IIe Deluxe Color	PREPAID=	$650*	
	2a	Apple IIc System 128K		$325	
	2a	Apple IIc System 128K	PREPAID=	$295*	
	2b	Apple IIc Color System		$495	
	2b	Apple IIc Color System	PREPAID=	$470*	
	2c	Apple IIc Deluxe Color System		$625	
	2c	Apple IIc Deluxe Color	PREPAID=	$595*	
	3a	Apple IIGS RGB System		$725	
	3a	Apple IIGS RGB System	PREPAID=	$695*	
	3b	Apple IIGS Deluxe RGB System		$820	
	3b	Apple IIGS Deluxe RGB	PREPAID=	$795*	
	4a	Ribbon Cable Drive for IIe		$ 59	
	4b	19-pin Disk Drive for IIe/IIc		$ 99	
	4c	Daisy Drive for IIe/c/GS/Mac LC		$159	
	4d	Duodisk for IIe		$169	
	4d	Duodisk Upgrade (with IIe purchase)		$ 69	
	5a	Extended 80-column card		$ 29	
	6a	Joystick for IIe/IIc/IIGS		$ 19	
	7a	Color Monitor (40-column)		$269	
	7b	Apple Color Monitor (80-col.)		$369	
	7c	Apple Monochrome Monitor		$ 49	
	9a	ImageWriter Printer		$199	
	9b	ImageWriter II Color Printer		$299	
	9c	Dot Matrix Printer & interface		$119	
	11a	Mouse for Apple IIc		$ 59	
	11b	Mouse & interface card for IIe		$ 89	
	13a	Super Serial Card		$ 59	
	13b	Grappler+ Card & cable for IIe		$ 49	
		Printer Cable (____to____)		$ 20	

We stock thousands of hardware and software accessories for the Apple IIe, IIc, or IIGS! Call for prices and list additional items below:

___ ___ __________ $___ _____
___ ___ __________ $___ _____

SUBTOTAL _____
Shipping and Handling (add 10% of subtotal) _____
Sales Tax (California only, add 7.25%) _____

All equipment comes standard with a one-year warranty. You can extend this to up to 5 years for a small additional charge.

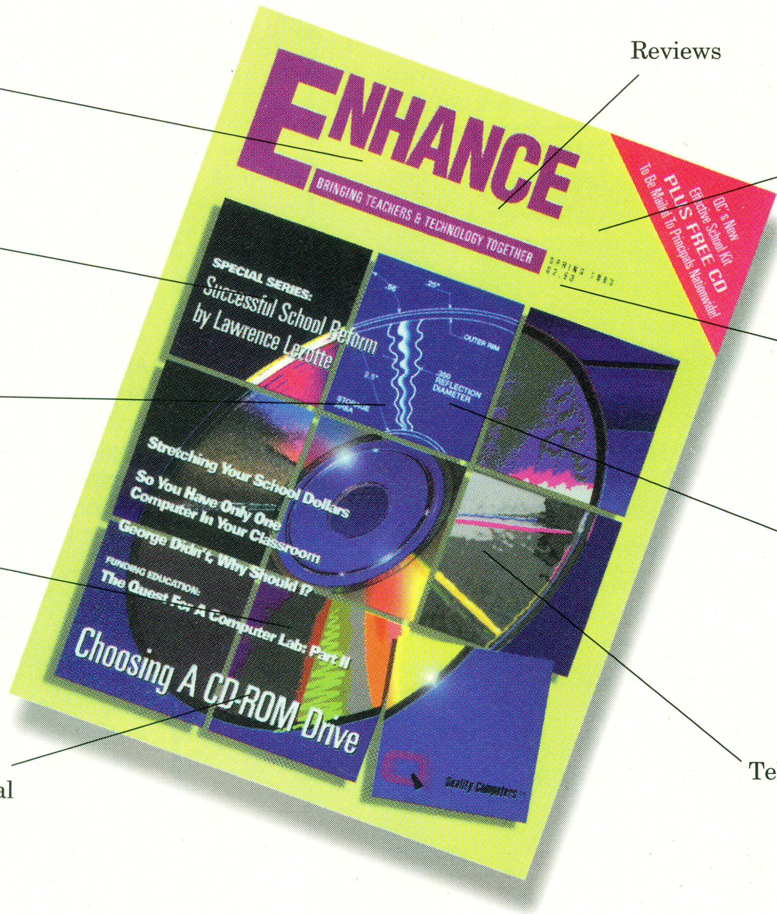
Extended Warranty (15%=3 years, 20%=5 years) _____
TOTAL DUE $ _____

* Prepaid discount prices are only available when payment is included with order. Orders received without payment will be billed at the regular price. Thank you.

Terms:

We ship most orders within two weeks. Backordered items may take longer. Choose carefully when ordering; orders may not be canceled, refunded, or exchanged after we receive them, unless authorized in writing, and subject to a restocking fee. Most Apple equipment is refurbished and comes with a 1-year warranty for free repair or replacement. Customer is responsible for shipping charges to and from CCC for warranty repair. Most equipment is manufactured by Apple; however, we may substitute monochrome monitors or disk drives by other manufacturers that are of equal or superior quality and performance to those made by Apple. Thank you. Apple IIe/IIc/IIGS and ImageWriter are registered trademarks of Apple Computer Inc.

Free help for teachers



Only Quality Computers gives you *Enhance*—our free publication designed to give computer-using teachers the kind of information they need to make an impact in their classrooms. Four times a year, *Enhance* gives you the kind of information you need to make your Apple the effective work/educational tool it was designed to be, like Product Reviews, information about Funding Your Classroom, Multimedia, Technical Tips, New Product Announcements, and more. You also get our complete catalog—full of all the hardware and software you need for Mac and IBM computers. Quality Computers has been publishing *Enhance* for over 6 years. It's free to all QC customers, or you can call for your free subscription today!

Quality Computers™ **1•800-777-3642**

If you know a teacher in education, help us spread the word that Quality Computers is the leader in educational technology!

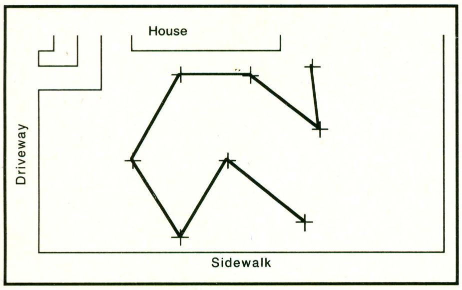


FIGURE 2

IMPROVING ON THE ALGORITHM

With a problem like this, a faster computer just isn't the answer. It doesn't hurt, of course, but what we need is a better way to solve the problem, not a faster computer. Even if you use a computer a thousand times faster than an Apple IIGS, the network problem is still going to take 40,000,000,000,000,000,000 years to solve (approximately).

There are some basic improvements we can make to the program's speed, and while they won't do well enough for a big problem with dozens of places to visit, it can't hurt to speed things up a bit even before we start changing the algorithm. The Apple IIGS is particularly slow at floating-point math calculations (those involving fractions). Hardware floating-point cards can speed things up quite a bit, and switching to integer math saves even more. Changing the program in Listing 1 to use long integers, then multiplying the lengths by 100, effectively gives you two decimal digit fixed point math. The finished program has enough accuracy to give the same answer as the floating-point version, but the long integer math version of the program runs in 1.6 seconds instead of 26 seconds—about 16 times faster.

Still, the real savings come when you find a better way to solve the problem. One of the simplest improvements is to stop working when you know the solution won't work. If you already have a solution with a length of 77 feet, and the total trenching for the solution you're looking at is longer than 77 feet when it's halfway finished, there's really no point in calling *LayPipe* further. For example, before the program starts considering routes that begin by digging a trench to sprinkler head B, it's already found the solution it will report. Even if it hadn't, it probably would have found a pretty good one. When the program starts looking at the route O-E-A-C, every single move from C to another unoccupied sprinkler head makes the solution longer than the best solution already found.

Listing 2 shows a modified version of LayPipe that doesn't bother calling to check the rest of a route once the total length of the trench exceeds the best solution it has already found. How much time does this save? In this case, the floating-point version of the program runs in 15 seconds, calling *LayPipe* 2009 times. And the best thing about this improvement is that it saves more and more steps the

LISTING 1

```
program Sprinkler (output);

const
  numHeads = 7;                         {# of heads}

type
  heads = (O, A, B, C, D, E, F, G);
  point = record
    x, y: real;
    end;

var
  visited: array[heads] of boolean;     {which heads have been visited?}
  loc: array[heads] of point;           {location of the heads}
  dist: array[heads, heads] of real;    {distance between the heads}
  minPipe: real;                        {min distance found so far}
  order: array[0..numHeads] of heads;   {order in which the heads were
visited}


  procedure WriteHead (h: heads);

  { Write the name of a sprinkler head            }
  {                                               }
  { parameters:                                   }
  {    h - sprinkler head                         }

  begin {WriteHead}
  if h = O then
      write('O')
    else
      write(chr(ord('A') - 1 + ord(h)));
    end; {WriteHead}


  procedure Initialize;

  { Set up the global variables                              }

  var
    h1, h2: heads;                     {loop/index variables}

  begin {Initialize}
  {no heads have been visited}
  for h1 := O to G do
      visited[h1] := false;

  {start with an absurdly high "minimum" distance found}
  minPipe := 1e10;

  {fill in the sprinkler locations}
  loc[O].x := 23.0;  loc[O].y := 22.0;
  loc[A].x := 24.0;  loc[A].y := 14.0;
  loc[B].x := 12.0;  loc[B].y := 10.0;
  loc[C].x :=  0.0;  loc[C].y := 10.0;
  loc[D].x := 22.0;  loc[D].y :=  2.0;
  loc[E].x :=  6.0;  loc[E].y :=  0.0;
  loc[F].x := 15.0;  loc[F].y := 21.0;
  loc[G].x :=  6.0;  loc[G].y := 21.0;

  {create an array of distances between the heads}
  for h1 := O to G do
      for h2 := O to G do
         dist[h1, h2] := sqrt(sqr(loc[h1].x - loc[h2].x)
                         + sqr(loc[h1].y - loc[h2].y));
   end; {Initialize}


   procedure LayPipe (head: heads; pipeDist: real; index: integer);

   { Lay pipe from head h to some other head              }
   {                                                      }
```

```
{ parameters:                                                }
{    head - head to lay the pipe from                        }
{    pipiDist - amount of pipe laid so far                   }
{    index - recursion level & order array index             }

var
   i: 1..numHeads;                  {loop/index variable}
   newHead: heads;                  {loop variable}

begin {LayPipe}
{mark this head as visited}
visited[head] := true;
order[index] := head;

if index = numHeads then begin

   {we have reached all heads - is it a minimum path?}
   if pipeDist < minPipe then begin

      {yes - print the path}
      minPipe := pipeDist;
      write('The route O');
      for i := 1 to numHeads do begin
         write('-');
         WriteHead(order[i]);
         end; {for}
      writeln(' uses ', pipeDist:1:1, ' feet of pipe.');
      end; {if}
   end {if}
else

   {try all of the possible heads}
   for newHead := A to G do
      if not visited[newHead] then
         LayPipe(newHead, pipeDist + dist[head, newHead],
            index + 1);

{finished with this head}
visited[head] := false;
end; {LayPipe}


begin
Initialize;
LayPipe(O, 0.0, 0);
end.
```

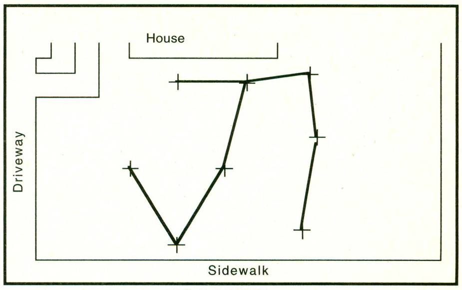


FIGURE 3

deeper the program has to search. If you're laying wire for a 30-node network, the savings will be far, far better than the 38% savings for seven sprinkler heads.

There are other ways to improve the program, too. The next best is to introduce a heuristic. A heuristic is the computer programmer's equivalent of a rule of thumb. You know, like "the temperature is colder in the winter than in the summer." That's not true all of the time—there are a few oddball days in any season—but it's true often enough that we can use it as a generalization.

For our problem, a good heuristic is "it's best to go from where you are to the next closest sprinkler head." That's not true all of the time, but it works most of the time.

If you're going to check all of the solutions anyway, using the heuristic doesn't do any good. But it does wonderful things when you combine it with the code that quits as soon as the path gets too long. It finds a pretty good solution right away, which helps cut off the really bad solutions that much earlier.

All of these tricks put together can cut the run time by many orders of magnitude, and there are other tricks that work for specific variations of the traveling salesman problem. If you're clever, you just might be able to solve that networking problem with 30 stations on an Apple IIGS—well within your lifetime.

A MORE REALISTIC SPRINKLER LAYOUT

Back at my soon-to-be lawn, though, we find another problem that has nothing to do with how fast the program runs. Figure 2 shows a diagram of the route the program found. Obviously, that just isn't the right way to lay pipe for a sprinkler system. The run from A-F is silly; it makes a lot more sense to split the pipe at the origin and run it from O-A and O-F. What we need is a slight modification to the traveling salesman problem. Instead of visiting every sprinkler head exactly once, we'll check any solution that goes from any head that has been visited to any head that has not been visited.

Listing 3 shows the modified program. The bookkeeping is a little more complicated, since we have to keep track of both the starting and ending sprinkler heads, but program isn't really all that different. The most significant change is in *LayPipe,* where we loop through all the sprinkler heads that have been visited, checking each one against the sprinkler heads that have not been visited. There's a shortcut there, too. Instead of actually checking a route from each of the old sprinkler heads to the new one, the program merely scans the lengths to each of the sprinkler heads that has already been visited, and picks the shortest connection right away.

The program runs for 98 seconds, calling *LayPipe* 5005 times and looking at four complete solutions. While the program calls LayPipe a lot less times than our original program, it takes four times as long to run—mostly due to the added floating-point calculations.

My tired back was pleased to see an improvement of 5.1 feet. The trench length dropped from 77.1 feet to 72 feet. Figure 3 shows the layout.

THE FINAL LAYOUT

Still, the shortest distance between a point and a line segment isn't a line to the *end* of the line segment. Digging right to the sprinkler head, the run from B to F is 12.4 feet. There's a pipe from F to G, though, and digging from B to the pipe is only 12 feet. And that's the final finesse I tried.

Listing 4 shows a much more complicated version of the program. This time, instead of looking at solutions from any sprinkler head that has been visited, the program looks for routes to the pipe connecting the previously visited sprinkler heads. That really cranks up the level of the floating-point calculations because the program has to figure out exactly where the pipe is. A lot of solutions that seemed pretty dumb using the old rules worked a lot better this way, so the program had to look at more cases. After all, an initial

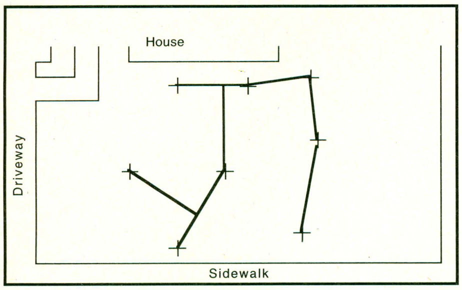


FIGURE 4

LISTING 2

```
procedure LayPipe (head: heads; pipeDist: real; index: integer);

{ Lay pipe from head h to some other head                    }
{                                                            }
{ parameters:                                                }
{    head - head to lay the pipe from                        }
{    pipiDist - amount of pipe laid so far                   }
{    index - recursion level & order array index             }

var
   i: 1..numHeads;                  {loop/index variable}
   newDist: real;                   {distance including a new head}
   newHead: heads;                  {loop variable}

begin {LayPipe}
{mark this head as visited}
visited[head] := true;
order[index] := head;

if index = numHeads then begin

   {we have reached all heads – is it a minimum path?}
   if pipeDist < minPipe then begin

      {yes - print the path}
      minPipe := pipeDist;
      write('The route O');
      for i := 1 to numHeads do begin
         write('-');
         WriteHead(order[i]);
         end; {for}
      writeln(' uses ', pipeDist:1:1, ' feet of pipe.');
      end; {if}
   end {if}
else

   {try all of the possible heads}
   for newHead := A to G do
      if not visited[newHead] then begin
         newDist := pipeDist + dist[head, newHead];
         if newDist < minPipe then
            LayPipe(newHead, newDist, index + 1);
         end; {if}

{finished with this head}
visited[head] := false;
end; {LayPipe}
```

trench from O to C seems pretty silly when you have to connect directly to another sprinkler head, but it looks a lot better when you can run a trench from B directly to the longer trench.

This version of the program called LayPipe 361,731 times, examining 35 solutions. The floating-point math got way out of hand. With no hardware boost to the floating-point, this program ran for *25 hours.*

Well, as Roger Wagner is fond of pointing out, "Anything worth doing is worth overdoing!" And I think this time, I overdid it.

Still, the solution you see in Figure 4 is nearly two feet shorter than the previous best. Coming in at 70.2 feet of trench, it was a welcome solution. And I'd rather sleep while my computer crunches the numbers than dig, any day.

OK, IT COULD BE BETTER

This still isn't the best possible solution, but other refinements are a bit outside of the scope of the traveling salesman problem. It's possible to look at solutions that don't run directly to a sprinkler head, but that involves a completely different kind of program, one that iterates from an approximate solution to a better one.

Renting a trencher instead of digging by hand is another solution—that way, I wouldn't be so concerned about saving a foot or two. That seems like a cop-out, but it's really not. Douglas Hofstadter, the author of the classic *Gödel, Escher, Bach,* calls this kind of thinking "jumping out of the system"—changing a basic premise of the problem to solve it faster. If you need a good solution, but not necessarily the best solution, you can take a lot of shortcuts. If I'd had a trencher, the second solution or maybe even the first would have been just fine. Or maybe I would have just eyeballed it.

In any case, there's always a better solution. Programs are never finished, they just ship.

While I was working on the program, another improvement came from an unexpected source. For two whole days, it rained. (That's pretty unusual for Albuquerque.) All of that Universal Solvent soaking into the sandstone-to-be of my lawn set things back a few centuries. Instead of making a measly four feet an hour, I started cruising through the dirt at 15 feet an hour.

So did I use this crazy patchwork solution just to save a few minutes of digging?

You bet I did!

FURTHER READING

The traveling salesman problem is a classic, and you can find it in any number of sources. One pretty good discussion is in *Algorithms,* by Robert Sedgewick, Addison-Wesley, 1983. It's a good book to have on your bookshelf for other things, too. ■

LISTING 3

```
program Sprinkler (output);

const
   numHeads = 7;                      {# of heads}

type
   heads = (O, A, B, C, D, E, F, G);  {sprinkler heads}
   point = record                     {location in the
yard}
      x, y: real;
      end;
   pipePath = record                  {path for one length
                                       of pipe}
      head1, head2: heads;
      end;

var
   visited: array[heads] of boolean;  {which heads have
                                       been visited?}
   loc: array[heads] of point;        {location of the
                                       heads}
   dist: array[heads, heads] of real; {distance between
                                       the heads}
   minPipe: real;                     {min distance found
                                       so far}
   order: array[0..numHeads] of pipePath; {order in which the
                                       heads were visited}


   procedure WriteHead (h: heads);

   { Write the name of a sprinkler head                    }
   {                                                       }
   { parameters:                                           }
   {     h - sprinkler head                                }

   begin {WriteHead}
   if h = O then
      write('O')
   else
      write(chr(ord('A') - 1 + ord(h)));
   end; {WriteHead}


   procedure Initialize;

   { Set up the global variables                           }

   var
      h1, h2: heads;                {loop/index variables}

   begin {Initialize}
   {no heads have been visited}
   for h1 := O to G do
      visited[h1] := false;

   {start with an absurdly high "minimum" distance found}
   minPipe := 1e10;

   {fill in the sprinkler locations}
   loc[O].x := 23.0;  loc[O].y := 22.0;
   loc[A].x := 24.0;  loc[A].y := 14.0;
   loc[B].x := 12.0;  loc[B].y := 10.0;
   loc[C].x :=  0.0;  loc[C].y := 10.0;
   loc[D].x := 22.0;  loc[D].y :=  2.0;
   loc[E].x :=  6.0;  loc[E].y :=  0.0;
   loc[F].x := 15.0;  loc[F].y := 21.0;
   loc[G].x :=  6.0;  loc[G].y := 21.0;

   {create an array of distances between the heads}
   for h1 := O to G do
      for h2 := O to G do
         dist[h1, h2] := sqrt(sqr(loc[h1].x - loc[h2].x)
                           + sqr(loc[h1].y - loc[h2].y));
   end; {Initialize}


   procedure LayPipe (head1, head2: heads; pipeDist: real;
index: integer);

   { Lay pipe from head h to some other head                  }
   {                                                          }
   { parameters:                                              }
   {    head1 - pipe just laid starts at this head            }
   {    head2 - pipe just laid ends at this head              }
   {    pipiDist - amount of pipe laid so far                 }
   {    index - recursion level & order array index           }

   var
      connect: heads;        {best connection found so far}
      cDist: real;           {distance from newHead to connect}
      i: 1..numHeads;        {loop/index variable}
      newDist: real;         {distance including a new head}
      oldHead, newHead: heads;      {loop variables}

   begin {LayPipe}
   {mark this head as visited}
   visited[head2] := true;
   order[index].head1 := head1;
   order[index].head2 := head2;

   if index = numHeads then begin

      {we have reached all heads - is it a minimum path?}
      if pipeDist < minPipe then begin

         {yes - print the path}
         minPipe := pipeDist;
         write('The route ');
         for i := 1 to numHeads do begin
            WriteHead(order[i].head1);
            write('-');
            WriteHead(order[i].head2);
            if i <> numHeads then
               write(', ');
            end; {for}
         writeln(' uses ', pipeDist:1:1, ' feet of pipe.');
         end; {if}
      end {if}
   else

      {try all of the possible heads}
      for newHead := A to G do
         if not visited[newHead] then begin

            {find the closest head to this one}
            connect := O;
            cDist := dist[O, newHead];
            for oldHead := A to G do
               if visited[oldHead] then
                  if dist[oldHead, newHead] < cDist then begin
                     connect := oldHead;
                     cDist := dist[oldHead, newHead];
                     end; {if}

            {evaluate the connection}
            newDist := pipeDist + cDist;
            if newDist < minPipe then
               LayPipe(connect, newHead, newDist, index + 1);
            end; {if}

   {finished with this head}
   visited[head2] := false;
   end; {LayPipe}
```

Continued

Apple II Peripherals - Upgrades - Service Parts - Hard Disks
School & University P.O.s Accepted! - School Qty. Discounts Available*

RAM Chips

Apple //e or][+ **4164 /4116**..50¢
Expanding your... Apple IIgs Memory Card, Æ GS-RAM, Æ RAMWorks, Super Expander & Compatibles
768K Exp. Set (24 41256) **$24.00**
41256..**$1.25**
Expanding Your GS Juice+, S&S ÆGS RAM + / II , CV Tech or Chinook 4000
41024 (1 Meg) **$3.00**
Bank of 8 ....**$24.00**
Expanding Your Æ GS-RAM Ultra or RAMPak4 GS
44256 (256x4) **$4.00**
Bank of 8 ....**$32.00**
Expanding Your OctoRAM, GS Sauce, Q-RAM GS, Pacemark 4 Meg RAM Card
1 Meg SIMM **$24.00**
4 Meg SIMM **$135.00**
NCS/Æ GS RAM III (1x4)....**$25.00**

Motherboards/Service Parts

//e or //c M/B W/Exch $89.00
//e or //c Motherboard $129.00
//e Keyboard Exch $45.00
II+ MThrbd. / Kybd $49.00/ ..$29.00
//e, //c , / IIgs Case $25./ ...$35.00
IIgs Battery (ROM 03).............**$4.95**
//c+ Parts available..............Call

Power Supplies

II+,//e Exch/No Exch. $35/$45.00
IIgs Exch / New .... $59 /$79.00
//e High Power add $10.00
//c Replacement Ext P/S......$29.00
//c External Power Supply ...$39.00

Modems

USR 1200 Modem (NEW) $19.00
Hayes SM1200 (Refurb).........$29.00
2400 Hyundai External...........$49.00

Refurbished Printers

ImageWriter I..........................$129.00
ImageWriter I (Wide) $139.00
ImageWriter II $199.00
IW-I/II Ribbons............**6 for...$12.00**
IWII Color Ribbon....................**$5.45**

Print Heads

IW-I (Exch Only) $35.00
IW-II (Exch. Only).....................$79.00
We rebuilt print heads

Accessories

16 Pin Joystick][+,//e...........................$9.95
9 Pin Joystick , IIgs $12.95
Apple IIgs System 6.0.1 & Manual$35.00
Apple ADB Mouse II (New).........**$69.00**
ADB Mouse Exch $39.00 .Refurb $49.00
Mouse Pad (Static Free) **$2.95**
//e Enhancement Kit **$49.00**
IIgs Internal Fan....................................**$14.95**

PC Transporter

101 Keyboard for PCT $29.00
PCT IBM Keyboard Cable......$7.00
3.5" 720K Ad-on Drive.........**$49.00**
5.25" Transdrive......................**$69.00**
4464 Zip /PCT Mem. Exp......$2. ea
8087-2 (M ath Co.) $39.00

Peripheral Cards

Zip GS 7Mhz / 32K Cache ..$159.00
71256 -45ns (SRAM)..................**$9.95**
Zip Chip 8 Mhz (//e, //c) **$139.00**
Super Serial Card $52.00
IW-II 32K Buffer......................**$30.00**
20 Pin 5.25" Disk Cont $39.00
19 Pin 5.25" Disk Cont. $45.00
][+ 80 Column Card $45.00
GrafStar II Graphic Parallel..$35.00
Graphic Parallel (Grappler) $45.00
AEC SCSI Card $59.00
16K Language Card (][+) $17.00
Mountain Clock (][+,//e) $24.00
VisionPlus Ehanced $209.00

Cables & Switch Boxes

METAL, GBBS Sysops. HST/v.32bis hardware hndshk cables! (Specify).....$14.95
DuoDisk Replacement Cable...$19.00
19 Pin Drive Converter $14.00
IIgs,e,c to ImageWriter I/II.....**$7.95**
Laser 128 Parallel Cable.........$14.95
//c Serial --> Parallel Converter..$35.00
Fullnet Connector **$14.95**
Switch Boxes........... $19.00 - $29.00
ADB Replacement cable.........**$6.95**
SCSI Cable, 25-50 or 50-50.......$9.95

Apple Chips & ROMs

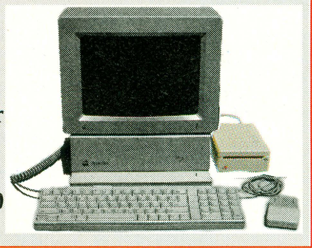
Call for Chips Not Listed!
//e & //c Related
IOU/MMU 344-0020/0010 ..$19.00
ROM 342-0303 or 0304..............$19.00
ROM 322-0134 or 0135.............$9.00
ROM 342-0133 or 0265.............$9.00
Keyboard ROM 342-0132 ....$9.00
HAL 342-0170 ....$15.00
KYBD Encoder AY3600PRO.$29.00
6502...$3.00
65C02...$7.00
//c Specific
IOU/MMU 344-0021/0011 ..$19.00
ROM 344-0272...........................$9.00
ROM (3.5" Drive) 344-0033 ...$29.00
TMG or GLU $15.00
6551...$3.00
IWM 344-0041 ....$24.00
IIgs Related
65C816 (Dip) $19.00
VGC 344S0046-C (Upgrade) .$29.00
344-0077-B (ROM 01) **$39.00**
Other IIgs Chips Available
II+ Related
ROM Set (C0-F8)......................$29.00
Other II+Chips..$3.00 to $7.00Call

Apple IIgs System

$599.00

System includes AppleColor RGB Monitor External Apple 3.5" Drive • Keyboard Mouse • 1024K • ROM 03

Apple //e System • Refurb $299.00
128K • DuoDisk • Green Monitor



SoundMeister!

"....S is for Sound!"
Bring out the sound in your IIgs! Alltech Electronics now has the SoundMeister Stereo/Digitizer card. Unleash those hidden sounds in your IIgs today! Includes 2 disks of sounds/music!:.................$69.00
SoundMeister & Speakers $79.95

Bulk Disks

5.25" DSDD Disks 100 for $12.00
3.5" DSDD Disks 50 for $14.50
3.5" DSHD Disks 10 for $3.90
5.25" Disk Sleeves 100 for $3.00

Ask for our Mac & IBM Product Lists

Alltech Electronics Co.

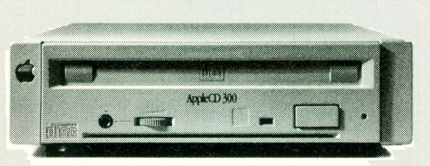
602 Garrison St. Oceanside, CA 92054
9-6 PM Mon.-Fri. - 9-5 PM Sat.

Orders Only:

VISA MasterCard

800/995-7773

All other Inquiries: & Tech Support: 619/721-7733
Fax (School PO's, Quotes, Info.): 619/721-2823
BBS (FutureNet #30) 300-14.4 v.32: 619/721-0705



DiscQUEST..... $79.00

DiscQuest: CD ROM for your IIgs!
Sequential Systems CD ROM Software
DiscQuest (Incl. Family Doctor)..........$79.00
DiscQuest & CD 150 ....$265.00
DiscQuest , CD 150 & RAMFast $395.00
Apple CD-ROM 150 External..............$189.00

Hard Disk Accessories

Apple //e & IIgs hard disks available.
Call for larger drive sizes
ProAPP //c External 20 Meg. Limited Supply
Requires Unidisk ROM..........................**$259.00**

Cables & Cases

25/25, 25/50 or 50/50 SCSI Cable............$9.95
SCSI Passthrough 50 pin terminator $12.95
SCSI Case & P/S for 3.5" or 5.25" HH $45.00

Sequential Systems

Q:Talk LTO (AppleTalk & 32K for IW-II)...$69.00
RAM //c with 1024K (//c or //c+) **$129.00**
bit Mouse card $49.00 ....with mouse $65.00
SuperCOMM $54.00.....SuperCOMM /c.$45.00

RAM Cards

RAM Expansion for Apple //e & IIgs Available
Extended 80 Column Card.........................$19.00
Super Expander //c (Older //c's, 1024K).....$119.00

Monitors

IIGS Compatible (Atari) RGB Monitor......$129.00
AppleColor RGB, IIGS (Refurb) **$169.00**
AppleColor RGB IIGS (New).....................$225.00
Apple Monitor][, /// (Refurb)..$49.00/ ..$39.00
Apple Monitor //c (Refurb)......................$59.00
Apple Color 100 RGB ....$109 / $199.00
Apple Color Composite (Refurb) .$159 / $199.00
//c RGB Monitor (CGA Conversion) **$129.00**
//c RGB Converter Box (Uses CGA).........$39.00
Composite Green (Refurb)..........................$29.00
9" Composite Green "Box" monitor $35.00
Color Composite (Not for 80 Col Txt)........$69.00

With Allison™ Digitizing software Instantly 'grab's from any video source. Upgrades available. Call .

VisionPlusEnhanced IIgs $179.00

Apple FDHD Ext. $199

Disk Drives

5.25" Generic Full Height (Refurb)............$59.00
5.25" Generic Full Height w/19 Pin.........$69.00
5.25" Apple Disk II (Refurb) add $10.00 for above.
Apple DuoDisk 5.25" Exch $129.00
Apple DuoDisk 5.25" (Refurb)..................$159.00
Apple Disk //c Assy Exch ..$75.00...........$85.00
AppleDisk 3.5" (Refurb)............................$165.00
Apple FDHD External (1.44) (NEW) **$199.00**
AppleDisk 5.25" (Refurb)............................$165.00
Apple 3.5" 800K Drive Service Exchange: Uni-Disk, AppleDisk(Platinum), Mac $85.00

CA Residents please add 7.75% Sales Tax • COD Orders, Cash/Certified Funds Only • Minimum Shipping & Handling Charge $5.00 (This covers most orders) • Some products are refurbished products and carry a 120 day warranty. • New products are covered by a one year warranty. All Repairs are warranted for 120 days. • Some RAM Chips may be new but pulled, all RAM has a one year warranty. • NO Returns without an RMA number, please call before shipping anything back to us. • A 20% Restocking Fee is applied to all returns due to incompatibility or unneeded product or Products purchased in error. Please be sure what computer you have before placing an order. • Educational P.O.s Accepted, Others on AOC only. • Prices and availability subject to change at any time without any notification whatsoever. *Discounts on certain merchandise only and min. Qty Applies.

computer clubs

To get more information about the Apple User Group nearest you,

If you want your computer club to be mentioned in II Alive, send a letter describing your club to:

Quality Computers
c/o Audrey Wolfe
P.O. Box 349
St. Clair Shores, MI 48080

ALASKA

Anchorage Apple Users Group
P.O. Box 110753
Anchorage, AK 99511-0753
Contact: Timothy Odell 373-7459

Apple Mousse User Group
P.O. Box 80176
Fairbanks, AK 99708
Contact: Jesse Atencio (907) 456-1333
$15 per year

ARIZONA

Apple-Res
9052 E. Sahuaro Dr.
Scottsdale, AZ 85260
$25 per year

Arizona Apple User Group
3035 E. Topaz Circle
Phoenix, AZ 85028-4423
(602) 423-9202

Tucson Apple Core
P.O. Box 43176
Tucson, AZ 85733-3176
Contact: Clay Evitts (602) 296-5491 days
$20 per year
BBS: (602) 882-2945

ARKANSAS

Apple Tree of the Ozarks
HC 62 Box 540
Flippen, AR 76234
$20 per year; $15 initiation

CALIFORNIA

Apple Corp of San Diego
P.O. Box 87964
San Diego, CA 92138-7964
Contact: Tom Kasner (619) 484-0695 Ext. 6

Appleholics Anonymous Apple II User Group
3875 Telegraph Rd. Suite A202
Ventura, CA 93003
Contact: Tony Pizza (805) 482-3453
$12 per year

AppleJacks of Inland Empire
Contact: Larry (909) 864-2309
BBS: (909) 369-6637

Apple Sac
P.O. Box 254645
Sacramento, CA 95825
Contact: Heidi Bylsma (916) 486-8326
New $30, Renewal $25
BBS: Future Vision (Metal) 481-1096

Fresno Apple II Computer Users Group
P.O. Box 16155
Fresno, CA 93755

GravenStein Apple IIGS Users Group
P.O. Box 964
Petaluma, CA 94953-0964
$25 per year per family
BBS: (707) 794-9333

Newton's Fruit Users Group
14639 Cashew St.
Hesperia, CA 92345-2702
BBS: (619) 956-2631

Orange Apple Computer Club
25422 Trabuco Rd., Bldg 105, Ste-251
El Toro, CA 92630
(714) 770-1865
$25 per year

Original Apple Corps
P.O. Box 90065
Los Angeles, CA 90009
Contact: Fred Duffy (310) 450-3336
BBS: (310) 454-4660

Peninsula Apple User Group
Redwood City, CA
Contact: Roger Lakner 367-8657

P.I.E. (Programming & Interfacing Enthusiasts, Inc.)
P.O. Box 2185
Santa Clara, CA 95055
Contact: Dean Nichols (408) 241-2185
$20 per year
BBS: (408) 733-4670

Stockton Apple Users Grou
6333 Pacific Ave., Suite 186
Stockton, CA 95207

Tri-City Apple User Group
P.O. Box 93123
Pasadena, CA 91109
(213) 258-0281
$20 per year
BBS: (818) 288-5640

Tri Valley Apple II User Group (TVAIIUG)
P.O. Box 2096
Dublin, CA 94568
Contact: Jerry Carleton (510) 828-0959

Valley Apple Computer Club
12978 Crowley St.
Arleta, CA 91331
Contact: William Trent (818) 988-1752
$24
BBS: (818) 504-9750

COLORADO

Denver Apple Pi
P.O. Box 280668
Lakewood, CO 80228-0668
$18 plus $7 new member application fee
BBS: (303) 987-3647

CONNECTICUT

Applelist Computer Club
P.O. Box 6053
Hamden, CT 06517

Appleshare
P.O. Box 200
Greens Farms, CT 06436
Contact: Joan Hoffman (203) 259-8513
$20 per year family membership

Hartford User Group Exchange (H.U.G.E.)
P.O. Box 380027
East Hartford, CT 06138-0027
Contact: Edward Sposito (203) 635-0557
$24
BBS: Bit Bucket (203) 257-9588

SMALL User Group
119 Red Stone Hill
Plainville, CT 06062-2608
Contact: Linda Frechette (203) 747-2036

DELAWARE

Delaware Valley
Apple IIGS Computer Club
P.O. Box 5956
Wilmington, DE 19808-0956
$20

FLORIDA

Apple Computer Enjoyment Society (A.C.E.S.)
P.O. Box 291557
Fort Lauderdale, FL 33329-1557
1-800-924-4709 & (305) 584-5923
$30 1st year; $25 renewal

South Broward Chapter (A.C.E.S.)
see above
BBS: (305) 431-5189

West Palm Beach Chapter (A.C.E.S.)
see above
BBS: (407) 483-8426

Apple Tree of Central Florida
2810 Nela Ave.
Orlando, FL 32809
$35 annually
BBS: 366-0156

Spring Hill Apple Computer Enthusiasts (SPACE)
11418 Long Hill Court
Spring Hill, FL 34609
$20
(904) 686-7069

SunCoast Apple Club
3300 Duncan Ave.
Sarasota, FL 34239
Contact: Shirley Cook (813) 922-1351
$20 per year, $10 for students

SunCoast Apple Tree
P.O. Box 7488
Clearwater, FL 34618
$25 per year
BBS: (813) 347-5104

SWACKS Apple Computer Club
c/o L.E. McLaughlin
384 Lancaster Ave.
Port Charlotte, FL 33952
$20 per year; $12 for Newsletter only

GEORGIA

Computer User Group (Any Type)
110 Peachtree Rd.
Rockmart, GA 30153
Contact: Donald Sullivan
(404) 684-5909
$15 per year

HAWAII

Hawaii Macintosh & Apple Users' Society
P.O. Box 29554
Honolulu, HI 96820-1954
Contact: Eugene Villaluz (808) 735-3750
$24 per year

IDAHO

Apple Boise User Group (ABUG)
934 River Park Lane
Boise, ID 83706
Contact: George Nummy (208) 344-9506
$12 per year

ILLINOIS

Apple Tree Computer Club
P.O. Box 823
Homewood, IL 60430-0823
Contact: Mary Ann Trzyna
(815) 469-1961
$28 family, $14 auxiliary per year
BBS: (708) 481-2211

Aurora Area Apple Corps
P.O. Box 2901
Aurora, IL 60507-2901
Contact: Howard Katz (708) 879-5818
$20

DAUG (Dupage Apple Users Group)
P.O. Box 294
Downers Grove, IL 60515

Lincolnwood Apple Users Group
2926 Leanne Court
Northbrook, IL 60062
(708) 480-8812

Northern Illinois Computer Society
P.O. Box 547
Arlington Heights, IL 60006
New $30, Renewal $24,
includes entire family
BBS: (312) 351-4374

Northshore Apple Users Group
c/o Babette Simon
5331 Carol
Skokie, IL 60077
Contact: Babette Simon (708) 967-7483
Family $20 per year

INDIANA

Apple Pickers
P.O. Box 20136
Indianapolis, IN 46220
Contact: Vincent Roth (317) 357-6207 after 5
New $25, $30 per family per year;
Renewal $20
BBS: (317) 897-1989

Apple Users Group of Michiana
P.O. Box 11398
South Bend, IN 46634-1398
$15 per year

Fort Wayne
Apple Computer Users' Group
P.O. Box 10004
Ft. Wayne, IN 46850-0004
$15 per year

Northwest Indiana Apple Users Group
7526 Independence St.
Merrillville, IN 46410
Contact: Nate Gagliardi 762-6818
$14 per year

IOWA

Applebyter Computer Club
P.O. Box 2092
Davenport, IA 52809
Contact: Shawn Beattie
BBS: 788-0314

Green Apple User Group
P.O. Box 171
Waterloo, IA 50701
Contact: Virgil Berg (319) 232-1842
$15 per year

Metro Apple Computer Hobbyists
P.O. Box 176
Crescent, IA 51526-0176
$20 per year

Roland Story Apple User's Group
P.O. Box 407
Roland, IA 50236-0407
Contact: Dave Graham (515) 388-4700
$10 per year

KANSAS

Apple Bits Users Group (ABUG)
P.O. Box 368
Shawnee Mission, KS 66201
Contact: Sandy Brockman
(816) 523-1007
$30 first year; $25 renewal

Apple Tree User Group, Inc.
306 West 5th Street
Larned, KS 67550
Contact: Shane Blanchett
$15 Initiation Fee; $20 Individual, $25 Family

Parsons Apple Users Group
P.O. Box 1081
Parsons, KS 67357
Contact: Randall Swearengin (316) 421-6953
Family $12 per yr, Full-time students $6 per yr
BBS: (316) 421-2027

Plane Apple User's Group
P.O. Box 47396
Wichita, KS 67201
Contact: Duston James (316) 685-2174;
Steve Specht (316) 265-5539
$24 per year
OMEGA PRO (316) 721-7735

Topeka Area Apple Group
5419 SW 28th St.
Topeka, KS 66614-1713
Contact: Ron Hurd (913) 272-5033
$15 family

KENTUCKY

Louisville Computer Society
P.O. Box 9021
Louisville, KY 40209-9021
$26

LOUISIANA

Cresent City Apple Core
2334 New York Street
New Orleans, LA 70122-5657
Contact: Dr. James H. Abbot (504) 283-3819

MAINE

Northwoods IIGS User Group
P.O. Box 550
Milford, ME 04461-0550
$15 per year

MARYLAND

Maryland Apple Corp.
Contact: Dave Smythe (410) 882-9234

Washington Apple Pi, Ltd.
7910 Woodmont Ave., Suite 910
Bethesda, MD 20814
(301) 654-8060

MASSACHUSETTS

Cape Cod Apple Users Group
P.O. Box 48
South Dennis, MA 02660
Contact: Ron Church (508) 540-2517
$20 per year

New England Apple Tree
26 Monument Street
Medford, MA 02155
$24 per year

MICHIGAN

Apple P.I.E.
P.O. Box 5055
Warren, MI 48090-5055
$25 per year

groups specializing in a certain subject, or how to start a new group, call Apple Computer 1-800-538-9696, ext. 500. or write to: The Apple User Group Connection • Apple Computer, Inc. • 20525 Mariani Ave., M/S 36AA • Cupertino, CA 95014

Apples for the Teachers
161 Cass Ave.
Mt. Clemens, MI 48043
Contact: Jim Wenzloff (313) 469-7206

Flint Apple Club
P.O. Box 460
Flint MI 48501
$20 per year
BBS: (313) 230-7754

Grand Rapids Apple II Users Group
P.O. Box 1811
Grand Rapids, MI 49501

Lansing Users Group
P.O. Box 27144
Lansing, MI 48909-7144
$18

Michigan Apple Computer User Group
P.O. Box 567
Warren, MI 48090-0567
$25 to start, $20 renewal per year

Midland Apple Club
1710 West St. Andrews
Midland, MI 48640

MINNESOTA

Lake Superior Apple Users Group
Duluth, MN
Contact: Don Jacobson (218) 723-4349

Minnesota Apple Computer Users Group
P.O. Box 796
Hopkins, MN 55343
Contact: Rand Sibet (612) 566-8571
$25 per year, $15 student

MISSOURI

American Public Domain Club
5821 Kerth Rd.
St. Louis, MO 63128
Contact: Michael Young
$12

Apple Squires of the Ozarks
P.O. Box 3986
Springfield, MO 65808-3986
Contact: Doug Kahler 833-4362
$15 initiation fee; $20 individual

Monsanto Apple Users Group
2 Pem Road
St. Louis, MO 63146-5407
Contact: John B. Wilson (314) 694-2447
(Leave voice mail)
GEnie E-Mail Address: JBWILSON
$10 per year–Monthly Newsletter

NEBRASKA

Apple-Link
P.O. Box 22246
Lincoln, NE 68542
$10 per year

NEVADA

Southern Nevada Apple Family User Group
P.O. Box 12715
Las Vegas, NV 89112-1715
Contact: George Lewis (702) 364-9093
BBS: Apples Only (702) 646-7007

NEW JERSEY

Bergen Apple Special Interest Club (B.A.S.I.C.)
The BASIC FACTS; 8-11 Creek Ct.
Fair Lawn, NJ 07410
Contact: Philip Smallen 791-3355
$15 per year
SSA-BBS: 472-8312

North Jersey Mac Apple User Group
P.O. Box 215 (WOB)
West Orange, NJ 07052-0215
Contact: Pete Crosta (201) 667-6369
$25 per year

Ocean County Apple Users Group
25 Long Road
Freehold, NJ 07728
Contact: Matt Weiss (908) 431-2339
$15 per year

Ocean/Monmouth Apple Users Group
55 Meadowbrook Road
Brick, NJ 08723-7848
Contact: Bill Scratchley (908) 920-3833
$15 per year

Princeton Apple II Users Group
100 Sixth Ave.
Trenton, NJ 08619-3223
$12 per year

South Jersey Apple User's Group
P.O. Box 4273
Cherry Hill, NJ 08003-4273
Contact: Jack Bullion 767-4913
$20 single/family, $10 student
SJAUG APPLELINE 424-1382

NEW MEXICO

Applequerque Computer Club
P.O. Box 35508
Albuquerque, NM 87176-5508

NEW YORK

BAUG (Big Apple Users Group)
P.O. Box 1822
Old Chelsea Station
New York, NY 10113-0945
(212) 603-9007

CRAB-Apple (County of Rockland Apple Branch)
P.O. Box 268
W. Nyack, NY 10994-0268
$10 per year,
free 1/2 year membership for new members

GAB'er (Guilderland Apple Byters)
Roger C. Mazula, Editor
28 Hay Path
Watervliet, NY 12189
Contact: John Buckley (518) 371-3624
$10 per year, $5 initiation fee
BBS: Pro Vanilla (518) 462-5953

Mixed Burnt Hills Apple Group
171 Birch Lane
Scotia, NY 12302
$8 per year
BBS: Plain Vanilla 462-5953

Putnam Valley
Educators Apple Users Group
142 Peekskill Hollow Rd.
Putnam Valley, NY 10579
Contact: Frank Reale (914) 528-8101

SACC Apple Computer Club
147 Deforest Rd.
Dix Hills, NY 11746
Contact: Gregory Dib (516) 499-7681
$10 per year

NORTH CAROLINA

Carolina Apple Core
P.O. Box 31424
Raleigh, NC 27622
$18 per year
CAC 783-9010; NIEHS 541-0041

Charlotte Apple Computer Club
P.O. Box 221913
Charlotte, NC 28222
BBS: 563-6233

Eamon Adventurer's Guild
7625 Hawkhaven Dr.
Clemmons, NC 27012
Contact: Tom Zuchowski (919) 766-7490
$7 per year
T.ZUCHOWSKI on GEnie

Triad Apple Core
c/o GTCC Small Business ASST. Center
2007 Yancyville St., Suite 220
Greensboro, NC 27405

OHIO

Apple-Dayton, Inc.
P.O. Box 3240
Dayton, OH 45401-3240
$25 per year

Apple GS Columbus
P.O. Box 27072
Columbus, OH 43227-0072
Contact: John Ledford (614) 855-0937
BBS: (614) 475-9791

AppleSiders of Cincinnati
P.O. Box 14277
Cincinnati, OH 45250-0277
Contact: Ralph Kitts (513) 829-1161
$25 per year
BBS: (606) 572-5375 or (606) 572-5685

COACH (Central Ohio Apple Computer Hobbyists)
P.O. Box 09028
Bexley, OH 43209
Contact: Mike Goodrich (614) 866-4860
BBS: (614) 262-4946

NEO Apple Corps
c/o Nancy Abbott
1935 Mattingly Rd.
Hinckley, OH 44233

OREGON

Appleeugene
907 River Road #289
Eugene, OR 97404
Contact: Larry Badten 895-2605
$15 per year

Portland Apple II User Group
P.O. Box 1608
Beaverton, OR 97075-1608
$20 1st year; $15 thereafter

Willamette Apple Connection
P.O. Box 18435
Salem, OR 97305
Contact: Neal Layton (503) 370-9527
$15 per year
WAC BBS (503) 363-0861

PENNSYLVANIA

Apple Butler Users Group
P.O. Box 39 Meridian Station
Butler, PA 16001-0039
$20 Single, $25 family

Columbia Apple Pi
c/o L.A. Winski, M.D.
P.O. Box 710
Millville, PA 17846
$6 per year

Delaware Valley Apple Branch
132 Eaton Dr.
Wayne, PA 19087
Contact: Curt Wilson (215) 473-6199
$15 annually

Erie Apple Crunchers, Inc.
P.O. Box 1575
Erie, PA 16507
$5 Initiation fee; $25 per year
E.A.C. Express (814) 838-8510

Hershey Apple Core
P.O. Box 634
Hershey, PA 17033
HAC Hotline: (717) 531-1300
The Library BBS: (717) 566-1699

Keystone Apple Core
c/o William C. Miller, President
1789 Braggtown Road
East Berlin, PA 17316
$10 annually

Philadelphia Area Computer Society SIG
P.O. Box 312
LaSalle University
Philadelphia, PA 19141
Contact: Martin Friedman (215) 353-2753
AOL: Meesh1 BBS: (215) 842-9600
$30 per year

RHODE ISLAND

Rhode Island Apple Group
P.O. Box 4726
Rumford, RI 02916-4726
$10 initiation; $20 annual dues

SOUTH CAROLINA

CSRA Apple Users Group
1342 Moultrie Drive
Aiken, SC 29803
$15 year

Greenville Apple Users Group
P.O. Box 4937
Greenville, SC 29608
Contact: Doug Durkee (803) 244-7396
$20 per year

TENNESSEE

AppleCore of Memphis
P.O. Box 241002
Memphis, TN 38124-1002
$20 per year

Music City Apple Core
c/o Gerald Dooley
1085 Woodcock Hollow Road
Kingston Springs, TN 37082
Contact: Gerald Dooley 952-2367; George Emge 833-1508; Everett Hertenstein 262-4778

TEXAS

Apple Valley Computer Club
Tony Rodriguez, Pres. 5900 N. 28th Lane
McAllen, TX 78504
Contact: Tony Rodriguez 682-9625
$10 per year

Coastal Bend Users Group (CBUG)
P.O. Box 8391
Corpus Christi, TX 78468-8391
$12 per year per household
STIX (512) 992-4855

San Antonio Appleseed
P.O. Box 290028
San Antonio, TX 78280-1428
New Membership $15, Renewal $10

Tarrant Apple Group (TAG)
912 West Broadway Ave.
Fort Worth, TX 76104
Contact: Bob Baggott (817) 332-3341
$15 per year

VIRGINIA

Northern Virginia Apple Users Group
P.O. Box 8211
Falls Church, VA 22041
PRO-NOVA BBS: (703) 671-0416

TideWater AppleWorms
P.O. Box 68097
Virginia Beach, VA 23455
Contact: Wayne Sheffield (804) 340-6799

WASHINGTON

TechAlliance CO-OP
P.O. Box 3308
Kent, WA 98032-0206
(206) 870-1556
BBS: (Seattle) (206) 281-8298
BBS: (206) 277-6070
$30 per year (King & Pierce Counties); $25 per year (Remainder of U.S.); $35 per year (Foreign)

WEST VIRGINIA

Apple Users Group of Charleston
2105 Weberwood Drive
South Charleston, WV 25303
Contact: John Howell 343-6422, or
Chas. Szasz 965-6965
$10 per year

Club Apple User Group
125 North Pinch Rd.
Elkview, WV 25071
$10

WISCONSIN

Milwaukee Apple Users Symposium
9818 W. Sheridan Ave.
Milwaukee, WI 53225
Contact: Helmut Wittbecker

Racine Area Users Group
P.O. Box 085152
Racine, WI 53408
$10/Family per year

Wisconsin Apple Users Club, Inc.
P.O. Box 20998
Milwaukee, WI 53220-0998
Contact: Bruce Kosbab (414) 771-6086

AUSTRALIA

Apple Users Society of Melbourne (A.U.S.O.M.)
P.O. Box 1071 Narre Warren MDA
Narre Warren, VIC 3805
Ph: (613) 796-7553

Tazmanian Apple User Group
P.O. Box 188
N. Hubart, Tasmania 7002 AUSTRALIA
10021 485200
Contact: Jim Fraser

CANADA

Apple SAUCE
Box 480 RPO, University
Saskatoon, Sask., Canada S7N 4J8
Contact: Brock Chatson (306) 374-0095
$20 per year
BBS: SAUCE (306) 955-8828

Apples Victoria Users Group
9184 Cresswell Rd.
RR 2, Sidney, B.C. V8L 3S1
Contact: Shirley Demeriez
(604) 656-2845 6-9PM
$24 per year, $30 family

Apples B.C. Computing Society
P.O. Box 80569
Burnaby, B.C. V5H 3X9
$35 per year, $30 renewal

Calgary Apple Corps Computer Club
Box 8, Station 'T'
Calgary, Alberta, Canada T2H 2G7
C Renzo Costa (403) 251-0525
$30 per year for individuals or families

Kelowna Apple Users Group
1622 West Kelowna Rd.
Kelowna, B.C., Canada V1Z 3B7
Contact: Robert Ashton
(604) 769-3140 6 to 9 pm
$2 per month

London Apple Corps
1555 Glenora Dr., Suite 103
London, Ontario, Canada N5X NV7

Winnipeg Apple Users' Group
P.O. Box 1798
Winnipeg, MB R3C 3R1
Contact: Don Soutter 256-0095
$20 indiv, $25 family; $5 initiation fee
Applebox BBS: 224-0683

ENGLAND

Cambridge Apple User Group
22 High Street
Cambridge CB4 1NG
England (U.K.)
Contact: Ian Archibald

The British IIGS Club
41 High Street
Great Shelford
Cambridge CB2 5EH
England (U.K.)
Contact: Peter Stark
(Produces bimonthly 'members' disks)

FRANCE

GS Club
6, impasse La Croix-Pommier,
94120 Fontenay-sous-Bois.
Téléphone: (1) 48.77.11.32
AppleLink: GS.CLUB

GERMANY

Kaiserslautern Apple Users Group
PSC 1 Box 8851
APO AE 09012

Ramstein Apple Club
PSC 2 Box 18
APO AE 09012

Rhein-Neckar Apple Users Group (RNAUG)
P.O. Box 525,
APO New York, NY 09063
Contact: James Clark
$10 per year

ITALY

Apple II Survivor Club Italy
Vis Dal Fabbro 4
37122 Verona Italy
Contact: Manuel Turtula

```
begin
Initialize;
LayPipe(O, O, 0.0, 0);
end.
```

LISTING 4

```
program Sprinkler (output);

const
   numHeads = 7;                          {# of heads}

type
   heads = (O, A, B, C, D, E, F, G);    {sprinkler heads}
   point = record                       {location in the yard}
      x, y: extended;
      end;
   pipePath = record              {path for one length of pipe}
      connect: point;             {location of the connection}
      head: heads;                {sprinkler head being connected}
      end;

var
   visited: array[heads] of boolean;      {which heads have
                                          been visited?}
   loc: array[heads] of point;            {location of the
                                          heads}
   minPipe: extended;                     {min distance found
                                          so far}
   order: array[0..numHeads] of pipePath; {order in which the
                                          heads were visited}


   procedure WriteHead (h: heads);

   { Write the name of a sprinkler head                       }
   {                                                          }
   { parameters:                                              }
   {    h - sprinkler head                                    }

   begin {WriteHead}
   if h = O then
      write('O')
   else
      write(chr(ord('A') - 1 + ord(h)));
   end; {WriteHead}

   procedure Initialize;

   { Set up the global variables                              }

   var
      h1, h2: heads;                     {loop/index variables}

   begin {Initialize}
   {no heads have been visited}
   for h1 := O to G do
      visited[h1] := false;

   {start with an absurdly high "minimum" distance found}
   minPipe := 1e10;

   {fill in the sprinkler locations}
   loc[O].x := 23.0;  loc[O].y := 22.0;
   loc[A].x := 24.0;  loc[A].y := 14.0;
   loc[B].x := 12.0;  loc[B].y := 10.0;
   loc[C].x :=  0.0;  loc[C].y := 10.0;
   loc[D].x := 22.0;  loc[D].y :=  2.0;
   loc[E].x :=  6.0;  loc[E].y :=  0.0;
   loc[F].x := 15.0;  loc[F].y := 21.0;
   loc[G].x :=  6.0;  loc[G].y := 21.0;
   end; {Initialize}
   procedure LayPipe (x, y: extended; head: heads; pipeDist:
extended;
      index: integer);

   { Lay pipe from head h to some other head                    }
   {                                                            }
   { parameters:                                                }
   {    x, y - location to connect to                           }
   {    head - pipe just laid ends at this head                 }
   {    pipiDist - amount of pipe laid so far                   }
   {    index - recursion level & order array index             }

   var
      cDist: extended;        {distance from newHead to cX, cY}
      cX, cY: extended;       {location of best connection}
      i: 1..numHeads;         {loop/index variable}
      newDist: extended;      {distance including a new head}
      oldHead, newHead: heads;     {loop variables}
      p: 0..numHeads;        {loop/index variable; pipe number}

      function LineDist (x1, y1, x2, y2, x3, y3: extended;
         var x, y: extended): extended;

      { Find the distance from a point to a line segment       }
      {                                                        }
      { parameters:                                            }
      {    x1, y1 - first point on the line segment           }
      {    x2, y2 - second point on the line segment           }
      {    x3, y3 - point to find the distance to             }
      {    x, y - (output) location of the connecting point    }
      {                                                        }
      { Returns: Distance from the point to the line segment   }

      var
         a, c, d: extended;             {misc. temp values}

      begin {LineDist}

      {find x, y (the point closest to the line through x1, y1
and x2, y2)}
      if x1 = x2 then begin

         {find x, y for lines with an infinite slope}
         x := x1;
         y := y3;
         end {if}
      else if y1 = y2 then begin

         {find x, y for lines with a zero slope}
         x := x3;
         y := y1;
         end {else if}
      else begin

         {find x, y for all other segments}
     a := (y2 - y1)/(x2 - x1);
         c := y1 - a*x1;
     d := 1.0/(sqr(a) + 1.0);
     x := (x3 + a*y3 - a*c)*d;
     y := (a*(x3 + a*y3) + c)*d;
         end; {else}

      {if x, y are not on the segment, pin them to an end
point}
      if x1 = x2 then begin
         if y < y1 then begin
            if y < y2 then begin
               if y1 < y2 then begin
                  y := y1;
                  x := x1;
                  end {if}
               else begin
                  y := y2;
                  x := x2;
                  end; {else}
               end; {if}
```

Continued

```
         end {if}
      else
         if y > y2 then begin
            if y1 < y2 then begin
               y := y2;
               x := x2;
               end {if}
            else begin
               y := y1;
               x := x1;
               end; {else}
            end; {if}
      end {if}
   else begin
      if x < x1 then begin
         if x < x2 then begin
            if x1 < x2 then begin
               x := x1;
               y := y1;
               end {if}
            else begin
               x := x2;
               y := y2;
               end; {else}
            end; {if}
         end {if}
      else
         if x > x2 then begin
            if x1 < x2 then begin
               x := x2;
               y := y2;
               end {if}
            else begin
               x := x1;
               y := y1;
               end; {else}
            end; {if}
      end; {else}

   {return the distance}
   LineDist := sqrt(sqr(x - x3) + sqr(y - y3));
   end; {LineDist}


begin {LayPipe}
{mark this head as visited}
visited[head] := true;
order[index].connect.x := x;
```

```
   order[index].connect.y := y;
   order[index].head := head;

   if index = numHeads then begin

      {print the new minimum path}
      minPipe := pipeDist;
      write('The route ');
      for i := 1 to numHeads do begin
         write('(', order[i].connect.x:1:1, ',', order[i].con-
nect.y:1:1, ')-');
         WriteHead(order[i].head);
         if i <> numHeads then
            write(', ');
         end; {for}
      writeln(' uses ', pipeDist:1:1, ' feet of pipe.');
      end {if}
   else

      {try all of the possible heads}
      for newHead := A to G do
         if not visited[newHead] then begin

            {evaluate all possible connections}
            for p := 0 to index do begin
               cDist := LineDist(order[p].connect.x,
order[p].connect.y,
                                 loc[order[p].head].x,
loc[order[p].head].y,
                                 loc[newHead].x, loc[newHead].y,
                                 cX, cY);
               newDist := pipeDist + cDist;
               if newDist < minPipe then
                  LayPipe(cX, cY, newHead, newDist, index + 1);
               end; {for}
            end; {if}

   {finished with this head}
   visited[head] := false;
   end; {LayPipe}


begin
Initialize;
LayPipe(loc[0].x, loc[0].y, 0, 0.0, 0);
end.
```

Letters

Continued from page 5

WHAT MISTAKES?

The Applesoft BASIC translation of Mike Westerfield's baseball simulation ("Major League Drag," May/June, page 53) has a couple of errors. The sqr function in Pascal takes the square of a number, but the identically-named SQR function in BASIC takes the square root of a number. Your editor hasn't used Pascal in a while and missed this little tidbit. Lines 150 and 160 should therefore read:

```
150 VX = VX - DT * SGN (VX) * VX
  ^ 2 * CD * DA * A / (2.0 * M):
  REM drag x
160 VY = VY - DT * SGN (VY) * VY
  ^ 2 * CD * DA * A  / (2.0 * M):
  REM drag y
```

Sirtech is alive and well, according to John Wanner, who informed us via America Online that the company can be reached by phone at 315-393-6633, or by post at P.O. Box 245, Ogdensburg, NY 13669.

In the AppleWorks GS Spring Training article, one formula given in the table had an extra opening parenthesis, which causes an error when attempting to evaluate the formula. The correct formula for Slugging Percent is:

```
=(J3+L3+(2*M3)+(3*N3))/H3
```

The article on disk recovery had some addresses switched around. Harold Portnoy is the author of Change-A-File and does not sell *ProSel-16.* The address given for Glen Bredon, the author of *ProSel 16,* was also incorrect. The correct address can be found in the virus protection article, elsewhere in the magazine. Thanks to Chuck Newby of Charlie's Appleseeds for finding this error.

The review of Quinsept's *Family Roots* program (May/June) listed the program's retail price as $59.95. The correct price is $179.95.

Joe Kohn, the publisher of Shareware Solutions II, was *amused to death* that we didn't tell you where to send your check or money order. We mentioned his publication in the May/June issue's News & Rumors section, but neglected to give the address, which is: Joe Kohn, Shareware Solutions II, 155 Alpine St., San Rafael, CA 94901-1008. ■

MarketPlace

CLASSIFIEDS

WANTED: AE RAMFactor Card. (409) 776-5513

HARD DRIVE REPAIR & DATA RECOVERY: Sider, Xebec, Trustor, DataMac, ProAPP, Apple Profiles, SCSI, Applied Ingenuity, Apple SCSI, and orphan drives. Call for others. DATA TRANSFERS (Apple, MAC & PC). B&D Computer Repair, 6115 S. Massasoit, Chicago, IL 60638 (312) 735-9010 Call between 9:30 am and 1 pm.

SUPERLEAD TEACHES THE BASIC READING SKILLS. Each of the six color coded disks tracks 22 students as they progress through the program. Complete manual includes pre and post tests for each level. Introductory offer for $40.00 contains manual and disk #1. Complete set for $180.00 or each additional disk for $35.00. Operates on Apple II and Macs with IIe card. Send for "Free Catalog" to LEAD Educational Resources, Inc. 144 Main Street, N. Bridgewater, CT 06752 (203) 355-1516.

APPLE II PUBLIC DOMAIN: Request FREE catalog or send $2 for demo disk and catalog (refundable). Buy as low as 80¢ per disk. Since 1986. Next day shipping. Visa/Master Card. CALOKE INDUSTRIES (Dept. RS), P.O. Box 18477, Raytown, MO 64133.

68502's FOR SALE $25. Development software available. Hybrid Concepts, 3102 Frankfort Ave., Louisville, KY, 40206, (502) 893-5529.

USED COMPUTERS $$$ WANTED! $$$

–Dead or Alive–

Apple IIe
Apple IIc
Apple IIGS
Macintosh
ImageWriters
Accessories

Trade Apple II Computers toward the purchase of new Macintosh computers or receive cash for them!

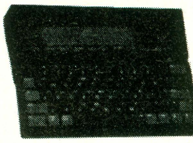
Computers for Schools

Nationwide pickup. FAX to (415) 289-7562 or

Call 1-800-800-6525

Laptop for Apple / Mac / IBM

128K PC4
2 lb w/case



Now with e-mail option

Built-in software: WP w/spell, DB, SS, BASIC
Transfer files between Apple II, Mac, IBM
Uses 4-AA batteries—A/C adapter included
LCD scrollable screen 4 lines by 40 char
Prints directly to most printers

PC4 for Mac or IBM **$239**

BridgeIt installs to AppleWorks menu.
Send files between PC4 and AppleWorks – $35

Apple II Compatibles

	Computer	w/Amber	w/Color
LASER 128	**289**	**414**	**488**
LASER 128EX	**359**	**484**	**558**

Largest LASER Repair Center & Educational Dealer

2400 Modem 79
Joystick 25
No-slot clock..............36
IIe 128K Card..............25
Hard Drive w/card - Laser 128, IIe: 20 / 40 meg..............289 / 339
Mouse for: IIe w/card........89, Laser 128, IIc, c+, Mac+............70
Laser 5.25 Drive for Mac LC, IIe, IIc, IIc+, IIGS, Laser..............99
Apple 3.5 800K Drive for Mac, IIe, IIGS, IIc+, Laser..............199
Apple SuperDisk 3.5 Controller 129
Goldstar Composite Monitor 125
Monitor Stand..............25
RGB 14" color + green + tilt & swivel for Laser..............210
IIe Printer Card w/cable: Parallel…45 Super Serial 62
ImageWriter I Compatible Printer 195

Memory

256K chips..............$18/set
RAM IIGS w/4 meg..............139
RAM Card w/1 meg: IIe..79 IIc..141 c+..156 Laser..101

1-800-726-7086

Intern'l / Local Orders
(407) 790-1070
Fax 790-0108

Perfect Solutions
12657 Coral Breeze Drive
W. Palm Bch, FL 33414

X-10 SOFTWARE IBM • APPLE II

XTEN UTILITIES (Includes XTEN MENUS)

- MENU driven. No programming required.
- Supports 128 events / 256 devices.
- Supports Direct Command, Set Clock Housecode, Upload/Download Events, ALL Lights or Units ON/OFF/DIM.
- Supports Selective/Multiple units/days for Events/Direct Command.
- EDIT events on/off line.
- OSCILLATE offers adjustable repetitions (to 999), ON/OFF/DIM. Great for testing Powerhorns.
- COUNT DOWN TIMER supports appliances and lamps. Offers delays up to 6 Days.
- REPORTS show event/unit useage on screen or printer.

XTEN MENUS

- Fast, Easy, Manual control of X-10 devices via MENUS.
- Replaces House and Unit number with YOUR description.
- Supports Selective/Multiple units/days per menu item.
- Supports Lamps, Appliance, Powerhorn.

XTEN UTILITIES with XTEN MENUS.............. $39.95
Shipping & Handling.............................. $4.50
Massachusetts residents - 5% Sales Tax.............. $2.00
Personal / Bank Check / Money Order (U.S. Funds ONLY)

- Specify IBM or APPLE and specify 5.25 or 3.5 diskette.
- CP-290 X-10 interface required for XTEN UTILITIES & XTEN MENUS.
- IBM PC/XT/AT version requires MS/PCDOS 2.0 (Min), 256K RAM, Text/Hercules/CGA/EGA/VGA video, Serial port, Floppy/Hard Disk.
- APPLE II+, IIe, IIc, IIGS, Laser 128 version requires ProDOS, 64K RAM, 40 col. screen OK / Super Serial card or equiv., Floppy/Hard Disk.

Call or write for catalog

Wilmington Computer Applications
Box 429
Wilmington, MA 01887
(508) 658-9950 9am-Noon (EDT) Mon-Fri

Best Value In Apple II Series Public Domain & Shareware

Largest selection available anywhere, over 1200 BOOTable and MENU-Operated Program Disks, Full of:

Entertainment • Business • Education • Utilities • Print Shop Graphics • Apple Works • Eamone • Telecommunications • Variety • HAM Radio • Novelty • GS. Specific • PRODOS. Specific....and More....!

Programs for Apple II+, IIc, IIc+, IIe, IIGS & Macs with IIe Simulator Card. Most are Available in 5.25 and 3.5 Format. **Low as $1 A Disk • MC and VISA Accepted•** Next-Day Shipping is available **• Special offers for 'new' customers**. Send $2.00 for **Christella's** Descriptive Catalog and Demo Disk..... **Be Pleasantly Surprised.**

Christella Enterprise
VISA
P.O. Box 82205
Rochester, MI 48308-2205
MasterCard

Test Drives

Continued from page 19

regain control of your computer nearly three times faster! Of course, since the printer itself isn't any faster, it still takes the same amount of time for the last page to come out—you can just work in the meantime.

Using Vitesse's *Harmonie* Imagewriter driver with MegaBuff results in approximately "standard" wait times and total print times. However, print quality is much better because the *Harmonie* driver prints at 160 x 144 DPI, double the ImageWriter's standard resolution. The difference is especially noticeable with *Platinum Paint* and other graphics programs. (Programs that print graphics-based text, such as word processors, don't show as much quality improvement.)

What about institutional users? I compared network printing performance in a school's IIGS lab using the AppleShare print spooler, with and without MegaBuff. MegaBuff was significantly faster than the Appleshare v2.0 Print Server, especially with ProDOS 8 programs. GS/OS programs also benefited from having MegaBuffs in the network ImageWriters. Speed improvement varied, ranging from as little as 5% to as much as 100%.

The bottom line: The MegaBuff is a high performance buffer and network interface product. For ImageWriter II owners tired of delays, my advice is simple. Plug in the card and lose some wait! ■

VENDOR INFORMATION

BIG RED COMPUTER CLUB
423 Norfolk Avenue
Norfolk, NE 68701
(402-379-4680)

PROCYON, INC.
PO Box 620334
Littleton, CO 80162-0334
(303-781-3273)

SEQUENTIAL SYSTEMS
1200 Diamond Circle
Lafayette, CO 80026
(800-759-4549)

SEVEN HILLS SOFTWARE
2310 Oxford Road
Tallahassee, FL 32304-3930
(904-575-0566)

ADVERTISER INDEX

Q-MODEMS

THE BEST OF THE INTERNET SHAREWARE DISK $9^{95}

"Your Link To the Internet"

FIND IT ON-LINE!

With a Q-Modem, you can access bulletin boards and computers close to home, across the country, and around the world. You'll find airline schedules, business news, free software, gardening tips, technical help, plus everything in-between! Name any topic—you can find it on-line.

IT'S COMPATIBLE!

The Q-Modem is Hayes compatible and works with virtually any computer. It is compatible with industry-standard, intelligent "AT" commands and all commonly used protocols (including Bell 103/212A, CCIT v.22, and CITT v.22bis). Since the Q-Modem operates asynchronously at 300, 600, 1200, and 2400 baud, you can communicate at whatever rate you need to. All these features let you use the Q-Modem to connect with most modems being used today.

IT'S EASY TO USE!

The Q-Modem is very easy to use. It features autoanswer and autodial (tone or pulse) for performing phone tasks, and it has a programmable-volume speaker for monitoring call progress. Storing phone numbers and user configurations is also simple because of the modem's programmable nonvolatile memory.

You also get:

- Compatibility with U.S. and international protocols
- Nonvolatile memory to store your configurations and frequently called numbers
- Auto-Answer AutoDial (tone or pulse)
- Two phone jacks for telephone line and phone
- Adjustable-volume speaker
- Introductory packages for on-line services
- One year warranty

All you'll need to use a Q-Modem is your computer, software, and a cable to connect the modem to your computer.

MODEM BUNDLES

Q-Modem, MicroPhone LT, and cable
MicroPhone Personal LT is the perfect low-cost solution for Mac telecommunications. It offers fast and reliable data transfer between Macs and other computers...............**$99.95**

Q-Modem, ProTERM 3.1, and cable
ProTERM's intuitive operation, on-line help, test editor, Scrollback buffer and Autolearn macros..**$139.95**

Q-Modem and ProComm Plus
ProComm Plus 2.0 is a popular, powerful, easy-to-use communications software package. It includes 33 popular terminal emulations with keyboard remapping.......**$139.95**

Q-MODEM

$74^{95}

Q FAX MODEM V.32 BIS

The Q FAX Modem V.32bis features 14,400 bps fax and data communications with up to 57,600 bps data throughput using V.42bis data compression. You also get the special advantages of Silent Answer™, optional Caller ID, and the unique alpha-numeric display panel that shows up to 26 different status messages. Comes with Fax STF, MicroPhone LT, and handshake cable.

$249^{95}

"Going from my old 1200 Modem to the Q-Modem 14,400 baud has saved me more time & money than any other computer peripheral I've purchased."

—Kevin Weiss—

Quality Computers™
20200 Nine Mile Rd.
St. Clair Shores, MI 48080

1•800-777-3642

TRIPLE YOUR SPEED, PRODUCTIVITY, AND EFFICIENCY

with powerful TimeOut programs for AppleWorks!

TIMEOUT WRITER'S SUPER PACK

Includes TimeOut Grammar, TimeOut Thesaurus and TimeOut SuperFonts

only $89.95

TIMEOUT DATA SUPER PACK

Includes TimeOut Graph, TimeOut ReportWriter and TimeOut SideSpread

only $94.95

TIMEOUT TOOLS SUPER PACK

Includes TimeOut UltraMacros, TimeOut DeskTools IV and TimeOut ShrinkIt Plus

only $89.95

TIMEOUT GRAMMAR $49.95
Make your writing stronger, clearer and more concise!
TimeOut Grammar detects grammatical errors, highlights the offending phrase, explains the error, and suggests a better way to say the same thing. You have the option to accept the suggestion, ignore it, or rewrite the phrase in a different way. It's you're writing, only better!

TIMEOUT DESKTOOLS IV $29.95
The essential tools for any AppleWorks user!
Includes Calculator, Scientific Calculator, Print Preview, File Search, Directree (steps through a visual map of pathnames and files), Measurement Converter (great for U.S. to metric conversion and vice versa), Print Manager, Calendar, an updated area code listing and Stop Watches.

TIMEOUT SHRINKIT PLUS $29.95
Never complain about disk space again!
If you're like most AppleWorks users, you fight the never ending battle for more disk space. TimeOut ShrinkIt Plus is a file compression utility that works inside of AppleWorks. Includes ShrinkIt and ShrinkIt GS.

TIMEOUT SUPERFONTS $49.95
Add style to your documents!
Add fancy fonts and graphics to your AppleWorks documents. Works with all the same formatting commands you normally use. Also includes a paint program.

TIMEOUT THESAURUS $32.95
Enhance your writing and improve your vocabulary!
A powerful, intelligent thesaurus with more than 5,000 root words containing over 45,000 synonyms!

TIMEOUT SIDESPREAD $32.95
Never complain about disk space again!
Print your spreadsheet files sideways, and gain an unbroken vista of your spreadsheet data. Prints in two different fonts and a wide range of sizes.

TIMEOUT SUPERFORMS $42.95
Design you own forms inside AppleWorks!
SuperForms is a utility that works with SuperFonts and lets you graphically design your own form right on the screen, so you know exactly how it will look when printed.

TIMEOUT ULTRAMACROS 4.3 .. $39.95
Automate your AppleWorks!
A great time-saving device! UltraMacros, allows you to prerecord series of strokes and activate those same strokes with the touch of a key. Save up to a total of 180 macros with over 4,000 total keystrokes.

TIMEOUT REPORTWRITER $48.95
Place data and text anywhere you'd like!
Combine information from several files into a print-ready report, or create a new database or spreadsheet.

TIMEOUT GRAPH $52.95
Create beautiful graphs quickly and easily!
Organize your spreadsheet and database figures into clear, informative graphs, with titles, subtitles, legends, borders, and more.

TIMEOUT OUTLINER $42.95
Organize anything and everything!
Organize information up to six levels deep. Create to-do lists and tests, prepare curriculum or generate lesson plans. Integrates seamlessly with AppleWorks.

Quality Computers™ **1-800-777-3642**

20200 Nine Mile Rd. • St. Clair Shores, MI 48080
810.774.7200 • Fax 810.774.2698

Adventuring Survey

by Jeff Hurlburt

Here's a chance to test your knowledge of Apple II adventuring. Match the item, person, or place in the left column with the correct adventure in the right column. (Each answer is used only once.) Score one point for each correct response and use the chart to obtain your rating.

__ 1. Bracelet of Turani	A. Dungeon Master
__ 2. Silver Snake	B. Ultima I
__ 3. Forcecage	C. Rings of Zilfin
__ 4. Jungle Skyway	D. Jinxter
__ 5. Bones of Dragon	E. Questron
__ 6. Namtar	F. Gate
__ 7. Goldilocks the Thief	G. Dragon Wars
__ 8. Electric Moths	H. Tass Times in Tone Town
__ 9. Zhentil Keep	I. Brimstone
__ 10. Rainbow-shell Turtle	J. Beyond Zork
__ 11. The Warlord	K. Bard's Tale
__ 12. He, Ho, Hum, Eyoh	L. Tangled Tales
__ 13. Gold Dragon Eggs	M. Ultima III
__ 14. Egnad the Robot	N. Magic Candle
__ 15. Duck of Sparks	O. Realms of Darkness
__ 16. Mem Santi	P. Windwalker
__ 17. Trained Giant Eagle	Q. Enchanter
__ 18. Spell of Eternal Winter	R. King's Quest
__ 19. Green Knight	S. Champions of Krynn
__ 20. Book of Sabano	T. Tracer Sanction
__ 21. LUX-23A	U. Caverns of Freitag
__ 22. FireMaster	V. Space Rogue
__ 23. Franklin Snarf	W. Keef the Thief
__ 24. Graham	X. Pool of Radiance
__ 25. Inner Sanctum	Y. Might & Magic
__ 26. Bogum the Dragon	Z. Wizardry V

Scoring

Score	Rating
0-1	Monster Fodder
2-5	Squire
6-10	Knight Explorer
11-15	Knight Quester
16-20	Adventure Lord
21-25	Quest Master
26	Archon

INSIDE THIS ISSUE:

The Apple II and the Internet

Travelling Salesmen and Lawn Sprinklers

The Apple II and Autism

ProDOS Building Blocks: Forked Files

ChronOS: History of the Apple II Operating System

REVIEWS: Spectrum, GNO/ME, Ancient Glory & MegaBuff

APPLEWORKS: Mastering Apple-Works 4

INTERVIEW: Jawaid Bazyar

HEAD OF THE CLASS: Growing an Apple Orchard

MEDIA A LA MODE: Standing on the Shoulders of Giants AND Making Your Own MOD Files

PLUS MUCH MORE!

Answers: 1-D 2-M 3-A 4-J 5-O 6-G 7-L 8-U 9-X 10-Q 11-P 12-B 13-S 14-T 15-Z 16-W 17-E 18-K 19-I 20-N 21-V 22-F 23-H 24-R 25-Y 26-C