ST.MAC

Say Good-bye
To the Good Old Days...

Macintosh
Moves In

Reviewing Lisa's
Office System

Industry Flips
Over Mac

First Look at Microsoft Basic

Students
Protest
Lack of Macs

Finder Authors Bruce Horn &
Steve Capps Interviewed

Meet Your New Business Team.
Macintosh & Davong.

Apple's new Macintosh™ is the most exciting development in personal computing since the original Apples started it all.

Teamed with Davong peripherals, the incredibly easy to use Macintosh becomes a powerful, flexible and expandable business tool.

Available at more than 1,000 Davong dealers nationwide is the new Davong Mac Disk system, providing from 5 to 32 megabytes of hard disk storage. Mac Disk gives you the increased capacity and enhanced functionality you need for your business applications. All at a very reasonable price and with no sacrifice in the speed and ease of use built into your Macintosh.

And coming very soon from Davong are a 28 megabyte streaming tape backup system, for fast and secure backup of your files, and a multi-tasking **local area networking system** to allow all of the Macintoshes in your organization to share information, programs, and the advantages of Mac Disk.

See your local Davong Dealer today for details.

Davong Systems, Inc.
217 Humboldt Court
Sunnyvale, CA 94089
(408) 734-4900
Telex: 176386

DAVONG

ST.MAC

In next month's ST.MAC

Macs on campus...getting creative with Microsoft's *Chart*...new Lisa software... Mac and Lisa user tips...who's buying Macs...and more.

VIEWPOINT
What Hath Cal Poly Wrought?

"We hold these truths to be self-evident: life, liberty, and the pursuit of happiness." Thomas Jefferson wrote it. The students of California Polytechnic at San Luis Obispo seek to amend it. They'd like to add ownership of a Macintosh to the list.

They feel so strongly about it that nearly one hundred of them bearded Cal Poly's president in his office to demand an explanation of why their school wasn't included in the Apple University Consortium.

The scenario lacks all the elements of good drama, or even of fidelity to history. There were no Mario Savios leading a Free Speech movement. There were no S. I. Hayakawas to defend the college administration from the antiwar protestors. All in all, it was a pretty tame affair, under the control of the college bureaucracy from beginning to end.

But that it happened at all says a lot about technology, microcomputers, and Macintosh and their place in today's world.

The nexus of the event was Apple's offer to leading institutions of higher learning to join an educational consortium dedicated to the development of educational software that would run on the Macintosh. *ST.Mac* will illumine that proposal in depth next month; here it needs only to be understood that the quid pro quo was generally as follows: the schools would be preferred customers of the Macintosh and would in turn pledge to sell a predetermined number of machines over the life of the agreement and would undertake to develop curriculum and research software on the Mac for sharing among consortium members.

All the members of the Ivy League, that bastion of academic excellence, leaped at the chance to corral the state-of-the-art Mac in quantity at deflationary prices. The list of enrollees for the AUC reads like an honor roll of scholastic achievement: Chicago, Stanford, Carnegie-Mellon, Brown, Yale, Harvard . . . . It seemed as if all the greats and near-greats were jumping on the bandwagon.

But not Cal Poly at San Luis Obispo. College president William Baker declined to sign the AUC agreement. It seems clear that president Baker stood above the fracas while the AUC idea was getting mired down in bureaucratic infighting.

That's not uncommon on any college campus. Politics, in the pejorative sense of the word, is probably even more prevalent in academic circles than in government. Some of the best ideas and many of the most talented educators go unrecognized in a milieu where petty power too often takes precedence over life, liberty, and the pursuit of truth.

For all the shortcomings of current college administrations, you don't see or hear of students banding together in protest like they did in "the good old days." If the students are marching at UCLA, it's because the Bruins are in danger of missing the NCAA basketball playoffs. If the students at Ohio State are marching, it's because they're members of the justifiably famed Ohio State Marching Band. If the students at the University of Florida are marching, it's probably because there's a shortage of suntan lotion and beer.

Of course, college administrations are no worse nor any better than other institutions. It's quite a sight to see twelve-figure deficits from a conservative president who got elected on his pledge to balance the budget. It's just as bemusing to witness his liberal opposition complaining bitterly about his lack of spending proclivities. Yet there's no general unrest among the body politic.

The point is that there's no peer pressure nor any adult role models that can be seen as motivating the students at Cal Poly. They just got angry among themselves. And there weren't even cosmic issues, such as pursuit of happiness, at stake.

Or were there?

Has our society changed so much that access to technological advancement in fact is a new concomitant of the pursuit of happiness? Is possession of the latest technology the handmaiden of liberty? Is the quality of life dependent upon the quality of the technology that supports it?

If the answer to any of these questions is yes, then the students at Cal Poly are the advance guard of a new wave founded upon solid Jeffersonian principles. And president Baker and his ilk are modern King Georges.

It's interesting to note that all the hubbub centers around the Macintosh. It's unlikely that the Commodore 64 or the IBM Personal Computer could have generated the same emotions.

The issue here is state-of-the-art. The issue wasn't just any old computer, but Macintosh. Apple's been so busy patting itself on the back over Macintosh that disinterested observers feared the company might sprain both figurative arms in an orgy of self-congratulatory contortionism.

But Apple seems to have hit upon something. Macintosh has already inspired a challenge to establishment power at Cal Poly.

Maybe the country really is ready for a different kind of Mac attack.—A.T.

Microsoft's greatest hits are now playing on Macintosh.™
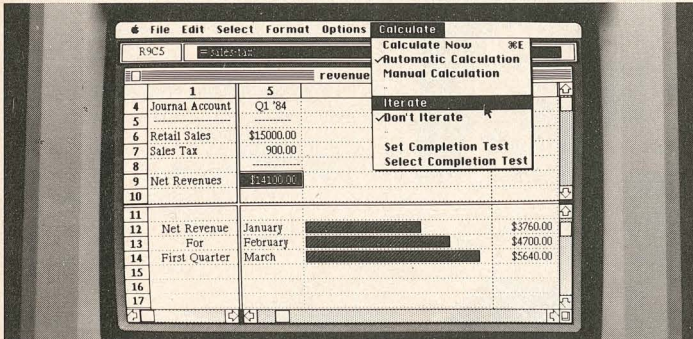
MULTIPLAN.® The most powerful spreadsheet on the market.
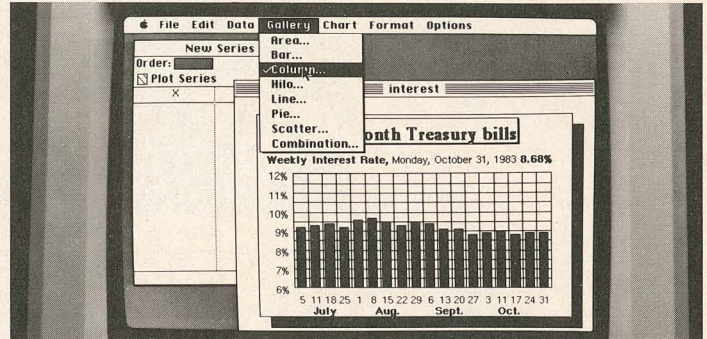
CHART. The first truly sophisticated and flexible graphics system for personal computers.
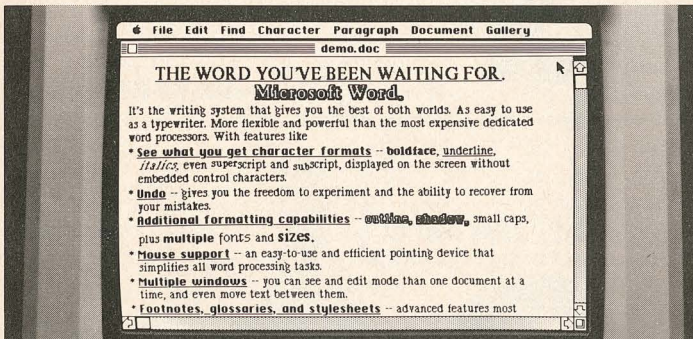
WORD. Most dedicated word processors can't touch it.
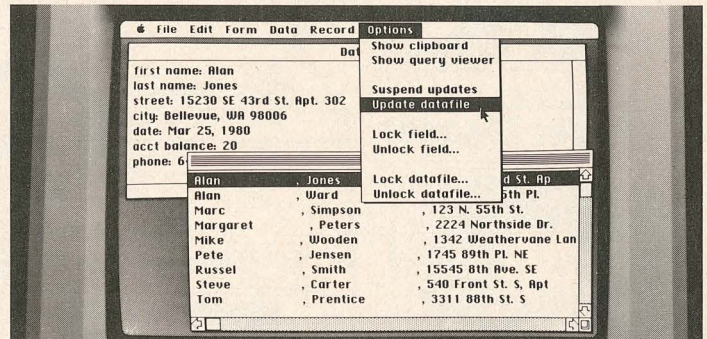
FILE. Finally, a data manager advanced enough to be simple.
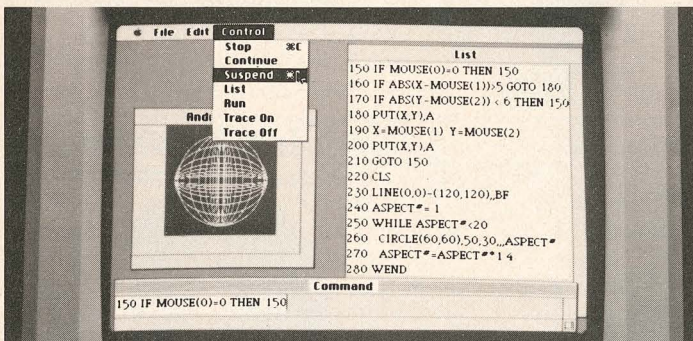
MICROSOFT® BASIC. The industry standard. Plus special commands for the mouse and bit-mapped graphics.

Microsoft was founded on the idea that computers should be simple, easy and intuitive. The new Macintosh environment gives us a great opportunity to prove that idea.

We'll start with four of our high performance programs to make the most of your Mac. And there are more to come. Lots more.

Pioneering like this is nothing new for us. We wrote the first microcomputer BASIC in 1975 and developed the operating system for the IBM® PC.

If you'd like to make the most of your Mac, call 800-426-9400 (in Washington state, call 206-828-8088) for a free brochure and the name of your nearest Microsoft dealer.

MICROSOFT

Andrew Fluegelman

Mike Berlyn

Gary Carlston

Steve Wozniak

Cary Lu

Bill Budge

Alan Boyd

Tom Meadows

Bruce Artwick

Marty Alpert

Jim Sadlier

Richard Garriott

Dave Needle

Dave Jorgensen

Roger Wagner

Dave Albert

Jeff Harbers

Tom Snyder

Dave Gordon

Trip Hawkins

First Impressions

The Mac has arrived. With a vengeance.

Everyone, it seems, has seen it or heard about it. And, predictably, everyone's got something to say. Some people like the machine, some don't. Others simply have their doubts. Who's going to buy it? How will it fare against the competition? (Just what *is* the competition?) And what about the rest of the Apple family?

Is the mouse for everybody? Do we really work better in windows? And where's the color, the slots, the games? The answers won't come easy, and will come only in time.

The curiosity and excitement that surround the Mac are contagious, and no one is more thrilled—or more skeptical—than the experienced leaders in the microcomputer field. What follows are their first impressions of Macintosh.

Bruce Webster, Vice President, Oasis Systems

My initial reaction when I saw Macintosh was slight disappointment—I was looking forward to a hacker's machine. But after spending an hour with it, I was impressed. It's a very slick machine—very revolutionary. There's been a lot of criticism about the lack of color graphics on the Mac, but after having had a month to think about it, I don't really care. I think that color graphics, outside of games, isn't all that necessary, especially given the high degree of resolution on the Mac and the ability to dump it cleanly on something like the Imagewriter.

I think that Mac is the first *personal* computer. It's the first computer that's cheap enough and easy enough to use that the average person can sit down and do something useful. And I speak as someone who spends at least two or three hours on the phone every day handling calls from people who are not calling because of bugs in our programs, but because they don't understand enough about the systems they're using to deal with problems.

The II will continue to be an entertainment machine and a hacker's machine. It's really a chameleon; you can make it look like just about anything. Indeed, Apple seems to be making a concerted push to make the II look like a Mac. I already have the new Apple II mouse and the Apple character ROM with graphic icons in it. I think the Mac group is determined not to repeat the error of the Lisa as far as the lack of third-party software's concerned.

Mitch Kapor, President and Chairman of the Board, Lotus Development Corporation

You can respect an IBM PC, but I think you can love a Macintosh. The Mac is going to be much less burdensome to the average user. We didn't just translate *1-2-3* a feature at a time to the Macintosh; we rethought the whole thing

through. The functionality of the product will not be compromised; it's going to require a 512K machine. I don't think Macintosh is going to compete effectively in the big corporate markets against IBM, at least not for a while. But if people get hooked on Mac, they're going to find it hard to go back. I don't think we'll see any Mac-alikes in 1984. It's significantly harder to reverse-engineer a Mac-compatible than an IBM PC-compatible; so, while people may want to do it, it'll be relatively difficult to do. Hardware people can be very clever, but of course Apple was extremely clever. There may be some *bad* look-alikes.

Dave Albert, Marketing Director, Penguin Software

I think the Mac's neat. I was a little disappointed in the price. I would love to have seen it released at a lower price, although I don't think it's overpriced by any stretch of the imagination. And Apple is doing a smart job of targeting its market.

I've been trying to get my brother, who's a filmmaker, to buy a computer for two years, but now he'll buy a Macintosh. He wants to be able to use it right away, and not take three weeks to learn to use it.

Macintosh may cut out some of the hobbyists that formed the ground floor of the Apple II. But I don't think that's who the machine was intended for, and they'll probably get some treats from Apple down the line.

I don't think the mouse is as good as a tablet for graphics, but it's far better than joysticks or paddles. It's better than any of the other similarly priced alternatives, and I like the approach of putting a single button on it. Something like a light pen isn't sensitive enough for the kind of graphics we do. It's going to be interesting to find out whether or not the mouse will be good for something like tracing.

Cary Lu, Microcomputing Editor, *High Technology*

Macintosh is the most important machine to come out since the IBM PC, without a doubt. It's the first machine that is fundamentally designed to work the way people already work, rather than have people adapt their style of working to the computer.

I'm convinced that in five years' time, regardless of who the leading microcomputer or software companies are, the predominant machines will look much more like a Macintosh than like the IBM PC or the Apple II. It is so much easier to use. You can sit down and work with the machine without having to read manuals.

Plus the Mac takes a tremendous load of designing the interface off programmers' hands so they can concentrate on adding features to a program and making them work more elegantly. I think programmers' productivity will improve with the Mac. People who work for

the more conservative companies will buy the more conservative computers. They don't buy Ferraris or Porsches, they buy Oldsmobiles. The fact that the Mac is actually the better machine is something they don't understand.

One key thing that the Macintosh has done is that it's restored to us five hundred years of development of typefaces. Other computers give you a single face. They're basically typewriters on the screen. They throw away five hundred years of printing heritage that we're accustomed to. Macintosh is the first popularly priced computer that brings it back to us. A lot of people will say they don't need those fonts. That's just because they haven't had the tools before.

Jim Sadlier, Chief Executive Officer, Continental Software Division of Arrays Incorporated

We have a lot of programmers working for us, and for the most part they're pretty jaded when a new box comes out. We've had the Mac now for a number of months and to a man they are all clamoring to buy their personal Macs to take home. It's the first time I've seen such a unanimous opinion on any piece of hardware.

The function of any computer is to deal with a given amount of information in the shortest amount of time possible. Based on that, if I were a small businessman, I would choose the Mac.

In most cases I would say that it's disastrous to bring out a machine without an existing large software base. In the case of the Macintosh, the machine has such potential that I think it can overcome that weakness and hang in for the few months it's going to take to get enough programs on it.

Bob Christiansen, Executive Vice President, Quality Software

We have a kind of wait-and-see attitude about the Macintosh. Personally, I'm nervous about the lack of color in the machine. There are some who feel that the resolution on the Macintosh is so great that you really *don't* need color, and they swallow that whole line. I'm not ready to swallow it. In my opinion, most games lose almost all of their appeal if you remove the color. The number two problem on my list is that Mac doesn't have slots. I hear rumblings about how, well, it's got a serial bus and it's an open computer and you'll be able to do things, but that remains to be seen. As far as I'm concerned, that's one of the reasons why we're taking a kind of wait-and-see attitude. I think Macintosh has a good shot at being a really good computer.

I'm not as enthusiastic about the mouse as Apple has become. I think it may be a pretty nice remote input device, but I hope it doesn't replace the keyboard. The mouse has some advantages over the keyboard, but I think those

COMPILED BY TOMMY GEAR

advantages have been exaggerated in some of the Macintosh publicity.

Steve Wozniak, Inventor and Cofounder, Apple Computer

Macintosh is an art form, and it really has to be seen and used before you know that. Anyone who sees it pretty much knows what computers are all going to be like by the end of the decade. Of course, enhancements for color will be expected. Macintosh is sort of a step above being a *personal* computer. It is more of a *complete* machine. We know what we want to do. Macintosh is for serious users—maybe taking computer science in school or using their first small business computer—but it's not just *business*. It's sort of a combination of personal and business, so I consider it more *serious*. There's some sort of serious computer category that hasn't been well defined yet.

Five years ago, you could say of any computer in the world ever dreamed of that only a technical expert would really ever handle it. Now we are trying to think of computers for nontechnical people who are afraid of calculators.

When we first began Apple, the marketing end—Steve Jobs and Mike Markkula—was basically oriented toward this attitude. "This is the first appliance computer ever," were their big words around the Apple II. So the word has been strongly on their minds, but it's one of those marketing/advertising-type buzz words. Macintosh is actually too good of a serious tool to use a word like that.

I think Macintosh would be the excellent personal computer king if it had color and slots. Those are the two big things that are missing, and even IBM followed the Apple II in those areas. So that limits some of Macintosh's uses. Macintosh is not intended to be a take-it-home / play-with-the-kids / have-fun-with-color computer. It's got some of the best graphics built in for what it is. Anytime you are really trying to accomplish something, the end product is what's really important. Color is an intriguing, artistic highlight. It's a key part of the Apple II market, but it won't be a major part of the Macintosh market initially.

Paul Freiberger, West Coast Editor, *Popular Computing*

Whether the Mac will be as big a hit as Apple, or Jobs, dreamed of I don't know, but I think it's a really impressive machine. People in small businesses will use it. It might be harder to get the machine used in big corporations initially. The Mac looks like a great buy, even if $2,495 is higher than what Apple originally planned to charge. It's a repackaged Lisa. The problem with Lisa was the cost, and they've cut that considerably. I think Apple will sell more Lisas now that the new Lisa looks like an upgraded Mac. The Mac is such an attractive package that if you like it and want a little more memory or a hard disk right away you might decide to buy a Lisa simply as an upgraded Mac.
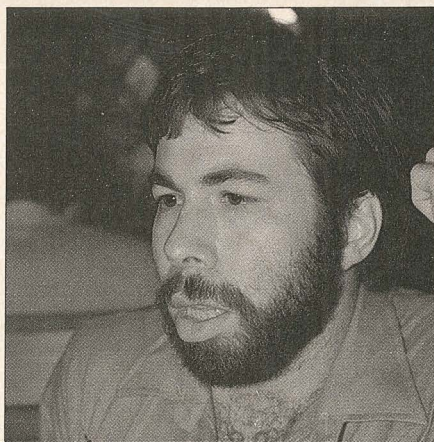
Steve Wozniak:
"Macintosh is an art form, and it really has to be seen and used before you know that. Anyone who sees it pretty much knows what all computers are going to be like by the end of the decade."

Ken Landis, Investments Columnist, *Softalk*

The business viability of the Macintosh is unquestionably going to be determined by the software that becomes available for it. The Mac will open up a new user community in the business world: people with the need for processing power but not technically oriented at all. I don't think it will sell to people who are presently using computers. The reason I think that is because, although the Mac is very powerful, its power has been directed toward noncomputer-literate people. Apple II and IBM users will be intrigued by the Mac but I don't think they'll purchase it, as they're already beyond the stage when many of the Mac's features would be of value to them. Its success will be determined by users, not businesses.

Because of Mac's portability, I don't know how well a second disk drive will sell. I think the portability is a key feature of the machine. Software developers are going to have to take advantage of that extra RAM, using parts of it so programs will load into it completely. If they don't, we're going to have the same problem with swapping disks as on the Apple II. Also, people are going to wonder why they should buy the single-sided drive when the double-sided ones may be just around the corner.

Human beings are not by nature multitasking. When a machine begins to stretch the capacity of the user, it's no longer a friendly or functional product. If you begin opening windows on a product with two or three applications running in different windows, it gets to the point where I begin to question what that's really good for.

Don Burtis, President, Rana Research

Mac's a good machine and has a lot of features for the price, a very good ratio of price

to performance.

The thing that made the Apple II so successful was not just the slots, but the fact that it was an open machine. The indications are that Mac will be open as well. The lack of slots may be a very good thing for the consumer because there will be fewer possible configurations to worry about. With an Apple II everybody unwittingly becomes a system engineer.

Rana will have some peripherals for Mac, and they will probably be along the same lines as peripherals we have already announced for other computers. The peripherals that really matter most to people will be available; it's just a matter of how cleverly those peripherals take advantage of the Mac as opposed to being some sloppy add-ons.

Robert Hardy, Programmer, Penguin Software

I'm absolutely overwhelmed by it. I've been programming for the Mac in Pascal and I've been told by the people at Apple that Macintosh Pascal runs 80 percent as fast as if you were to code in 68000 assembly language. From the response I've been getting to the applications programs I'm doing, I think it may be even better than that. Obviously the Mac is faster than the Lisa. Just the screen-drawing routines are on the average twice as fast. You get approximately twice the resolution of the Apple II, a really tremendous depth of field, and a good feeling of texture that I think makes up for the lack of color.

Dave Jorgensen, Director of Marketing, Davong Systems

Basically, our hard disk system for Macintosh is very similar to our IBM and Apple IIe product, except it will connect through the RS-422 port on the Mac, and it'll be available in five, ten, fifteen, twenty-one, thirty-two, and forty-megabyte sizes. The unit's a Winchester, about four and a half inches high, fourteen inches wide, twelve inches deep. We're pretty enthusiastic about our response to date. We expect to add nearly two hundred additional dealers to our base of over eleven hundred now, just because of the Macintosh.

It's my understanding that a good portion of the software being developed for the Macintosh performs better in a hard disk environment. I view the Macintosh as being targeted at the upper home or small business, or what I like to call the low-end PC market. Of course, Apple doesn't like to call it that.

In the third quarter of 1984 we'll offer networking and tape and some other products as well. We're going to offer a product in the third quarter that we're currently calling MacLink. It'll allow you to network a number of Macintosh drives and will also provide an option to be able to interface to our IBM network. We will provide a combination box that will include a hard disk, a tape backup if users want one, the MacLink network for the local Mac net, and an option for another card that will interface to our MultiLink and talk to a PC network.

Jeff Harbers, Development Manager for Macintosh Products, Microsoft

We've lived with the Mac for a couple of years now, and it's just gotten better and better. Microsoft is very committed to this machine. One thing Apple and Microsoft have both agreed upon is that we would adopt the same religion as far as the interface is concerned. Everything we've done in our Mac products is exactly to spec, and that spec is a joint effort between Apple and Microsoft.

I'm certain the Mac won't appeal to the Fortune 1000, and I think Apple realizes that too. Those are IBM people. The Mac is definitely going to appeal to smaller businesses, people now using the Apple II in business. As a middle-management computer, the Mac is unsurpassed. Add some communications software so you can talk to the IBM mainframes and PCs, maybe a mail network, and that's much better than putting a terminal on a guy's desk.

Wayne Green, Founder and Publisher, *inCider* Magazine

Four main features on the Mac are not of much interest to me personally. Having tried the Lisa, I am not a window fan. I am not a mouse fan. I am not a black-and-white computer fan. And I am not a fan of computers that speak. Which seem to be four of the main things that Mac does.

I have a feeling that the windows and the mice, which are supposed to make the Mac

so easy to use, are going to be lost on most people because it doesn't take long to get used to a computer. The fact that you make the first two hours easier—I'm not sure how much someone would want to pay for that in the long run. I kind of feel like I'm being looked down upon as some jerk when Apple gives me icons. I understand English all right. I don't have to be shown a wastebasket so I can throw something out.

I suspect the Mac will be primarily of interest to the same people who were buying the Apple II: small business people, hobbyists, and scientifically minded people.

It's strange that Apple pioneered color and then dropped it. That was kind of disappointing. Ever since color was pioneered on micros I've felt that it was an important benefit.

Bruce Artwick, President, SubLogic

I'm not a big advocate of interfaces that use the mouse and windows, that sort of thing. I think it's very easy to learn on the Macintosh, and that's a positive point. But I think that, especially in the area of graphics, Apple is relying too much on patterning and shading; I think they should've used color. Of course, with the kind of stuff I'm into—entertainment programs—color is probably a lot more important than in business applications.

There's a lot you could do with the mouse that isn't currently being done. On *Flight Simulator* for the IBM PC, Microsoft was adamant about my getting some mouse control into it,

just in the past five or six months or so. At first I was kind of skeptical, not being such a big enthusiast for the mouse; but since they were kind of pushing me into doing it, I decided to give it a try. And you can really do some good things with it! The new *Flight Simulator* that's going to be out for the PC*jr* in about eight weeks has mouse control. I think that for entertainment programs the mouse has a lot of potential. Everyone who buys a Macintosh is going to have a mouse, so you might as well make good use of that.

Alan Boyd, Manager of Product Acquisition, Microsoft

I'm real gung-ho on the Mac. It's a new type of computer that opens up a new class of users. All computers to date have been used primarily for productivity increases. I see the Mac as being the first computer that can be used for productivity, but also for creativity. The pure brute horsepower of the machine will obviously be attractive to anyone who knows anything about computers today.

The fact that the Mac uses the concept of the desktop is a lot less intimidating than having to think about video screens and text buffers and things like that. Now it's just like a desktop where you can work as you normally do. These things are more than a trend. It's a stage the industry has reached. For the foreseeable future, things like windows are a good way of thinking about presenting information.

When I first started to use a mouse, I thought it would be difficult to use along with the keyboard for things like word processing. But it's no problem. After using a mouse for any length of time, few people want to give up using it. You can do things much faster with it. It's recognition versus remembering.

Computers are getting closer everyday to the way people actually work. In the past, to use a computer you had to change your work habits. Now the computer and software manufacturers are getting smart. For them to sell more computers they're going to have to get the computers to work more like people, rather than the other way around.

David Eyes, Director of Technical Products, Hayden Software

My first thought when I saw the Macintosh was that, a year or two from now, you'll walk into an office of any large company—by large I mean those doing more than a couple of million dollars in sales—and you'll find Macs littering the desks the way calculators do now. Maybe you'll find one or two Lisas on every sixth desk to handle some of the bigger stuff. As far as Mac's application in the home market's concerned, I'm not sure. Obviously there's a business market; the home market is rather nebulous. It's the *personal* computer market that's the operative term. I think a lot of people will be very attracted to the Macintosh in a much more intuitive way than they are to a conventional computer with a keyboard, line-at-a-time input, and all that. It's obviously going to depend on how quickly the applications get

developed and what kinds of applications there are. I'm convinced we'll see the same kind of software phenomenon as we saw on the II.

Roberta Williams, Creative Director of Product Development, Sierra On-Line

I think the Macintosh is real cute. The resolution is outstanding and you're not going to see better-looking pictures on any other computer. The fact that it's black and white bothers me, but I don't think it's going to affect how I write games for it. To me the screen seems a little small. But then again, it's hard to really say until you see the software that comes out on it. You'd be surprised what people can come up with.

My personal opinion is, I don't think I would buy the Mac. If I were a home user, a person with a family, I'd prefer to buy an Apple II. To me the Mac doesn't seem like a home, consumer-oriented computer. It looks like it would be great for an office. And I hope Apple does well with it; I think it's a fantastic machine. I love the mouse and I think the price is really reasonable. We're [Sierra On-Line] currently developing *Homeword* for the Mac and that whole line of products. *Homefile* and *Homespell* will probably follow soon. I hope I'm wrong, I really do. I hope the Mac gets accepted in the home as well as in the office.

Richard Garriott, Vice President, Origin Systems

I was pleasantly surprised with the power and capabilities and speed of the Macintosh. I was fully expecting to be disappointed with the lack of color. And I still consider it something I would like to see. If Apple comes out with a color board for Macintosh first, then there will at least be a standardized method of doing color.

I find it interesting that it's intended to be a very nonprogramming computer, because programmers can do great things on the Macintosh. The resolution of the graphics, combined with the speed and the higher memory capabilities, is a plus for doing outstanding types of entertainment software, in particular, those with graphics. Arcade games don't really need any more memory than the smaller computers have. But for logic or strategy games that require very complex algorithms, the large quantity of memory, and especially the speed, is very important.

Andrew Singer, Vice President, Think Technologies

I'm a computer scientist, my research area was human factors, and the first time I saw the Mac I was just green with envy. I was jealous of all the people who had created such a fantastically new idea. People say the Macintosh is really just Xerox technology, but I don't believe that's true. The technology base perhaps came from Xerox, but it's not just technology, it's also taste. The Xerox people had terrible taste. The small scale of the Mac forced Apple to simplify and distill the essence of all these wonderful user interface ideas into

Roberta Williams:
"The resolution is outstanding, and you're not going to see better-looking pictures on any other computer. The fact that it's black and white bothers me, but I don't think it's going to affect how I write games."

something that would really have mass appeal. People say it doesn't have PC compatibility. Personally, I say screw that. It's not for the people who have PCs.

Our Pascal for the Mac is radically different from other Pascals on the market on any machine because it's completely interactive. Using our Pascal is just like using Basic: You type in a little program, run it, and it goes. You can make a little change, run it again, and it goes. You can step through the program and watch it execute a line at a time, and you're able to look at the values of variables while you're stepping through. It has a Macintosh-style cut-and-paste editor, and the environment automatically formats your program, inserting bold key words as you go along.

People who have played with Microsoft Basic say that Pascal in this form is probably a little easier to use than Basic.

Andrew Fluegelman, Editor-in-Chief, *PC World* and *Macworld*

I think that the Macintosh offers a real alternative to the IBM PC. It's perhaps the only popular computer that does right now. I don't think Apple is going to run IBM out of business, but I believe they've wisely created a computer that really is different. It will create and define a new market, and that's why it will be successful. The PC defined its market, created its market. That's what a machine has to do to make it now. I think the Macintosh will set its own style. The interesting phenomenon to watch will be the third-party software that gets developed for the machine. That's where the key lies to the success of Macintosh.

I find it a very creative machine. I use it in a different way than I use the PC. I use it more intuitively, more subjectively; it puts me in a totally different mode of thinking. And that's

the key to the style of Macintosh. PCs are still very powerful machines, but the Macintosh offers an alternative for people who own two computers.

Tom Snyder, President, Tom Snyder Productions

Macintosh has captured the imagination of the public, but there's really nothing magical about it technically. The processor gives it extra speed, but it's nothing that can't be done on the IBM PC in hi-res mode. Programs written for the Mac may be different than programs written on other computers because of the spirit of the Macintosh. There is a kind of excitement about the whole approach that gets designers thinking in a different way.

The Mac is absolutely great for the younger age groups as well as for college-age students. The kids have a lot of fun looking at the high resolution—it's intriguing. At the college level it's going to take a real integrated effort from a lot of people for the Mac to make a difference. It takes a long time for appropriate software to emerge. The few pieces of software I've seen on the Mac are so powerful that just leaving it around the classroom would promote lots of interest and experimentation.

Gary Carlston, Chairman of the Board, Broderbund

One of our top programmers, a free-lancer, says, "I don't think you guys realize what the Mac means to the programmer. Here's a machine that we can write for that won't be obsolete in a couple of years and will enable us to do all the things we want to do." Dan Gorlin, who wrote *Choplifter*, has been working on another Apple project for nearly two years now. He has an IBM in his office that he's been stroking lovingly, hoping to get to it when he's through with the other project. Now that he's seen the Mac, he says, it's too bad about the IBM. He plans to work on the Mac next. Suddenly, here is another consumer machine that has enough power to fascinate programmers. It will take years to explore it.

For us the Macintosh is significant because it has enabled us to keep some of our most talented programmers who were frustrated by the Apple. They were tiring of it and felt they didn't need to write for the Apple anymore to keep them in the business.

I think the Apple II will boom because of Mac. My theory is that a lot of Apple II owners are going to trade up to the Mac, which is going to create a large market for used Apple IIs. It's obvious that Apple has pricing strategies that will enable them to drop the price of the II this year and the market for used IIs will push the price way, way down. It seems to me the best one can hope to do on a used II is $300 to $400 by the end of the year. A used Apple is an awfully nice computer to get for that price.

David Durkee, Associate Editor, *Softalk*

I shudder when I hear Apple call Macin-

tosh a business computer. While Macintosh has all the power necessary to be a first-class business machine, calling it a business machine, or an "appliance," sells it short. It has the necessary pizzazz to become anything people want it to be.

Macintosh has an effect on people, hackers and nonhackers alike. Mac somehow acts as a mirror to people's potential aspirations and inspirations. People look at Mac and almost instantly see things they want to do with it. What could possibly be a more exciting art form than something that does that to people? Mac will go places with people. Artists will use it, as well as writers, designers, students, poets, and architects. Given the software—and the software will come because programmers will be among the first to feel the excitement of the machine—Macintosh will find a way to serve anyone whose vocation or avocation stretches the resources of the human mind.

Dave Needle, News Editor, *InfoWorld*

It's fun. I haven't had the need yet to refer to any documentation. The speed is nice. It's a lot faster than working on a IIe. I'm not up on using mice; I think it would take me a while to get comfortable with using the mouse in creating a document. I like to work with my keyboard in my lap; to use the mouse you sort of have to be hunched over, or have "mousepads" for your legs.

Apple's calling Mac the "everyman's" machine, but I don't quite see that. We haven't come up with the "every use" yet, the universal need for a computer. It isn't enough to make a computer easy to use; you have to convince people that they have a pressing need for one. The Macintosh will help sales of the Lisa, which isn't selling well.

I expected to be kind of disappointed by the Mac's lack of color, but when I saw all those shades of gray and the different things you can do with them, it didn't bother me that there wasn't color.

Fred Sirotek, President, Sir-tech Software

The Mac seems like a nifty little machine. We hope to develop *Wizardry* for the Mac. We have managed to do on the Apple II with some degree of difficulty what is standard on the Mac, namely the overlapping of windows and that sort of thing. That is actually running on the third scenario of *Wizardry* on the Apple II. The same style of production would apply to the Macintosh, which is built for just that.

I would not think that the Mac will ruin the Apple II market. It obviously splinters it a little. I think the Apple IIe is here to stay for some time to come.

Mike Berlyn, Senior Project Manager, Infocom

In a sense I see the Mac and the Apple II as being the exact same machine with some distinct differences. I think they will both be very broad-based and have an extremely large base of software. I don't see the lack of color as a problem with the Mac. Given the price range Apple was targeting for, I think the high-resolution black-and-white monitor is just fine.

Expandability is something that has been touted as important, and at one time it was. We're at a point now in hardware and software design where expandability is no longer a problem that needs to be addressed. Making an Apple II expandable to thousands of kilobytes of RAM doesn't do a bit of good considering you can address only 64K at a time. Expanding an IBM PC to 512K when 99 percent of its applications use up 64K of RAM is a joke. A double-sided 3 1/2-inch disk that can hold 800K is more than anyone could reasonably need for a single drive. And 128K that you can address and use is better than 512K in an Apple II where you can't even use it.

Genelle Bedke, Vice President of Marketing and Sales, Software Publishing Company

We feel that the Macintosh opens up another range of possible applications in the industry. One thing about Lisa was that it was priced higher than was practical for small businesses. With the Mac, Lisa Technology has been brought down to a price point for the average individual that's really going to open up that market.

Our basic approach to developing Mac product will be to take the same fundamental concepts we've implemented in our other programs—*PFS:File* and *Report*—and carry them over into the Macintosh environment while preserving the simplicity of operation our programs are known for, avoiding the bells and whistles such as going overboard with the use of icons.

Rich Friedman, Managing Editor, *Popular Computing*

I'm pretty excited about the Mac. There are limitations to the machine, but I think the whole thing rests on what the outside software sellers can do for it. To do any kind of backups, protect your data, or do any kind of copying, sometimes it takes four or five disk swaps. I think the one disk drive is a limitation.

I don't see IBM being very much hurt by the Mac at all. IBM has a base of computers out there approaching three million. I don't see how any machine can compete on the same turf with that. I don't know whether you can give IBM a three-year lead time on this marketplace and expect that the Macintosh will set another standard. I think that's unrealistic. I think the Macintosh will do very well, but I still don't think it will be in the same league saleswise as the IBM.

Marty Alpert, President, Tecmar

We're very excited about the Mac. We think it's a technological advance in the user interface and in the software. The hardware is extremely impressive in terms of functionality, but the user interface is what has made it a technological advance. People like to *know*; they don't like to *learn*. People can use a computer like Mac without really having to learn how to use a computer.

I think the Macintosh user is someone who has recognized the need for the computer, doesn't have the time or the inclination to learn how to use a system, and has limited technical requirements from the computer. We're going to have some expansion systems available the first quarter, like a Winchester interface, a telephone modem interface with the software, and a EEE48 interface. These will all be standalone peripherals that plug into a socket—a port—that Apple has provided, and you can plug in plotters and two types of printers—letter-quality and dot-matrix. You'll be able to do all the expansion on Macintosh that you do on the other systems through the AppleBus.

PALANTIR™ WORD PROCESSING
WE DON'T HAVE TO BEEF UP OUR GUARANTEE WITH A LOT OF BULL

Other software companies give you disclaimers. Palantir gives you a real guarantee: Palantir backs its software with 90 days of free phone support via a toll-free number. When you call, we won't tell you to ask your dealer or read your manual; we'll answer your question, free of charge. If we can't solve your problem, we'll replace your Palantir Software with any competitive software of comparable value.

We couldn't make an offer like this if we weren't confident about Palantir word processing. It's easy to learn, easy to use, easy to live with.

Palantir word-processing software is designed for microcomputers. Yet it gives you all the features of a dedicated word processor. You won't find a better system on the market today.

To find out more about Palantir software, mark the reader-service card in this issue or call, toll-free: 1-800-368-3797. In Texas, call 713-520-8221.

We'll respond with detailed information on Palantir software and a free "No Bull" button. A closer look will convince you that we're not just one of the herd. And that's no *bull*.

Palantir Software   3400 Montrose Blvd.   Suite 718   Houston, Texas 77006
™ Palantir is a trademark of Palantir, Inc.

Mark Ursino, Marketing Manager, Microsoft

So far what I've seen of the acceptance of the Mac has been very, very good. The dealers seem to be very enthusiastic about the machine. Apple is backing it up very strongly, providing third-party software vendors a lot of good support. It looks to me to be a very thorough and well-thought-out program built around a machine that is truly unique in the market. I don't see any way it can really fail. Mac represents yet another quantum jump in microcomputer technology that will open up still other new markets where the micro has not been as well accepted. There will be quite a bit of overlap between the Mac and the IBM PC battling for desktop territory because of the price performance and ease-of-use standards set by Mac. But the Macintosh is really going to be pioneering markets where people wouldn't normally even consider an IBM PC.

Tom Meadows, Senior Vice President, BPI

There are some things about the Mac—notably the thirty-two-bit architecture—that make it a handy machine for accounting purposes. We have seven products that we're developing for Mac: four of our accounting series, *General Accounting*, *Accounts Receivable*, *Accounts Payable*, and *Payroll*. And then our home series, which includes *Personal Accounting*, *Personal Investing*, and *Speed Reading*. All those projects are in various stages of development now.

We think Mac is a fine host for entry-level accounting systems. Once the auxiliary drive becomes available, Mac will be a very popular business computer to use for accounting purposes. The second drive will be absolutely required for all our accounting applications, which require both a program disk and a data disk. Unlike a lot of popular software in which your programs can reside entirely in memory—as is the case with a spreadsheet or a word processor—ours must be on-line at the same time. Until the auxiliary drive is available and widely distributed, we don't really have a market for our accounting products. On the other hand, our home series should work fine on a single-disk system.

Dave Weiner, President, Living Videotext

Initially we started doing the conversion of our software to the Mac following much the same rules we followed on other computers, and it actually went very quickly. Then we started taking a look at what special things Macintosh could do, and started seriously playing around with the mouse. As it turns out, with a program like *ThinkTank*, we found we could really cut through a lot of the overhead, the things people would have to learn in order to use the program. It allows you to create your own command interface much more so than a conventional computer does; it gives more to the software designer in the way of options in creating self-explanatory programs. We feel right now that we've barely scratched the surface as far as Mac's capabilities are concerned.
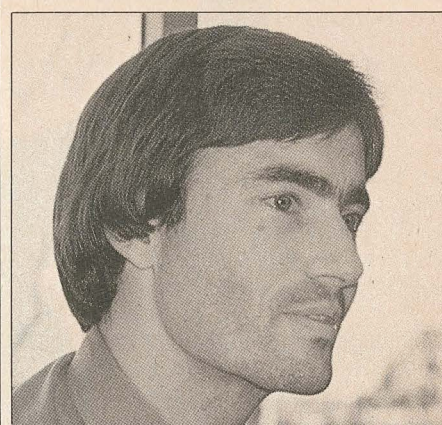
Trip Hawkins:
"The Mac is very much an Apple product and there are both good and bad points about that. The good points are the aesthetics, the simplicity of design, and the value in terms of what you get for the price."

On the Mac, *ThinkTank* is generally a more flexible tool than it is on other machines. Because of the Mac's graphics you can explain things better. The mouse gives you a different channel into the program, separating the processes of commands and text entry. This gives you more power because you're not trying to accomplish everything through the keyboard. The actions with the mouse and keyboard respectively can be given more significance.

We see the Macintosh as almost a restart for the whole industry. A lot of programmers are totally uninterested in working on machines other than the Mac. The university market that Apple has chosen to emphasize is largely untapped, and that market is truly excited about the machine. It seems to us that it's really going to work, that it's really going to hit. Everything about it says *standard*, and anybody that looks at it can't fail to believe that.

Trip Hawkins, President, Electronic Arts

From day one I've been most impressed by the fact that the Macintosh is designed for simple manufacturing. It's basically built like a TV set, and there's a very important distinction between the way TV sets are built and the way most people build computers. Even though it's priced now at $2,495, I think over the next couple of years that this manufacturing technique will allow it to become cheaper and cheaper.

I've always felt that Lisa Technology would be a watershed in the industry. I think Macintosh is probably a better reflection of the price that people are willing to pay for a lot of the ease-of-use features that were introduced on Lisa. I don't think the capabilities of the Mac are really new and different; they're extensions of existing capabilities.

You can't really use a joystick with Mac, and I think that's a real big weakness as far as playing games is concerned. It's going to be really hard to have good multiplayer games, or even two-player games. In terms of one-player games, certain games will be hard to do. For example, something like *Pinball Construction Set* will be wonderful on the Mac, while something like *Archon* will be impossible. That's really unfortunate. The one thing that's really different is the speed of the CPU, so you can do high-speed animation that hasn't been done on the current machines.

Apple always concentrates on reducing costs by stripping out things that aren't absolutely essential. The reason you can't use a joystick on the Mac is because Apple's convinced that the mouse is everything to everybody for every purpose. The interface to the mouse is designed to work only with a mouse.

I think the Mac will be successful, particularly when you consider that IBM came into the market without a very innovative product and showed that marketing and service count for a whole lot. The product advantage of the Apple II has been slipping away from year to year; I think the Mac is going to bring some of that back.

Dave Gordon, President, Datamost

I love the user interface; it's so easy to use. We got the Mac up and running without reading one page of documentation. We're supporting it very heavily—believe it or not, we're going to be doing some games for it. The lack of color is very much a drawback; however, I think the high resolution greatly adds to the machine and one could do some very strong text-adventure-type games on it. I love the mouse, and if there's input required on games I think the mouse is the way to do it.

Mac's ease of use is setting a standard that I don't think anyone should fall below. All the other manufacturers are going to have to make their products as easy—I don't care what the box has inside it. This user interface is so far advanced in this industry that manufacturers will be foolish to buck it. If they don't maintain this simplicity of user interface, they're going to lose their market share.

Peter Olivieri, Business Columnist, *Softalk*

It's hard to make a judgment about a machine without having a lot more software available to play with. It certainly is an easy machine to learn to use. I hate to type. In a lot of the seminars I have given to executives at the managerial level it's unbelievable how often the interface becomes the stumbling block. Although I was a little bit fearful of the mouse at first, it certainly is easy to get used to. I found myself amazed at how little I had to interact with the keyboard. A lot of the cryptic commands one would normally use are no longer necessary.

I like the fact that the machine worries about things that I won't have to worry about. When I put the disk in the drive, it keeps it there and will not let it go out until file man-agement has taken place. I can't accidentally pull out a disk and lose something by mistake.

With 128K you can't do a lot of great business stuff. It just isn't enough yet, but I believe that will change. I think you really do need a second disk. When I started switching a lot of files and saving them with the Mac, it started talking me into doing a lot of disk swapping.

Bill Budge, Software Artist, BudgeCo

I've done graphics stuff on the Mac and it's about six to ten times faster than the Apple II, and it's got about four times as many dots; so overall games should be able to go about two to three times faster. I think it's going to be a great game machine.

I don't think the absence of color will be a drawback. You'll start seeing 3-D games like flight simulators without color that just use shading and different kinds of patterns to get the same effect, and they're going to run a lot faster.

I think the first game I'll write for Mac is a construction set. The Mac is fun to work on for more sophisticated programs. It has a great operating system. The ROM is just packed with good stuff.

For me, actually, it is ideal. It is a standard machine. There are no slots and extra peripherals to worry about or three different types of joysticks, Mockingboards, or this and that. It's got standard built-in sound.

Games will have to change a little bit to take advantage of the mouse. It's going to be hard to figure out how to do *Space Invaders* with the mouse, but maybe someone can think of something better to do.

The Mac is powerful enough that the current arcade games can be programmed by anybody who knows how to program. It is that much faster, and high-level languages work better. That's going to force arcade developers to get more sophisticated.

Roger Wagner, President, Roger Wagner Publishing

One of the big advantages of the Mac is the tremendous amount of onboard ROM. I'm a big fan of calling any routine that exists already in the machine and taking advantage of it. This means that ultimately programmers should be able to do a lot of things by using simple calls that otherwise would be very tedious.

The first time I ever saw a Mac, it was the educational implications that impressed me most. I can see in my mind's eye Macintoshes sitting in libraries and work areas in every school in the country. I don't know what it is; the Mac just has that feel about it.

The three-inch disk alone thrilled me. I can just see shoving one into your back pocket. You can treat them like you would a pencil, and that, in and of itself, is really neat.

I get the feeling that it's just like the calm before a storm. The air is heavy with computers just about ready to be sold. I think you're going to see Macs, and computers in general, in a lot more department stores and people doing all sorts of things with them.

A Guided Tour of MacPaint

BY DAVID DURKEE

It is exceedingly rare for a computer graphics program to be as easy to use as pencil and paper. Typical elements of graphics software, such as the keyboard, paddles, and joysticks, as well as difficult-to-memorize commands and modes, serve mostly to stand between the artist and the high-resolution canvas.

MacPaint, by Bill Atkinson, uses the abilities of the Macintosh to such advantage that the computer becomes easier to use than conventional artist's tools. To call a graphics program *freehand* before *MacPaint* would have been laughable; now there's no other word for the way computer graphics ought to be done.

The tools of *MacPaint* are a lot like those an artist uses: pencil, paintbrushes (with patterns instead of colors), an airbrush, a straight-edge, and a lettering kit. To these *MacPaint* adds some tools an artist might only long for: an eraser that leaves no rubbings behind and doesn't wear through the paper if you use it too much; a button that magically undoes your last mistake; and a magnifying glass that allows you to blow up part of your picture eight times and manipulate it at a dot-by-dot level. In addition, *MacPaint* gives you the ability to design an object in any free space and then move it onto a finished background without messing with paste or risking sloppy borders around the object. It also allows you to duplicate anything in a picture perfectly and effortlessly.

MacPaint makes it all so easy that you almost feel like you're cheating, except that it's so much fun you don't want to give it up. Where most computer graphics programs are so difficult to learn and use that they seem a poor alternative to the "old-fashioned methods," *MacPaint* is, in many ways, easier to use than conventional artist's tools.

Still Life with Mouse

The mouse may be the best way yet devised for giving graphic input to the computer. Other computers have a variety of options, all of which are less than perfect. Joysticks and trackballs are popular for game playing, but the mouse is far more accurate (and it requires only one hand). The light pen and the touch-sensitive screen are acquiring followings of their own, but they require you to hold your arm up to the screen, which gets tiring fast. The mouse allows you to work with your arm in a relaxed position on the desk.

The graphics tablet is well liked by computer artists, but it's terribly bulky (about the size of a record album and at least a half inch thick) and it diverts the attention of the artist from the screen (it also requires two hands). Amazingly, after you've used the mouse for a while, your attention moves from the object in your hand to the pointer on the screen. The mouse becomes an extension of your hand.

Perhaps there will be better tools in the future. One can envision a user interface that combines the best features of the graphics tablet, the touch screen, and the light pen. Say a flat, lap-held monitor, perhaps 8 1/2 by 11 by 3/4 inches, very high resolution, of course, with a light pen and/or touch sensitivity built in. Perhaps that, with the right software, would be the ultimate graphics tool.

Inside Mac Graphics

Without good software, the best graphics user interface, be it mouse, light pen, or graphics tablet, is worth little more than an Etch-A-Sketch to a computer artist. *MacPaint* isn't the only good software involved in creating those neat images on the screen. Inside the Macintosh ROMs is a set of graphics routines called QuickDraw, which, like *MacPaint*, were written by Bill Atkinson. *MacPaint* is essentially a *front end* for QuickDraw. That means that *MacPaint* doesn't actually watch what you do with the mouse and draw something on the screen; it watches what you do with the mouse and *tells QuickDraw* to draw something on the screen.

That distinction may not mean much to someone using *MacPaint*, but it foretells great things for Macintosh. QuickDraw is always in the Mac's memory. Any programmer who knows how to talk to QuickDraw can create effects as impressive (or more) as those seen in *MacPaint*.

On with the Tour

The best way to express the wonders of *MacPaint* is to create a picture of seeming complexity and then show, in a series of simple steps, how it was done. Let's start with the archway in the center of the picture, shown by itself in figure 1. With *MacPaint*, it doesn't matter where you choose to start because it's easy to draw elements of the foreground individually, then draw the background and move objects onto it.

The painting window in *MacPaint* shows about a quarter of the area of the *MacPaint* page at a time. The whole page, when printed out, covers an 8 1/2-by-11 sheet of paper. To make it easier to draw objects separately from the background, you can use the top third of the page as a working area and construct the final picture on the bottom two-thirds.

Because the archway required such fine detail, the front section was drawn entirely with the Fat Bits option from the Goodies menu. We used a somewhat conventional artist's technique to line up the bottoms of the columns: We leaned a triangle against the computer screen (the flatness of the front panel allows this) and used that as a guide.

Once the front section of the archway was created, the rest was easy. We lassoed it, used Copy and Paste from the Edit menu to duplicate it, and then moved the duplicate carefully in front of the original. Then we took that construction and did the same thing to it, which resulted in the four-sectioned archway shown in figure 2. Lassoing an object treats any white space enclosed within that object as solid white, although white space that isn't fully enclosed becomes transparent. Sections of the arches in back that were supposed to be visible through the arches in front had to be redrawn.

Figure 1.

Figure 2.
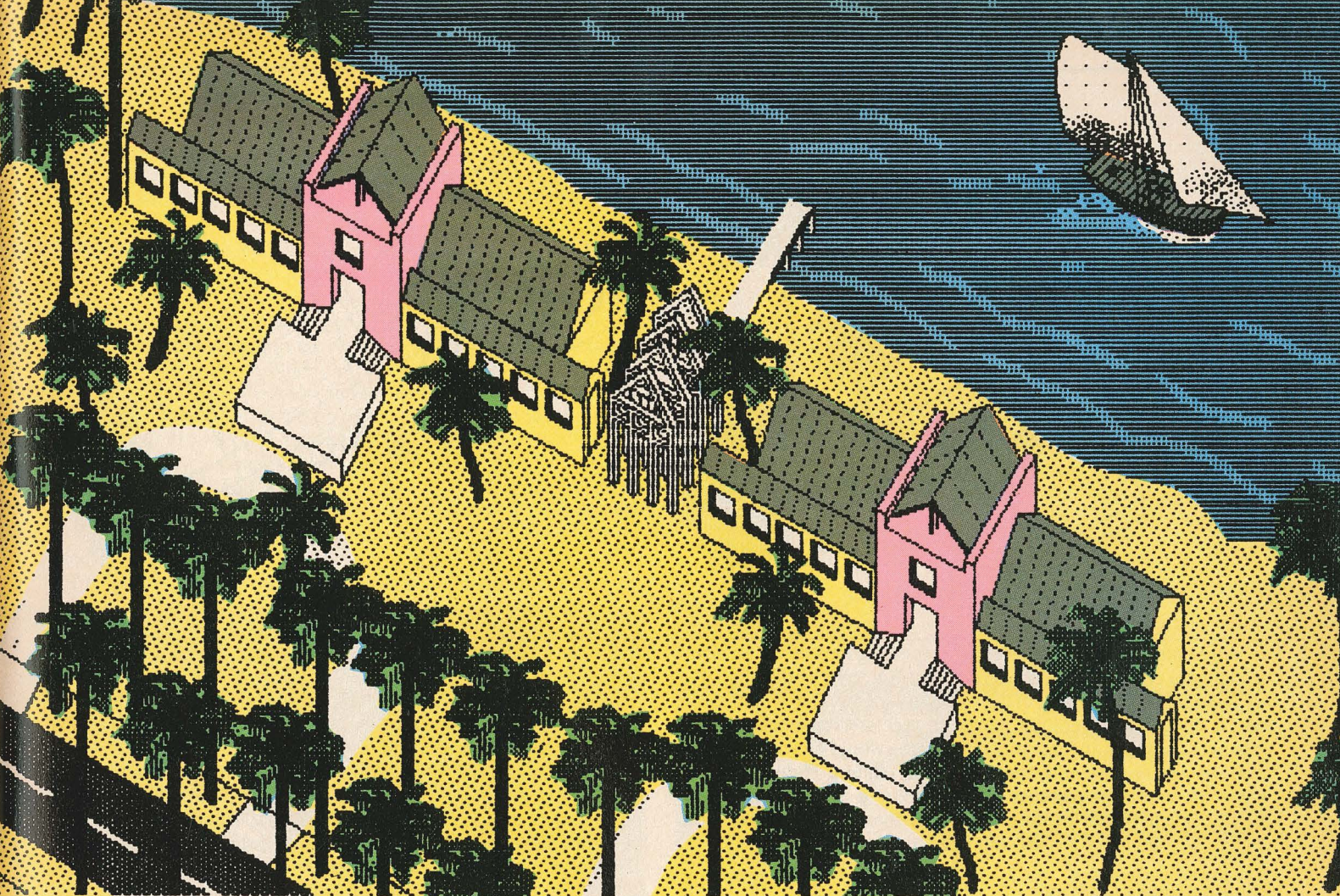
The houses in the picture were inspired by a design of twin houses by architect Fernando Montes. Since the houses were to be identical, we only had to draw one and then copy it. After drawing the outline for part of the house, we created a window (a house window, not a Mac window) in the center section of the house, moved it out into the open as shown in figure 3, and then made ten copies of it. The results can be seen in figure 4.

The roof section of the right-hand wing of the house was drawn in place as well. Then it was carefully lassoed and copied onto a free area of the screen. There it was edited; some lines from the house around it had been copied too and had to be removed with Fat Bits. Figure 5 shows our progress at this point. Then, rather than bringing the roof to the building, we moved the building onto the roof. This saved a lot of extra work because the center section of the building blocked out those parts of the roof that weren't supposed to be visible.

Designing an Environment

At that point the house was mostly complete, lacking only a few details that we set aside for later. Now it was time to create the landscape the houses would rest on. Using the Show Page option from the Goodies menu, we moved the working window to a blank region of the page at the bottom. Then we penciled in the shoreline, still using an artist's triangle to ensure that the houses would be parallel

to the water. We filled in the ocean with one of the darker patterns but left the land for later.

As shown in figure 6, the blank land area was used as a workspace to draw a boat. For all its detail, the boat was fairly simple to draw using the pencil, paintbrush, paint bucket, spray paint, and straight-line tools. When it was done, we painted in some waves, a sailboat wake, and the foam splashing off the side of the boat. Then we dropped the sailboat into place and filled the land area with a light, sandy-looking pattern, as shown in figure 7. As always, the easiest route was to design foreground and background elements separately and then combine them when each was perfect. Only a tool like the lasso could make this possible by allowing you to pick up an object without picking up any of the white space surrounding it.

The area we wanted to use for the final picture was larger than the size of the working *MacPaint* window; so, while we could view the entire page using the Show Page option, we could fill with the paint bucket on only one portion of the screen at a time. To complete the background for the entire picture, we had to move the window around and fill additional areas with the darker pattern used for water and the sandy one used for land. This caused a problem, but not the one we anticipated.

The problem we expected was one of aligning the patterns. Each of *MacPaint*'s thirty-eight patterns,
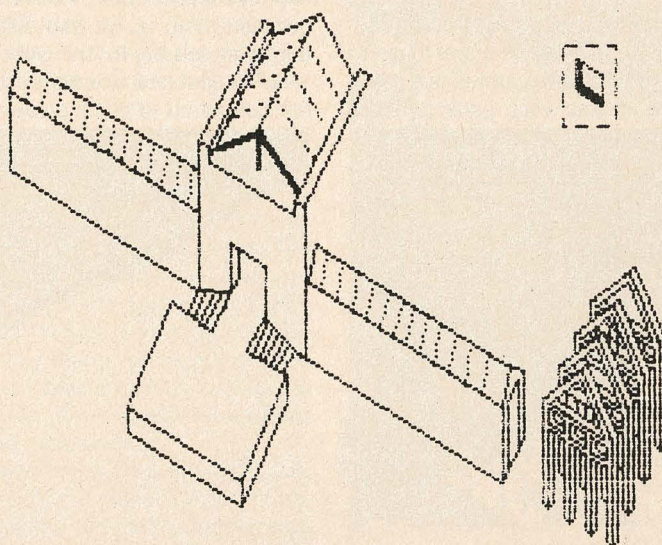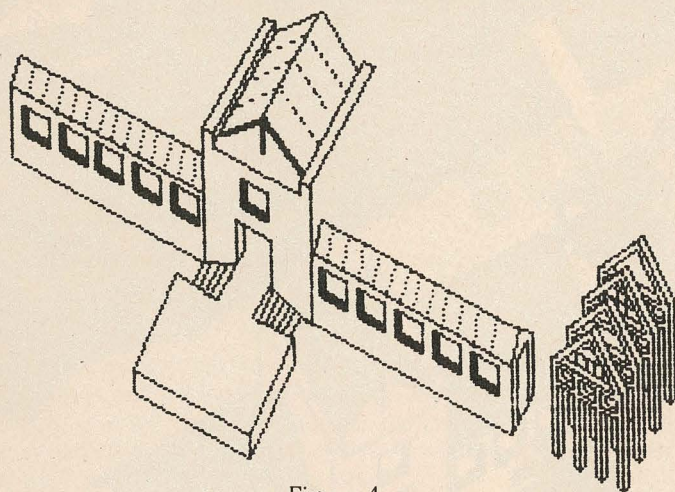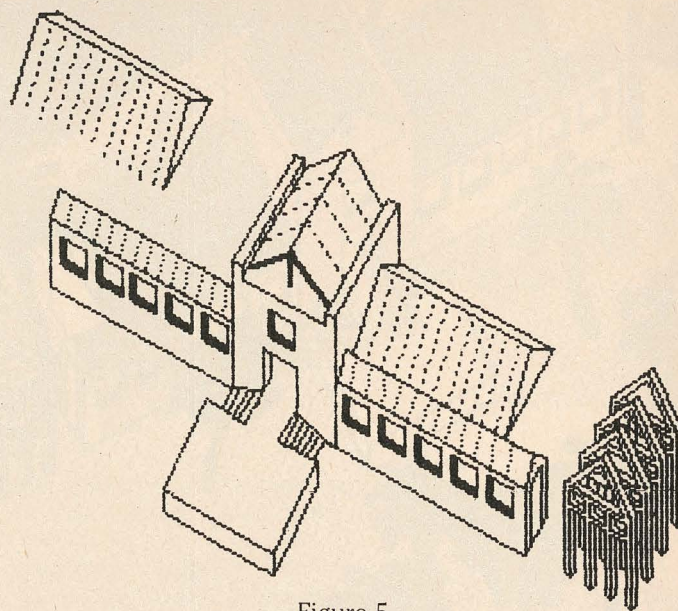

Figure 3.

Figure 4.

Figure 5.

as well as any you design yourself, are repeating mosaics of an eight-by-eight dot grid. We were concerned that *MacPaint* would align that grid with the top left corner of the working window. That would mean that, if the window were not moved in precise increments of eight dots both vertically and horizontally, an area would fill with a pattern that was out of sync with the rest of the pattern. Fortunately, this wasn't the case: The painting patterns seem to align themselves to the upper left corner of the page rather than that of the working screen. We had no problems with filled areas not lining up.

The Boat Takes On Water

Unfortunately, sometimes a pattern can "seep" through an already filled area. We knew that an area to be filled had to be totally enclosed; one dot out of place would allow the paint to leak to other areas. What we didn't know—until we saw it happen—was that a pattern could leak through a filled area and come out on the other side of the screen. When we filled the peripheral areas of what would be our final workspace with the water pattern, it completely erased our waves and colored the sails of our boat.

Fortunately, the Undo feature from the Edit menu saved a lot of time fixing our mistake. Undo can be a life-saver—a magical command to set time back to just before we did the last thing we did. Wouldn't it be nice to have such an option in real life?

Magnifying the pattern we were filling with revealed the problem. The dark gray we were using was composed of parallel black-and-white lines. The white lines provided pipelines for the paint to get to the waves and the sailboat. The waves, which were painted in a much lighter pattern, were obliterated by the darker water pattern, and a

single dot missing from the sails had allowed the boat to take on water.

Having figured out the cause of the problem, we were able to solve it. Were it possible, we could have moved the window in such a way that the vulnerable areas were off-screen at the time we filled the peripheral areas, but that would have required moving the working window partially off the page, which isn't allowed. Another way out might have been to isolate the vulnerable area with a solid line while filling the remaining white space and then paint over the solid line with the fill pattern. We could also have selected a very large brush and painted the area rather than filling it. What we ended up doing was using the selection rectangle tool to copy portions of the existing ocean onto the unfilled area. Sometimes, the hardest thing about *MacPaint* is remembering all the things it's able to do that you wouldn't ordinarily think of. Fortunately, for any given problem there are often two or three possible solutions.

The Arrival of the Prefab Homes

Moving the houses from the top of the screen to the bottom looks like a monumental task, but it was actually only one step more complicated than moving the sailboat onto the water. In that situation, the boat and the place where we were going to put it were all on the same work screen. But the houses weren't on the same screen as their destination. Also, we couldn't grab both houses and the archway between them as we had originally planned because, all together, they were too big for the work screen.

So we grabbed one house with the lasso and selected Copy from the Edit menu to place a copy of it on the Clipboard. Then we used
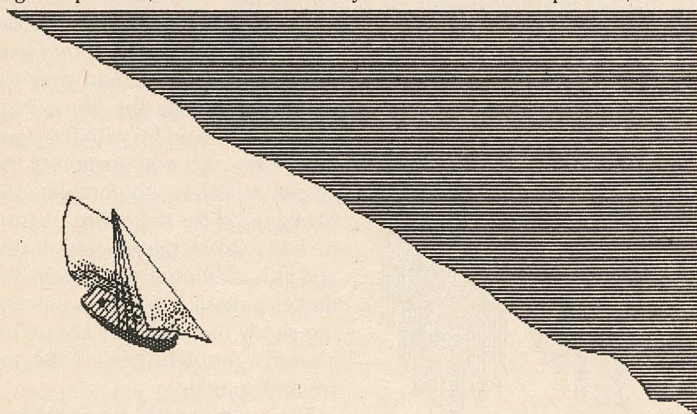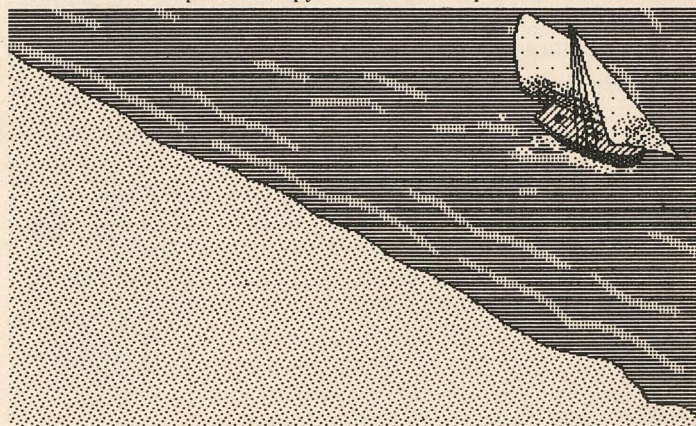

Figure 6.

Figure 7.

the Show Page option to move the window to the desired destination. Finally, we pasted the house down. Since the house was still in the lasso when it came off the Clipboard, we were able to position it where we wanted before actually letting go of it. Then we repeated that procedure with the archway and the second house. The results appear in figure 8.

The picture still looked kind of barren at this point, so we drew in driveways, sidewalks, and a street, and prepared to plant trees. The ocean was so calm that we decided to go for a tropical look—palm trees. They were created, as seen in figure 9, in the working area at the top of the page. This area still contained the original houses, which was useful, as it allowed us to scale the trees correctly. You'll notice that we created only four different trees. The blob on the right-hand side of figure 9 is a cutoff version of the top of the tallest tree, created with the selection rectangle, Copy, and Paste.

We felt pretty familiar with Copy and Paste by this point, so we used the keyboard equivalents, command-C and command-V (holding down the command key, which is marked with a cloverleaflike symbol, and pressing the appropriate letter key), to speed up the landscaping process. We copied one tree at a time from the top of the page, then moved the working window to the bottom of the page
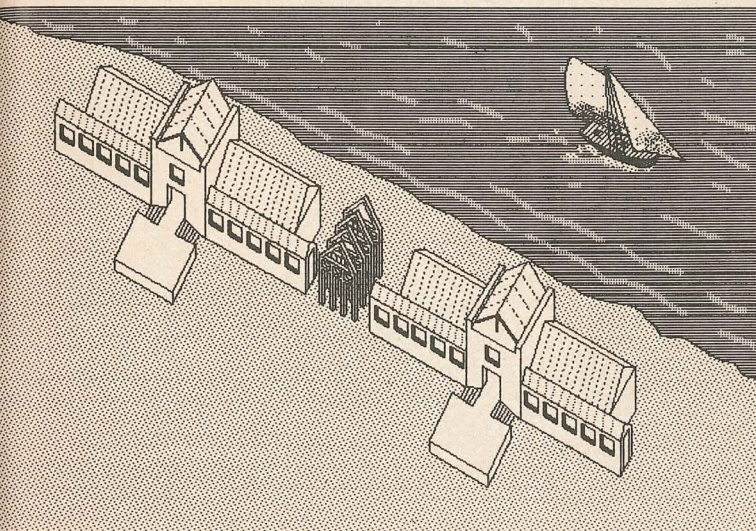
Figure 9.

Figure 8.

Figure 10.

and pasted that tree in as many times as we needed it, repeating the process with each tree.

After creating a dock behind the houses, we touched the picture up a bit with Fat Bits. Figure 10, for instance, shows how Fat Bits can be used to remove extraneous dots from the shoreline. The solid line was necessary for filling in the ocean area, but it made the shoreline look *too* well defined. Compare the original shoreline in figures 7 and 8 with the shoreline in the finished picture, figure 11, to see the difference.

On Being a *MacPaint* Expert

MacPaint has great potential as a graphics tool for use both by artists and by those who find pencil or paint too unforgiving. The secret is in knowing all that it can do and choosing the easiest or most effective method to solve a particular problem. Fortunately, there's a lot of room for experimentation. It's a good idea to save your picture frequently by selecting Save from the File menu. Undo will get you out of a lot of messes, but only if you catch your mistake before you make another change. To go back several steps, select Revert from the File menu to retrieve the last version you saved.

Experimenting and having fun with *MacPaint* is the best way to gain expertise. Try things you've never tried before. Look at the Short Cuts screen, available from Goodies, to figure out the more sophisticated techniques. Stretching items in the selection rectangle, for instance, can provide hours of meaningful amusement; it may also give you new ideas. And that's really what it's all about.

Figure 11.

Lisa Cracks the Books

Reviewing BPI's General Accounting System

BY CAROLE WEAVER

Up to now, there has been a dearth of serious accounting software for the Lisa. BPI Systems of Austin, Texas, has changed all that and created *General Accounting*, a package that small to medium-sized businesses can put to extensive use.
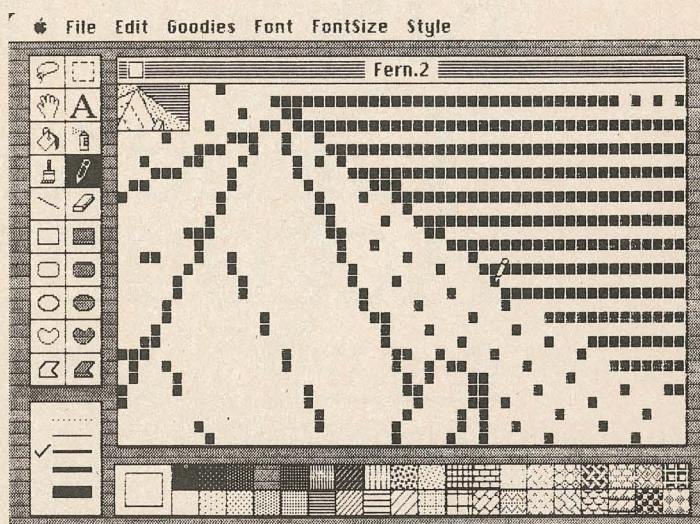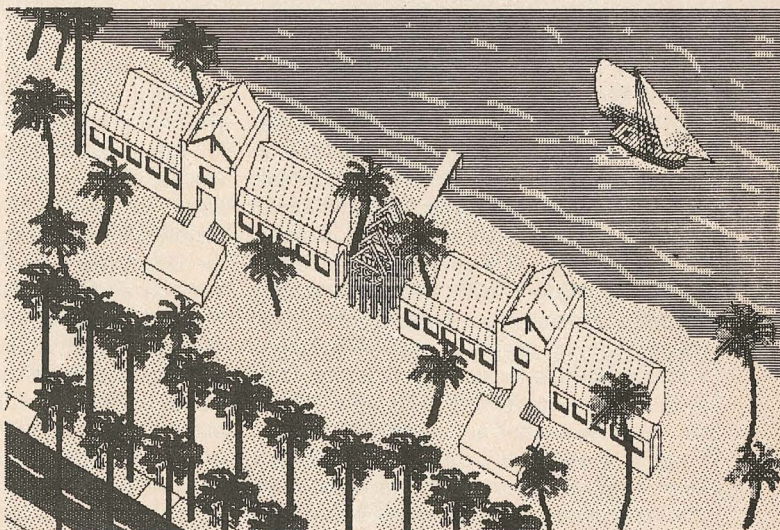
BPI has already had great success with its accounting packages for the Apple II and III; the major limitations have been in the hardware (How much can you do with a 6502 processor and 64K of memory?) With the Lisa's 68000 chip and a megabyte of memory (that's 1,048,576 characters, for those who are counting), you can bet that standard accounting procedures are a cakewalk for this machine.

There are other BPI packages available to expand the capabilities of *General Accounting*, but a small or medium-sized business might not require them. *General Accounting* includes a general ledger, general journal, cash register journal, accounts receivable ledger, accounts payable ledger, payroll ledger, cash disbursements journal, cash receipts journal, merchandise journal, and an invoice register. Happily, it uses Lisa's mouse for menu and item selection. Accounting is a dry enough discipline without having to use a keyboard for *everything*. There'll still be enough data entry to satisfy touch typing fanatics, and it's a true pleasure to grab the mouse on your desk, move it around, and watch the pointer on the screen follow your moves.

Some of the things you can do with *General Accounting* include producing single, multiple, and consolidated statements; recording budgets; comparing this year's activity to last's; creating and printing more than forty different reports; and keeping a financial history of all general ledger accounts. You can also print checks and customer statements—on a separate printer, if you prefer to. You can keep track of addresses and histories of vendors, customers, and employees.
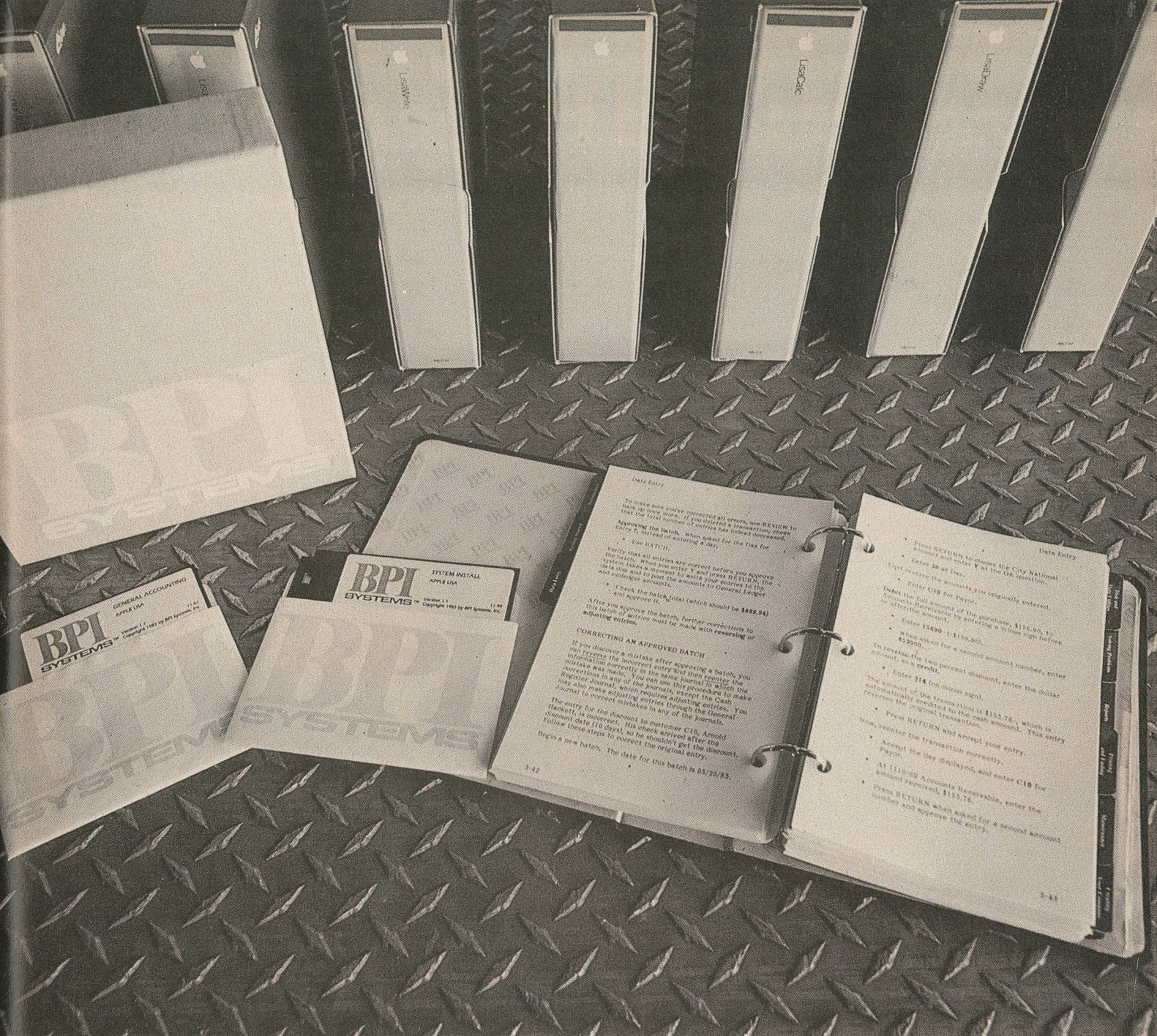
The package includes some rare conveniences, such as the "query" feature, which allows the user to examine reports and data on disk without leaving the task at hand. Since the Lisa prints a true screen image, you can see what a report will look like before sending it off to the printer. And since printing is a slow (dull) task, it's nice to be able to set up a string of print jobs in a queue, start up the whole batch, and walk away from the computer and toward the coffee pot. Or continue performing other accounting activities. The *General Accounting* system will let you know if it's okay to go ahead with the task or if you'll have to wait. There are only a few items (checks and customer statements among them) that Lisa will be unable to print while you go on with other activities. It's also possible to edit a printing job without stopping it.

So how much can the program handle without choking? After running into systems written in Basic, some users find that a definite concern. Not to worry, though; *General Accounting* is written in Pascal. Amounts up to a billion dollars (less a penny) are handled, accurate to the penny. The system can deal with up to 9,999 accounts per ledger, 100 departments, twelve months or customized accounting periods (one to thirteen per year), eight-digit invoice numbers, and up to twenty-nine-character names for vendors/employees/customers, payees/payors, general ledger account names, and so on.

What's it take to install the system? Only a willingness to follow a decently written set of instructions. The manual was not written for programmers only. A first-time user can expect to get through the step-by-step procedure successfully. There is also a "solving problems" feature to help you over the rough spots. Not only does an adequate explanation of errors appear on the screen, but the manuals give an understandable explanation of what the problem may be and (better yet) a probable solution.

Of course, the system isn't perfect. You don't have total control over the report for-

mats, which is a shame. The report capability is not up to the standard of the rest of the package. Considering Lisa's printing and formatting capabilities, a report generator would be a plus—perhaps as a separate package. Another welcome feature would be an alphabetic or alphanumeric chart of accounts numbering. The system allows only three- or four-digit numeric codes plus a one- or two-digit department code. Oh, well. It does do consolidations.

The payroll ledger is really lightweight. But if you need only a summary of payroll activity, it will probably be sufficient.

A business requiring more than an overall bookkeeping system will find the other BPI packages essential. These include a full payables, a full receivables, and a very complete payroll package. Needless to say, they will integrate with *General Accounting*. In fact, it

is possible for *General Accounting* to use files generated by other packages. The documentation gives fair warning, though: This procedure is tricky and not for the beginner. This is the only function that is so difficult.

The *General Accounting* system gives just consideration to the luckless data entry person. Bells and whistles don't count for a lot in a piece of software if the day-to-day work is a nightmare for the person who ends up using the system. In the BPI package, the screens that come up are not overcomplicated or dense, the menus are clear, and the prompts are self-explanatory. It's easy to back up and correct errors, and even unsophisticated users may find themselves setting up sophisticated procedures.

In short, *General Accounting* more than adequately fills a void for those most likely to purchase a Lisa—those with a business to

run. It can perform most, if not all, of the accounting tasks necessary to run a small or medium-sized operation. A second release of *General Accounting*, as well as the *Accounts Receivable*, *Accounts Payable*, and *Payroll* modules, has been customized to take advantage of the Lisa 2/5 and 2/10's multitasking abilities, increased disk storage, and the upgraded 2.0 operating system—all on 3 1/2-inch floppies.   ◇

General Accounting
BPI Systems
3423 Guadalupe
Austin, TX 78705
(512) 454-2801
Version 3.0.0 for the Lisa
$595

Turn Information

Instantly. Without complex codes. Without

Now the same typographic communication used by professional publishers and printers is just as easy and accessible as ordinary office typewriting or word processing.

New personal composition systems from Compugraphic let you mix everything from headlines and graphic symbols to *italics* and **boldface** type in a wide range of eye-catching styles. With instant visuals, charts, and tables to emphasize important points. But without the difficulty associated with conventional typesetting.

Don't let your important documents become lost in the flood of typewritten information that inundates office workers every day. Compugraphic of-

into Communication
special skills.

To: All Branch Managers
From: T.R. Martin, senior vice president
Subject: Typographic Communication

As you know, corporate headquarters has directed all administrators to expedite internal communications by **limiting all messages to a single page**. This requirement can best be met by using newly developed typographic composition systems instead of typewriter-style printers.

Typographic composition not only saves space but also enables you to create publisher-quality documents as easily as you currently produce word-processor output. This gives you several important benefits:

1. **More easily read materials that will be understood immediately and remembered longer.**
2. **Professional quality and credibility.**
3. **Unlimited mixing of sizes and styles for highlights and subheadings.**
4. **Easy creation of charts, tables, and other attention-getting graphics.**

In addition, typographic communication's **copy compaction** saves money because typeset copy occupies approximately half the space required for typewriter-style text. That means paper, postage, and handling costs can be cut by as much as 50 percent. The graph below illustrates this benefit.

For further details about this new form of office communication, please contact the purchasing department.

Space saved by setting typewritten documents in type.

Pages

100 —
75 —
50 —
25 —
0 —

74% 10 Pitch Double-spaced
65% 12 Pitch Double-spaced
48% 10 Pitch Single-spaced
34% 12 Pitch Single-spaced

Reviewing Lisa's Office System

BY CHRIS LINCOLN

Our expectations for software programs such as those on Lisa have changed a great deal in a short time. For example, though many early spreadsheet programs did not allow variable column widths, today few of us would buy a spreadsheet that did not provide them. Not so long ago, no one expected personal computer word processors to have dictionary/spelling add-ons, and yet today most word processors do have such add-ons, along with the ability to produce form letters from lists of names and addresses. The state of the art of personal computer software advances almost daily.

The Lisa Office System applications are both of the past and the future with respect to certain features, but all share the consistent Lisa Technology user interface. All are operated by pointing and clicking Lisa's mouse and by pulling down menus and opening windows on Lisa's electronic desktop. That alone makes them powerful tools for general business use because all can be learned quickly.

In addition, learning one of the Office System applications *leverages* the learning for all the applications. The Lisa Office System learning curve is smooth and easy and there is not the "relearning curve" apparent with so many personal computer programs when one does not use the programs constantly.

The Lisa Office System consists of the Office System operating system ($295) and seven applications programs. The applications are *LisaCalc* ($295), *LisaDraw* ($395), *LisaWrite* ($295), *LisaGraph* ($295), *LisaProject* ($395), *LisaList* ($195), and *LisaTerminal* ($295). Any of these programs can be purchased separately, although the Office System operating system must be resident to run any of them. Running any of the programs also requires a Lisa with at least one megabyte of random access memory and a hard disk. Specifically, the Lisa 2/5 (with a five-megabyte external ProFile hard disk) and Lisa 2/10 (with a ten-megabyte internal hard disk) require a 512K RAM upgrade ($1,495) to run the Office Sys-

tem applications.

A new Version 2.0 of the Office System that was released in January provides welcome improvements in speed (disk input/output is roughly twice as fast) and is included as part of the upgrade of original Lisas to Lisa 2/5s. Let's look now at each of the Lisa Office System applications programs, considering their features and ease of use and comparing them with similar programs on other personal computers.
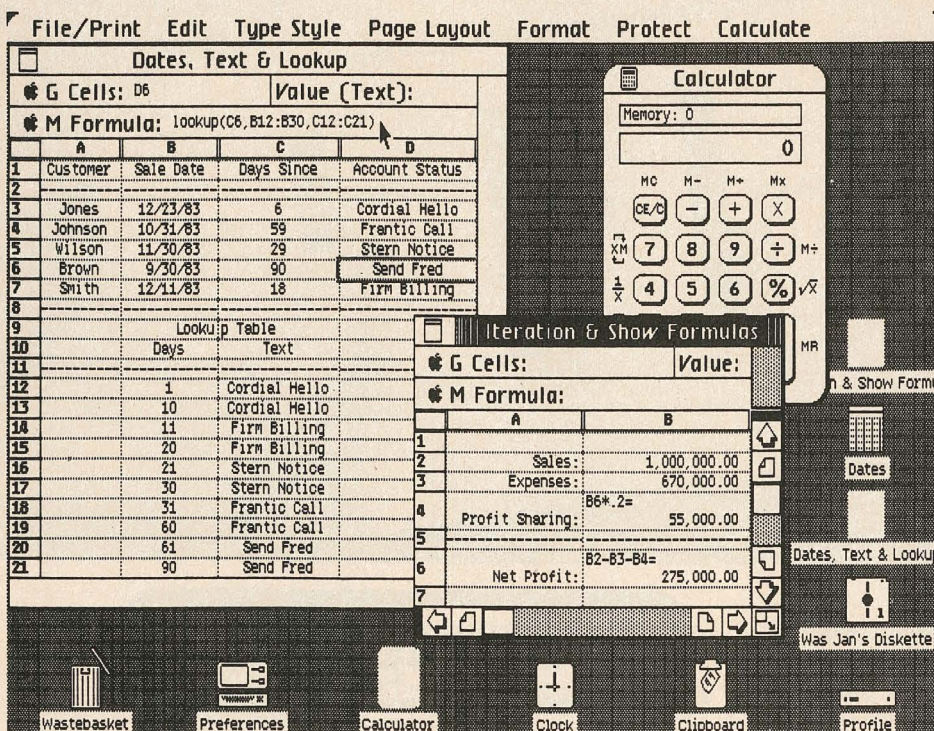
LisaCalc

LisaCalc is a "new generation" financial spreadsheet for Lisa. First made popular by the now venerable *VisiCalc*, spreadsheets generally consist of a large matrix of rows and columns in which numbers and text can be entered and formulas applied to the data. Spreadsheets can be used to do things as simple as balancing a checkbook or as complex as preparing a pro forma income statement. In fact, virtually anything you'd normally do with pencil, ledger, and calculator can be done quickly and easily with today's many spreadsheet programs. *LisaCalc* is no exception.

The *LisaCalc* matrix is 255 rows by 255 columns, and Lisa calculates to fifteen digits of precision. As you'd expect, the column widths can be varied using mouse or menu. The *LisaCalc* screen can be "paneled," or subdivided, into several horizontal and vertical screens at once, allowing you to see totals in row 255 while editing row 1. This feature is common only among the better new spreadsheets.

A list of the *LisaCalc* functions includes such financial functions as net present value, compound interest, and annuities. *LisaCalc* also provides a standard complement of trigonometric functions and conditional logic operators. In addition, several new functions have been added to *LisaCalc* in Release 2.0, among them standard deviation and an index function. Release 2.0 also increases the maximum formula length from seventy-five to ninety characters.

LisaCalc's handling of date arithmetic and text strings is especially noteworthy. Days, months, or years can be added and subtracted easily. The conditional logic function can be used to evaluate text in one cell and to return different text to another cell as the result of the evaluation. For example: If A1 = "Over-

LisaCalc *supports lookup tables, which can be cut and pasted from one part of a model to another. Note that formulas and values may be viewed on the spreadsheet at the same time.*
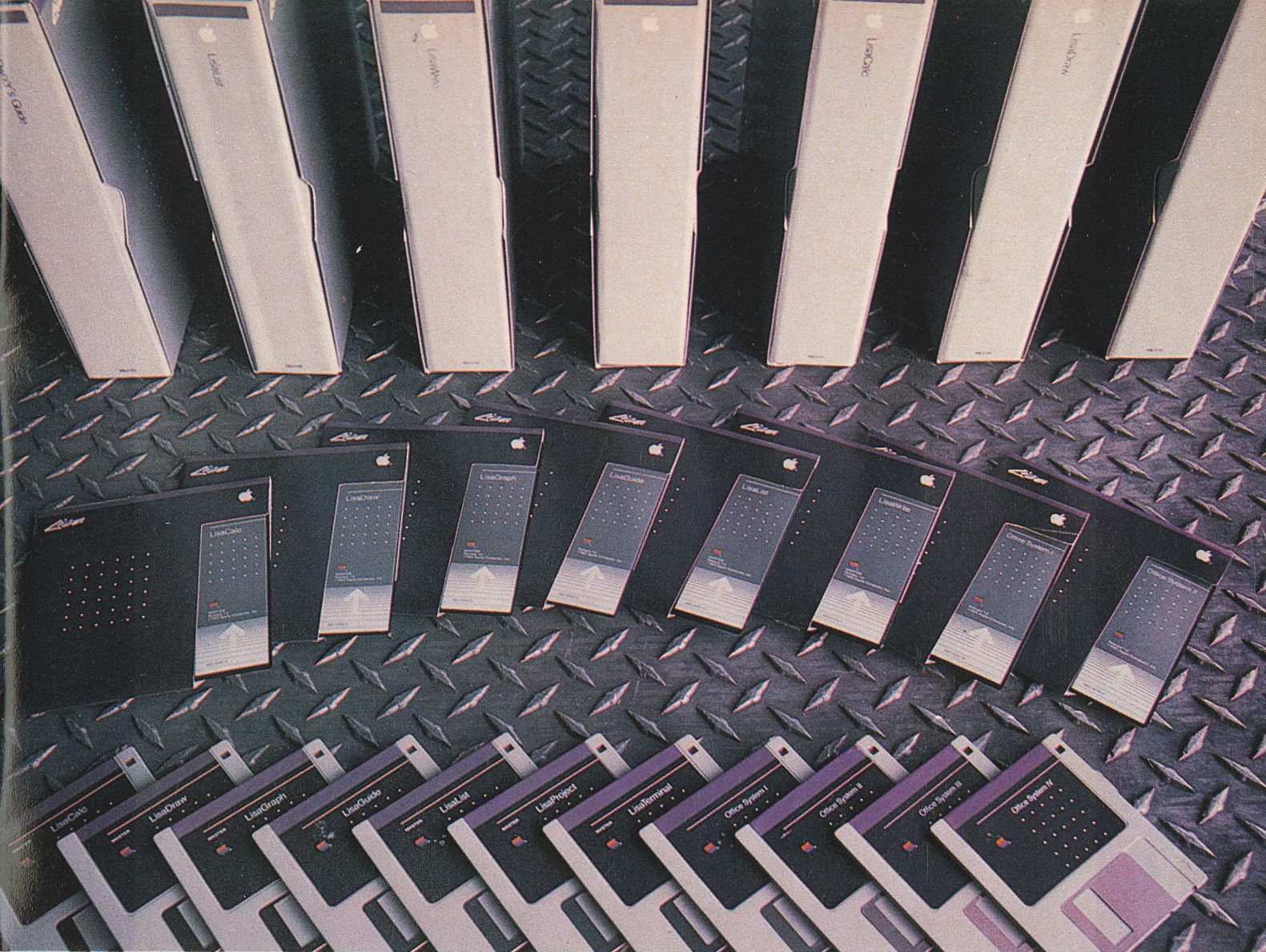
due", "Send Bill", "OK" (the commas in the formula represent THEN and ELSE).

LisaCalc also supports lookup tables, which are useful in calculating rates and conversions. The tables can be cut and pasted from one part of the model to another, with the relative position of all formulas maintained. This is so of all cut-and-paste maneuvers in LisaCalc.

Extensive formatting of text and numbers including exponential notation and several money formats is possible from the Format menu. LisaCalc also has the unique ability to print out formulas *and* values on the spreadsheet at once—excellent for checking spreadsheet logic.

LisaCalc automatically calculates models to convergence. In other words, when two formulas depend on each other for their solution (circular reference), LisaCalc calculates until there is no change in the value of either cell (convergence). Most spreadsheets do not do this at all or only allow the user to specify how many passes (or iterations) over the formulas to make. The problem with this arrangement is that most of the time you do not know how many passes it will take. The job becomes even more impossible when large

models with many circular references are being considered. A simple iteration example is included in the illustrations.

Lisa's standard type styles (except for the very large ones) are supported by LisaCalc, as is the ability to view small, medium, and large type without actually changing the pitch of the printout. No other spreadsheet we know of allows the user to view additional rows and columns of a spreadsheet model by changing the text display size.

Using the mouse as a pointer, you can select cells and ranges and install their locations in formulas. The mouse mechanics are the same as always—point and click. Many of the more often used menu selections can be performed directly from the keyboard by holding down the Apple key while pressing the appropriate letter. The cursor controls on the numeric keypad allow quick data entry and movement in LisaCalc without leaving the keyboard. LisaCalc tables (no formulas) can be cut and pasted into LisaWrite documents and into LisaGraph documents as data for the graph. The cut-and-paste operation is as simple as selecting the range of cells to be copied with the mouse and choosing Copy from the Edit menu or Apple-C from the keyboard.

Its extensive set of functions and tremendous available memory make LisaCalc as powerful as any spreadsheet now on the personal computer market. There may be spreadsheets with larger matrixes such as Lotus's 1-2-3, but LisaCalc models can actually be larger because of Lisa's one megabyte of RAM. 1-2-3 would run out of RAM long before it hit row 2,048.

There are, however, a couple of things that LisaCalc should do that it doesn't and some things that it would be good to see in a future release. To begin with, LisaCalc does not allow text "spillover"—that is, text cannot be typed continuously into a cell and spill over into adjacent ones (the entire text can be seen only on the cell contents line of the LisaCalc window). This spillover feature is now a common and popular way of inserting titles and narrative into a model. In addition, when it is supported, titles usually can be "frozen" so they don't scroll with the model. Also, LisaCalc requires that full pages of a model be printed. A range of, say, only a few cells cannot be selected for printing; instead an entire page must be printed. Both these things would seem easy to change.

A couple of features that would be nice to

see in future releases are named ranges (a la *Multiplan*)—the ability to name areas of the model and then compose formulas containing the names rather than cell coordinates (for example, Profits = Sales − Costs)—and macro capability. If *LisaCalc* had macros, a la *1-2-3*, it could execute entire jobs without requiring the user to enter and move much of the information.

LisaDraw

LisaDraw is the generalized graphics package for Lisa. With *LisaDraw*, anything you can imagine or conceptualize can be represented in dramatic fashion and printed out just as you see it on the screen.

Few, if any, programs on personal computers compare with *LisaDraw*. It is an outstanding package with which to do general technical and business illustrations. Drawings are prepared by making selections from a template at the left side of the *LisaDraw* screen. For example, pointing to the box symbol on the template allows you to draw a box by doing a click/hold and dragging the mouse until the box is the desired size. The box can then be moved around on the screen or resized by pulling on the "handles" that highlight it when it is selected. Text, lines, circles, polygons, and arcs are all drawn in the same manner and may be filled with any of the thirty-six patterns from the Shades menu.

LisaDraw also provides additional type styles, as well as the option to customize text with a hollow or shadow effect. Lines can be made black, gray, or white, with a choice of four different widths, and can also be made into arrows.

The Arrangement menu allows shapes and text to be aligned together and put in front of or behind each other. Shapes can be grouped together to behave as a single object. This ability to place shapes on top of each other and to group and ungroup objects makes *LisaDraw* distinct from *MacPaint*, which cannot do these things. A thoughtful feature is the ability to duplicate objects, place them next to each other, and continue duplicating as Lisa positions the duplicate objects exactly in a row or column. Scale is generally maintained when shrinking or enlarging grouped objects; text is the exception.

Features on the Page Layout menu let you display horizontal and vertical page rulers in standard or metric units and customize the ruler increments. The paper grid can also be made standard or metric. The actual size dimensions of a shape can be measured by selecting the Show Size option. The virtual size of the drawing paper can be enlarged to 96 inches by 48 inches—thirty-two square feet!

A powerful feature on the *LisaDraw* Edit menu, and generally included on the Edit menus throughout Lisa, is Undo Last Change—a real life-saver that is nevertheless somewhat unpredictable. Redo Last Change can usually be chosen, but Lisa sometimes alerts you that it cannot undo the last change.

Graphs from *LisaGraph* and charts from *LisaProject* can be cut and pasted into *LisaDraw* for customizing. At this writing it is not possible to paste a *LisaDraw* drawing into *LisaWrite*, or to place a *LisaGraph* graph into *LisaWrite*. This will change with the mid-1984 release of new Office System software, in which the cut-and-paste matrix will be complete.

LisaDraw renderings printed on Apple's Dot Matrix or Imagewriter printers are very impressive. Quality is excellent for going directly to overhead transparency film or for printing or copying.

The combination of the Lisa user interface and *LisaDraw* is the "essential" Lisa. It has the potential to improve our business and personal communications as well as to stimulate our imaginations and amplify our thoughts. Apple evidently observed some time ago the strong need for general-presentation graphics capability (including high-quality output) on personal computers. To many, however, the state of the art of personal computer graphics is pie charts and bar charts. *LisaDraw* breaks new ground.

LisaWrite

LisaWrite is generally the most maligned of the Lisa Office System programs. There may be some valid reasons, but on balance *LisaWrite* fits with the other programs and does word processing as Lisa would do it. *LisaWrite* is capable of doing virtually all general word processing tasks except perhaps writing a textbook with automatic indexes and an automatic table of contents.

LisaWrite has the standard features of most good word processing programs—headers and footers, full find-and-replace facilities, variable line and paragraph spacing, and so on. What *LisaWrite* doesn't have is the complex and difficult-to-use formatting and text-editing commands that most word processors have. All text editing and formatting is done with menus and mouse (or partially from the keyboard if desired). Any text formatting done from the keyboard always makes sense with a single character mnemonic—Apple-B for Bold, Apple-U for Underline, and so on.

A unique feature of the Search menu is that you can find and change text *including* its type style. *LisaWrite* also allows search and replacement of partial words and the substitution of wildcard characters when searching and changing text. Tabs are set by displaying a Margin/Tab Ruler across the page; using the mouse and the ruler, four types of tabs and three tab fill patterns can be set.

Release 2.0 of *LisaWrite* makes some cosmetic changes to the appearance of the tab and margin settings so they look even more like those on a typewriter. The margins on a particular section of text (a quote, for example) can be reduced by selecting the text with the mouse and using the Margin/Tab Ruler. Changing the header and footer areas is done in a similar way by displaying a Page Ruler. Pages can be previewed as they will be printed with headers and footers.

There are eleven different type styles, from fifteen-pitch Gothic to one-third-inch Classic. Two of these styles are proportionally spaced. All type styles can be made bold, italic, and so on. Few word processors have the ability to mix type styles on the screen and printer as Lisa does. This entire article was written using *LisaWrite*—point and click, peck, peck, peck.
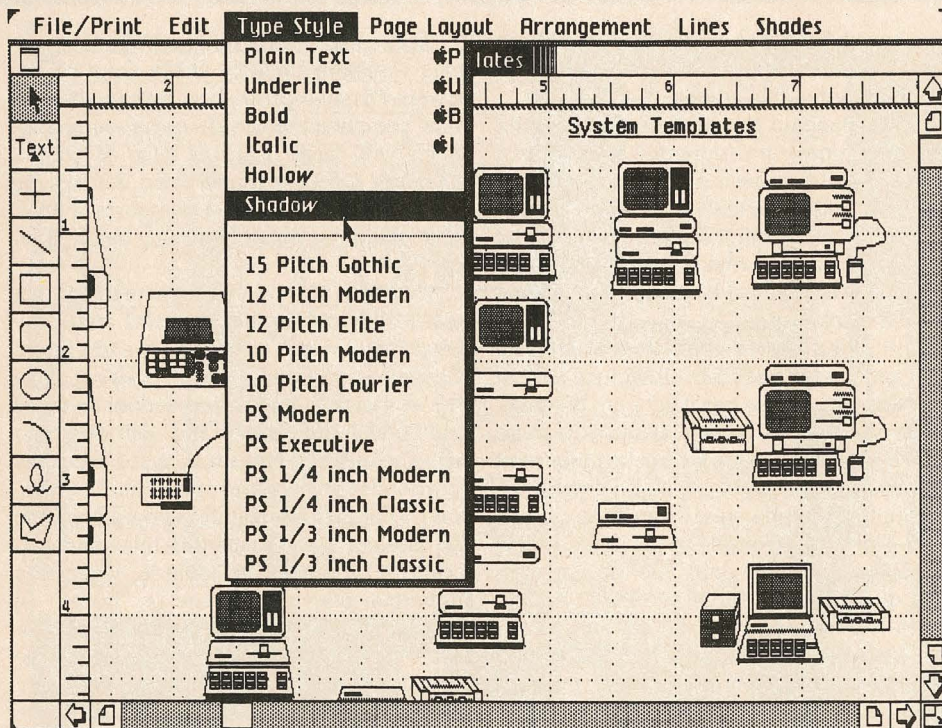
Text and graphics may be mixed in LisaDraw *and will print out just as they appear on-screen. Rulers and grid lines on the screen are formatting aids that may be "hidden" when not in use.*

Data Spectrum™

Bringing Information to light.™
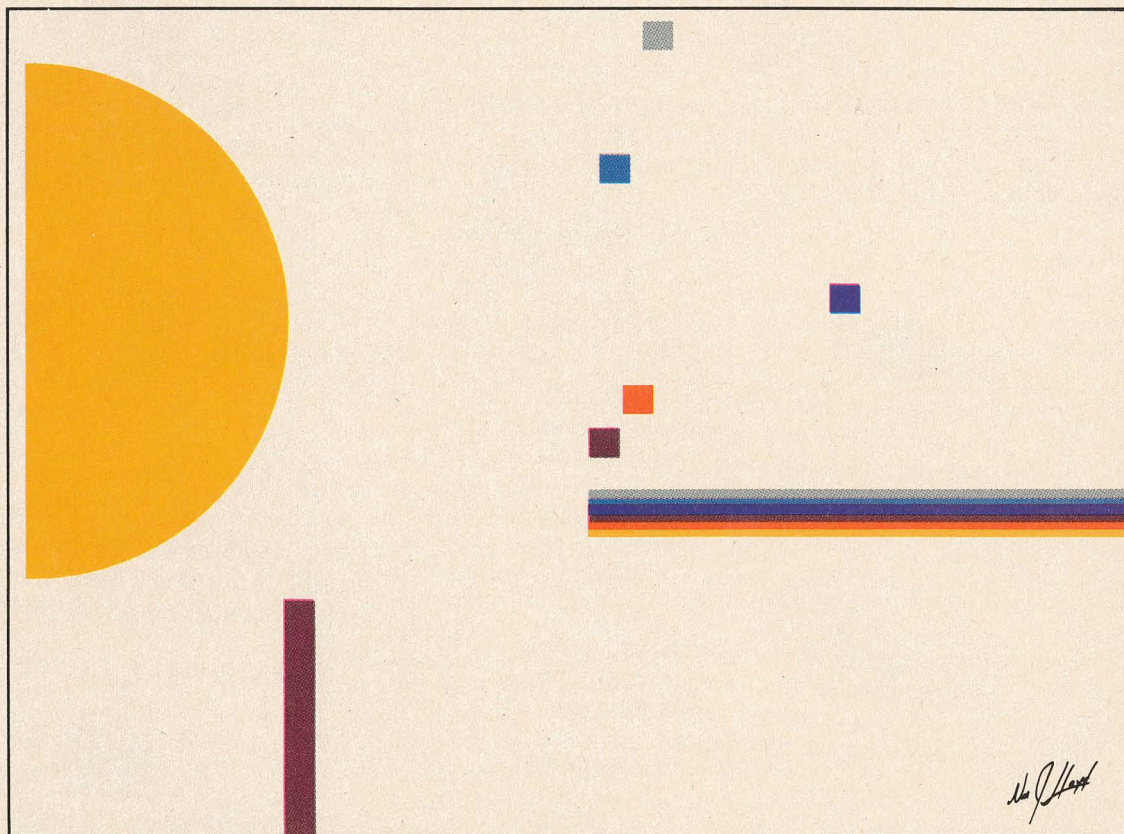
Would da Vinci have settled for a Crayon?

A great artist must command his tools and his medium to fully express his creativity...to solve his design problems. To solve information problems, you need to command an information management system that allows you the flexibility to design solutions tailored to your needs.

Data Spectrum™, is a comprehensive and powerful information management system for business and personal use that goes beyond "user friendly" to downright "people affectionate."

Data Spectrum is the master work of information management. It allows you to customize the organization of information to your specific needs. To search, sort, analyze, edit, evaluate and merge files as you want. And even to generate forms, reports, labels and standard letters.

All menu driven. All built in. With special features like "RAPID TRANSIT": A new file management approach that processes smaller files at blinding speed and automatically switches to "LARGER FILE ORIENTATION" to handle larger files.

Destined to set a new standard in the industry, Data Spectrum's on-screen tutorial blends text, graphics, humor, sound and animation to lead the user through the concepts, uses and operating techniques of this innovative software package.

On-screen tutorial and powerful program functions make Data Spectrum the universal information management system and problem solver. And a work of art. Our program up-date service and information "Hot-Line" will keep it that way.

So if you are looking for solutions rather than just software, Data Spectrum will bring them to light.

Write or call us for your nearest computer dealer carrying Apple or IBM versions of Data Spectrum. If there is no dealer in your area, you can order Data Spectrum directly from us...with a full money back guarantee. 1-800-232-6902 Illinois Residents 312-298-0997. Advanced Business Computing, Inc. 1580 N. Northwest Highway, Park Ridge, IL 60068-1471.

abc

As for its limitations, *LisaWrite* does scroll slowly. It also has the disconcerting habit of returning you to the top of the text after a Save & Continue, requiring you to scroll down to your last position.

The alert windows are well handled everywhere on Lisa. Error conditions usually evoke soft beeps at first, and if the condition persists, a window appears offering a clear description of the condition and its solution. Alert windows are also used to describe actions such as Save & Put Away and printer setups. Experienced users sometimes wish there was an option to turn off the alert windows—sort of an "I know what I'm doing, leave me alone" mode.

Even with the foregoing criticisms, there is only one real weakness in *LisaWrite*—the lack of a mail list facility and spelling checker. These features are standard issue on many word processors today, so why not on Lisa? It seems absurd that when *LisaWrite* documents can be made into stationery pads, names and addresses cannot then be repeatedly applied to the form letters. Again, the remedy is expected in the midyear release of new software. All in all, it is easy to get what you want out of *LisaWrite*, and the process is nearly as natural as using a typewriter.
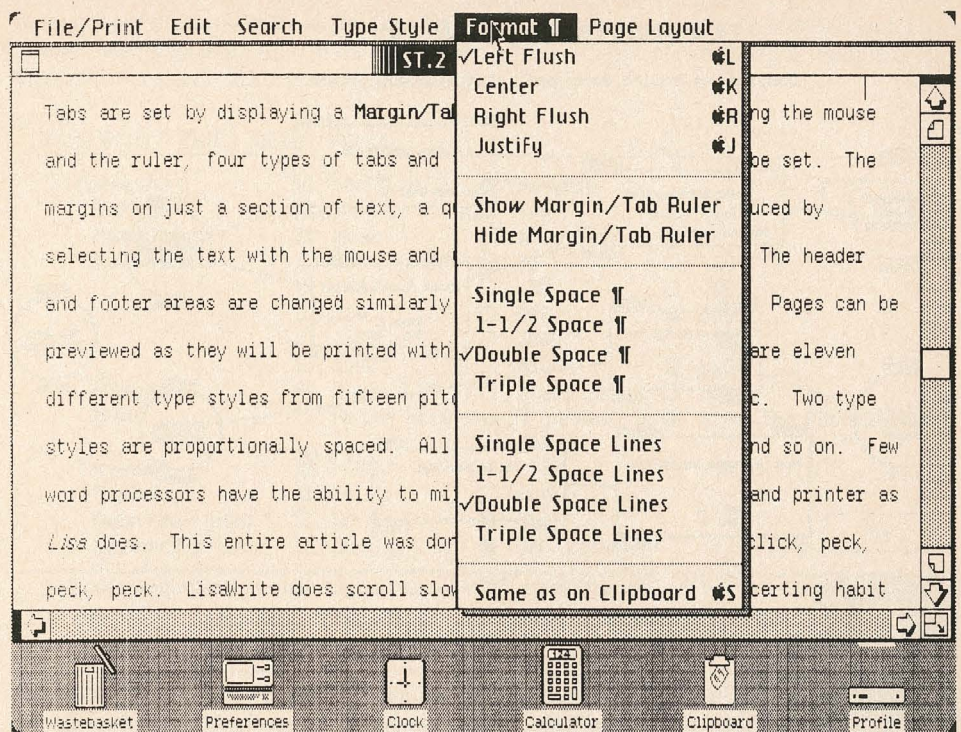
LisaGraph

LisaGraph is the simplest application to learn and use. Numbers are entered into a table the same size (255 × 255) as the *LisaCalc* matrix. No formulas can be applied to numbers in the *LisaGraph* table. Column A is always the X axis for the graph—usually time (months, quarters, or whatever). The next eight columns can contain data for the graph. Any data outside the first nine columns must be cut and pasted into those columns before it can be graphed. The *LisaGraph* screen can be paneled like the *LisaCalc* window to show more of the table or more of the graph.

LisaGraph can show pie, bar, line, and scatter graphs instantly. Changing data in the table left of the graph produces a new graph immediately. In bar graphs, single data sets can be plotted as lines for emphasis with the bars. *LisaGraph* can hide or show the legends and customize the Y axis minimum, maximum, and increments. Graph grid lines can be shown or hidden, and *LisaGraph* prints the graph grid lines *behind* the bars. Type styles can be modified in the usual way, and, unlike most business graphics programs, text can be inserted for emphasis anywhere in the graph, not just as titles and subtitles.

Graphs can be reduced to fit or displayed actual size. They can be printed out in quarter-, third-, half-, and full-page orientations without having to change aspect ratios to get the page layout right. The printout is always right the first time, provided you review your screen until it looks the way you want it to.

Data from *LisaCalc* is usually cut and pasted into *LisaGraph* for graphing. Graphs can be moved into *LisaDraw* for customizing. If, for example, a quarter-page orientation is cho-

Choices on the Format menu in LisaWrite *allow you to format text in a number of ways, as well as set paragraph and line spacing. The Margin/Tab Ruler, hidden in this example, lets you set margins and tabs quickly and accurately.*
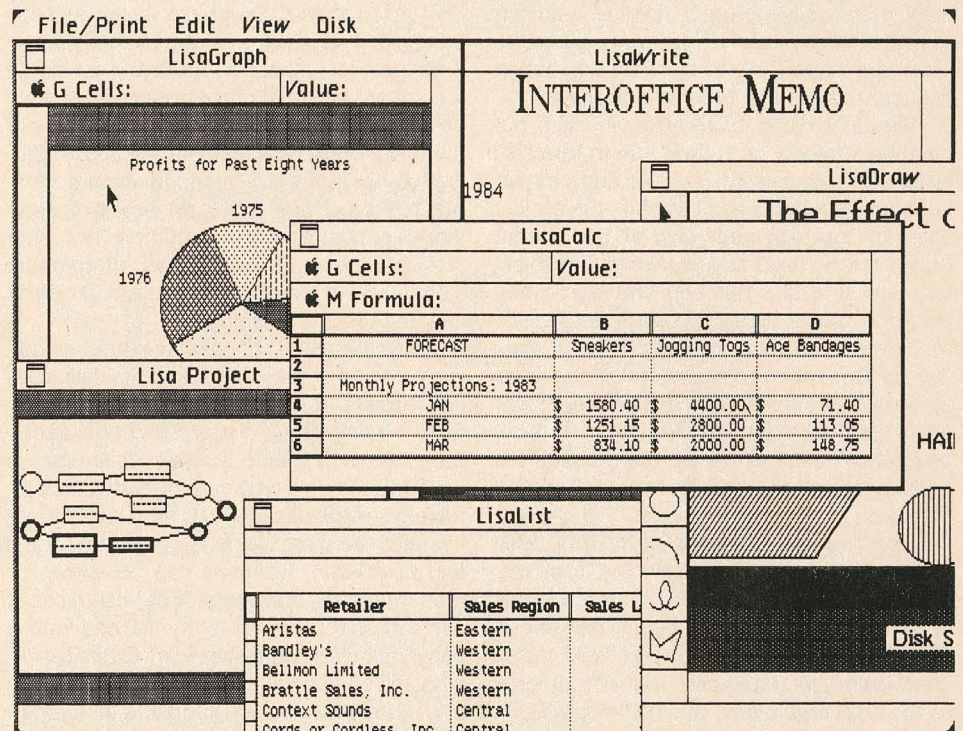
In this example, six of Lisa's Office System applications are shown in overlapping windows on the same screen. True cut-and-paste integration among all programs is not yet complete.

sen, four different graphs can be placed on a single page of *LisaDraw* paper. Moving the graphs to *LisaDraw* is done the same way that movement throughout Lisa applications is done—point, click, and paste. However, you can only move table information back to *LisaCalc* or into a *LisaWrite* document.

LisaGraph is fast and easy. It would be nice, however, if, like *1-2-3*, *LisaGraph* allowed you

to plot horizontal or vertical data sets situated in any part of the table, instead of requiring you to cut and paste data lying outside the first nine columns.

Stacked bars would also be a welcome addition. Business graphics usually are of certain specific types, and *LisaGraph* does most of them admirably. Graphs of scientific or mathematical functions are another matter, not

Construction Work

Completing Ground Work and Utility Installation
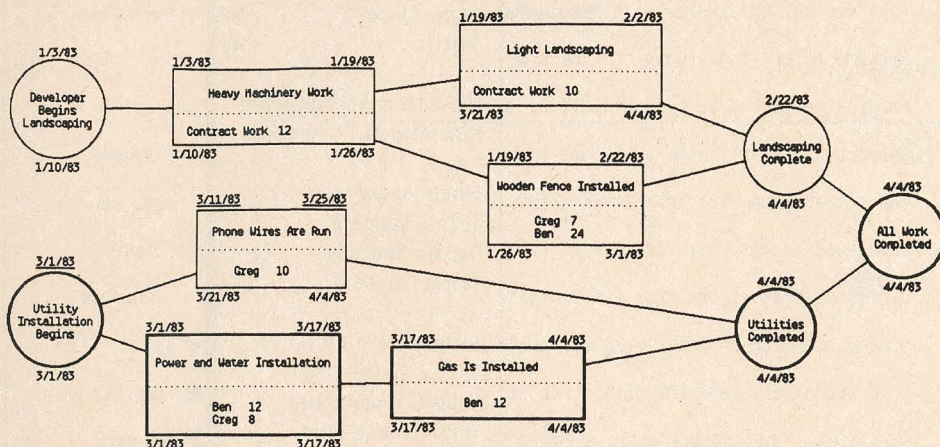
LisaProject has been called the "sleeper" of the Office System applications. In LisaProject, *complex items can be put on a time line in terms of tasks and resources. Project junctures, or milestones, are shown as circles.*

as easily done in *LisaGraph*, and perhaps the province of a smart third-party developer.

LisaProject

LisaProject is the "sleeper" among the Office System applications. Critical path or PERT project scheduling is not a well-known skill. Mechanically, *LisaProject* operates as the other applications do, but project scheduling concepts take a little time to "snuggle up to."

Basically, using *LisaProject*, you can put complex projects on a time line in terms of tasks and resources. You can add and subtract tasks and resources and "what if" the project dates by inserting your own or the current project status. Tasks are inserted in the project by filling in a task box with the task name, resource, and a number of days (expressed as a whole number) to complete the task. Project junctures are called *milestones* and are shown as circles. *LisaProject* always highlights the critical path of the project. If these tasks and resources on the critical path fall behind, the entire project cannot be completed on time.

LisaProject displays and prints out three types of charts: Schedule, Task, and Resource. The Task and Resource charts show slack time or overscheduling. The Schedule chart shows tasks with Early and Late Start and Finish dates. The *LisaProject* paper can be made as large as the *LisaDraw* paper, about sixty pages. In 2.0, up to forty vacation days can be specified, and the number of work days in a week can also be set. Projects can be up to ten years in duration. An entire project can be reduced to fit on the screen; editing and changing type styles are done the same way as in the other Office System applications.

LisaProject brings project scheduling capabilities to personal computers (particularly the graphics) that have been seen only on larger computers costing thousands of dollars more. In fact, *LisaProject* is superior to many of these programs. Some welcome enhancements would be the ability to schedule in hours rather than whole days and to track costs as well.

LisaList

LisaList is the Office System list manager. List management programs have recently emerged as a subset of complete database programs. List managers generally are single files where several records are viewed at once because of the row and column structure of the list. Unlike full-fledged database programs, list managers generally cannot compile reports or new databases based on summaries or combinations of multiple files. With *LisaList*, lists of, say, personnel information, sales leads, or lists of customers and products can be built and maintained.

In *LisaList*, a list of up to one hundred columns can be designed with specific data types and formats. The data types are Text, Number, Date, Money, Time, Social Security Number, Zip Code, and Phone Number. Data entry is verified as to data type and formatted according to user specifications. Column widths are changed the same way as they are in *LisaCalc* and *LisaGraph*. Columns can be shown or hidden to view only parts of the list. *LisaList* can sort on all columns of the list and Find & Show records selected based on extensive criteria and comparisons, for example, = (same as), < (up to), or < = (up to and including). *LisaList* "virtualizes" the list to the hard disk so lists larger than available memory can be built. The ability to change the records in the list or to add and delete columns is not restricted, but once the column data type has been chosen it cannot be changed.

As list managers go, *LisaList* is powerful and capable, but it is not a full-fledged database program. It was not intended to be. It can do things, such as sort on all columns, that no list manager and few databases can. The predefined data types make the building and management of personal databases quick

and easy. *LisaList*'s weakness is that it is the "orphan child" application. Although it is the logical place to put mailing list information, *LisaList* information cannot be copied into other Office System applications programs yet, so form letters can't be done for *LisaWrite*. (This will change by midyear too.) When *LisaList* is further integrated into the Office System, it will be easy to perform detailed analysis on *LisaList* data with *LisaCalc*, to graph results in *LisaGraph*, and so on.

LisaTerminal

LisaTerminal is the late-arriving communications program for the Office System. Telecommunication is fast becoming the most asked-about personal computer application. If your computer can "talk" to other computers, then you're really on your way into the Information Age. There are now hundreds of commercial and noncommercial databases and community bulletin boards to contact with your computer. The Source, CompuServe, Dow Jones, and Dialog are but a few of the big ones. In addition, most companies have large mini or mainframe computers that must provide information and data processing for the company on a daily basis. *LisaTerminal* is Lisa's "ear to the ground."

Mechanically, *LisaTerminal* operates as the other applications programs in the Office System do. (So what else is new?) A piece of *LisaTerminal* paper is torn off and configured from the Compatibility menu for the particular host requirements. A configured *LisaTerminal* document can be made into a stationery pad and used to auto-dial the host in the future.

LisaTerminal is a full asynchronous communications program. Baud rates can be set from 50 to 9600 with parity checking. *LisaTerminal* has Hayes 300 and 1200 baud modems as configuration choices. It also supports full DEC VT 52 and VT 100 terminal emulation with 132 character lines.

It is not immediately obvious, but Lisa's multitasking Office System can support communications to more than one host computer at a time. With two modems, for example, two pieces of *LisaTerminal* paper could be torn off and two host computers dialed up. Information could then be cut and pasted from the DEC minicomputer in engineering to the IBM mainframe in accounting. Other Office System applications could be on the desktop at the same time. Text from one *LisaTerminal* window could be cut and pasted into the other or into *LisaWrite*. This simply is not possible with any other personal computer.

With the availability of the Apple Cluster Controller, *LisaTerminal* will also support IBM host communications in similar fashion. *LisaTerminal*'s strength is its faithfulness to the Office System metaphor. It doesn't "look" like a communications program and can be learned and used as naturally as the other Office System programs. Unfortunately, *LisaTerminal* has only recently become available, and there was a crying need for it six months ago. Installation of *LisaTerminal*, prior to

Release 2.0, required an upgrade to Version 1.2 of the Office System operating system, and this may account for some of the delay in its arrival.

When you've finished working on the Lisa desktop with one or all of the applications, the desktop can be "saved," neat or not. The "state" of the Lisa desktop is saved whenever it is powered off. That is, when Lisa's power switch is pressed, Lisa begins a set of housekeeping maneuvers to save the desktop. An alert window announces that "Lisa is putting everything away before turning off." If the desktop is cluttered, the next time Lisa is powered up it will bring up the cluttered desktop just as it was left. In addition, the only time that ProFile "scratchpads" are recovered is when the system is powered down.

It is clear that the individual Office System applications need work in some areas, but it's also clear that they do some things that no other applications programs have done before them. It is certain, too, that all the limitations described will be overcome in the short-term. The point is, and always will be, that Lisa is the sum of its parts. The Lisa Office System is the most complete set of general business applications of its kind. There simply aren't any equivalent programs generally available on personal computers to compare with *LisaDraw* and *LisaProject*.

And other Office System components reviewed here all get the job done in well above average fashion, even without the Lisa Technology user interface. And that's the point: When the very good, even great, applications in the Office System are combined with Lisa's profound ease of use, the Lisa solution is hard to beat.

In LisaTerminal, *as in all Lisa Office System applications, selections are made by pointing to a menu choice and clicking with the mouse. Baud rates in* LisaTerminal *can be set from 50 to 9600 with parity checking.*

FROM THE PEOPLE WHO BROUGHT YOU JUPITER...

In 1979, Voyager I completed a journey of 483 million miles, sending back to Earth the first close-up pictures of Jupiter and its moons. **TELOS** played an important role in the Voyager mission, supporting the Telemetry and Image Processing systems in the areas of engineering and science data processing and display, image reconstruction and enhancement, and still photo animation.

From a stream of digitized bits, **TELOS** software helped to reconstruct the first photographs ever taken of active extra-terrestrial volcanoes on the moon Io. And **TELOS** software assembled hundreds of still photos of Jupiter's clouds to create a startling color motion picture of the turbulent Jovian atmosphere.

In 1981, **TELOS** received from **NASA** a Public Service Group Achievement Award, "In recognition of outstanding accomplishments in the development and operation of the ground systems which allowed spectacular results in the exploration of the Jupiter and Saturn systems and interplanetary space."

In 1986, **TELOS** software will help to guide the **Galileo** spacecraft to Jupiter and when the Instrument Probe separates from the satellite and parachutes into the thick Jovian mist, **TELOS** will be ready.
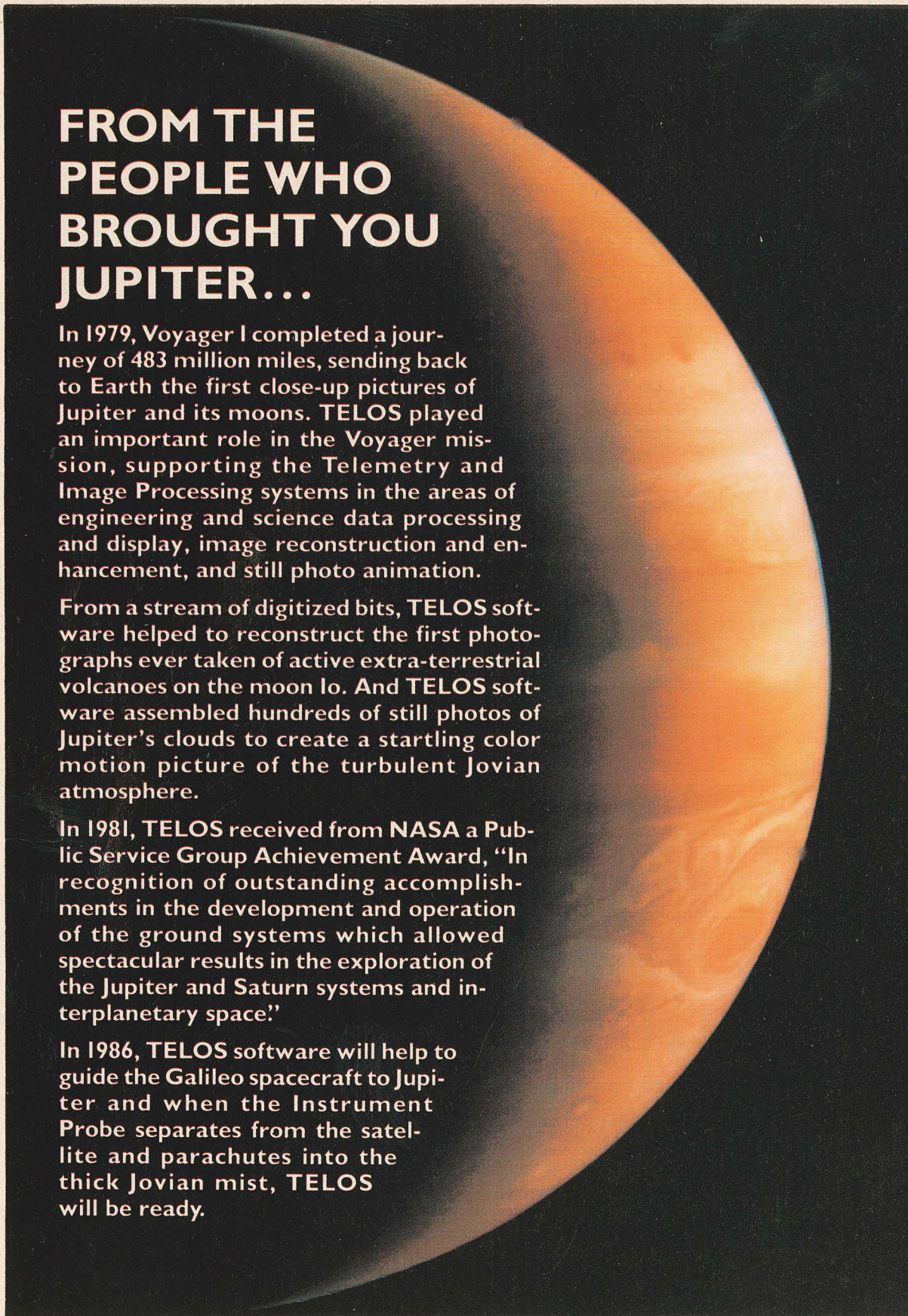
TELOS EXPLORES THE OUTER LIMITS OF THE Macintosh™

Telos software is helping NASA explore the heavens. And soon a remarkable new line of Telos software for the Macintosh will help you do some exploring of your own.

Telos congratulates Apple on the introduction of the Macintosh

TELOS SOFTWARE PRODUCTS • 3420 OCEAN PARK BLVD. • SANTA MONICA, CA 90405 • (213) 450-2424

TELOS™

Secrets of The Finder

A Revealing Interview with Programmers

Bruce Horn and Steve Capps

BY DOUG CLAPP

Bruce Horn and Steve Capps are programmer's programmers. Together, they authored the Macintosh System and Finder, putting in grueling workweeks of ninety hours and more. Their creations are large, complex programs that give Macintosh and Lisa their distinctive, icon-oriented personalities. System governs the appearance of windows, menus, and everything else on the Mac screen, and the Finder serves as a visual way to manage documents and tools.

Mac wouldn't be Mac without the software achievements and innovations of Horn and Capps. Both recently met with writer Doug Clapp for a no-holds-barred interview that covers topics ranging from Mac's design to the experience of being a programmer in the environment that is uniquely Apple Inc. The result was a freewheeling, opinionated, and completely candid conversation.

ST.Mac: What do you guys think of all the media attention?
Capps: It's fun. There's inequities in it, and sometimes it bothers you and other times it doesn't.
ST.Mac: A lot of Macintosh articles have appeared recently. Were the articles accurate, or did they miss the point of Mac?
Capps: Fifty-fifty.
Horn: I've noticed that Microsoft's taken a lot more credit than they deserve. They're trying to take credit for a couple new designs...in the file system they did have one little idea, but they're trying to take credit for helping to design the Memory Manager and the Finder and

all this stuff, and they did none of the above.
Capps: They *helped* with it. Neil Konzen especially.
Horn: Neil *only*, pretty much. He was the only person. But Microsoft didn't have nearly that much to do with the software design.
Capps: It was [Bill] Gates taking credit for Neil's work.
Horn: And taking credit for *our* work, too. And that's completely wrong.
ST.Mac: A phrase that's omnipresent when discussing Macintosh is *user interface*. How do you two define user interface?
Horn: Well, everybody's an expert on "user interface."
Capps: Including us.
Horn: I think it's just the way that the system is presented, and the way that all the options are presented to the user. You can either type a command to a keyboard, or you can point to a command, or....
Capps: The best quote I've heard about it is "You don't get any manuals with video games." I don't know who said it, but you can certainly walk up to a video game and, three or four quarters later, you've figured it out.
Horn: That's a definition of a *good user interface*.
Capps: So that's the idea: How can we make a *computer* like that? And the neat thing about that is, it could be *fun* to learn, too!
Horn: The other way is...how do you encapsulate the complexity of something that's easy and, as you get better and better, "unfolds" instead of presenting you with all the complexity at once?

Opposite page: Bruce Horn (left) and Steve Capps, authors of the Macintosh System and Finder.

Photos by Bill Fitzgerald

Capps: To further what Bruce said, that's what they did with Lisa: They designed it for the imbecile. So when you suddenly discover that you *do* have a brain, you're stuck with this thing that, every two minutes, tells you "Well...I'm about to print. Is that okay?" One thing that would be interesting is a system that, once you got a little bit better, would tailor itself to you, that would unfold.

ST.Mac: Like an "expert mode"?

Capps: Well, it wouldn't be a "mode."

Horn: Levels. Expert levels.

ST.Mac: How do you explain Macintosh to people who don't know anything about computers?

Horn: You basically say that Macintosh is a computer that you don't have to be a computer whiz to learn how to use, and it's ten times better than an IBM PC at about the same price. And it's going to be one of the next neat things that happen.

Capps: How about this: "A vibrator for the mind?"

ST.Mac: I like that.

Horn: He's really on to something. But make sure that *he* said that, not me!

ST.Mac: How about a quick breakdown of what's in ROM, and what the System and Finder are?

Horn: Okay. What's in ROM? Basically, the whole user interface code: windows, menus, dialog boxes, and things. That's all in ROM. The file system's in ROM, Memory Manager, Textedit, lots of things.

Capps: But none of the "looks"—none of the ways that things look. The way menus work is in ROM, but not the way they look.

Horn: The way they look is in "definition procedures" that are loaded from the resource file in the system. There's also a thing called System, that has all the fonts and all the definition procedures for windows...

Capps: ROM patches...

Horn: And ROM patches.

Capps: But we won't talk about those.

Horn: Right.

ST.Mac: Then what's the Finder?

Horn: The Finder's this application that "sits on top" and lets you do operating system-like things: copy disks....

ST.Mac: It's the visual shell you see on the screen.

Horn: That's right.

ST.Mac: The System's about 145–146K?

Horn: Mostly fonts.

ST.Mac: How do you break that down percentage-wise?

Capps: Three-quarters fonts, maybe a tenth desk accessories.

Horn: More than that! The Control Panel's 10K!

Capps: Okay, maybe a quarter desk accessories. Then there's def procs.

ST.Mac: What's a "def proc?"

Capps: Those are the things that implement the scroll bars, the buttons, menus, windows, controls.... Those are probably another eighth of the size, there. And then there's miscellany. It adds up to seven-eighths.

ST.Mac: So most of it's fonts, some of it is desk accessories, and the rest is def procs?

Capps: That's the order of bigness, yeah.

ST.Mac: And the Finder is about 45K. What's that made up of?

Horn: It's hard to say.

Capps: About 36K code, and the rest menus and things. There's 3K of text in the Finder.

Horn: And there's another maybe 1K of text for dialog boxes. All the dialogs and menus are in the resource part of the Finder. Actually, the whole Finder is in the resource part of the Finder!

Capps: There are messages in there that you will probably never see.

ST.Mac: How many messages?

Horn: Over a hundred.

ST.Mac: What's a really bizarre message?

Capps: My favorite dialog—which I could never put into English, so we left it in Yiddish—is this: Say you've got a folder named "Fred," and its window is open. Now, say you drag a file named "Fred" into that folder. And they're on different disks. Now, obviously, you replaced
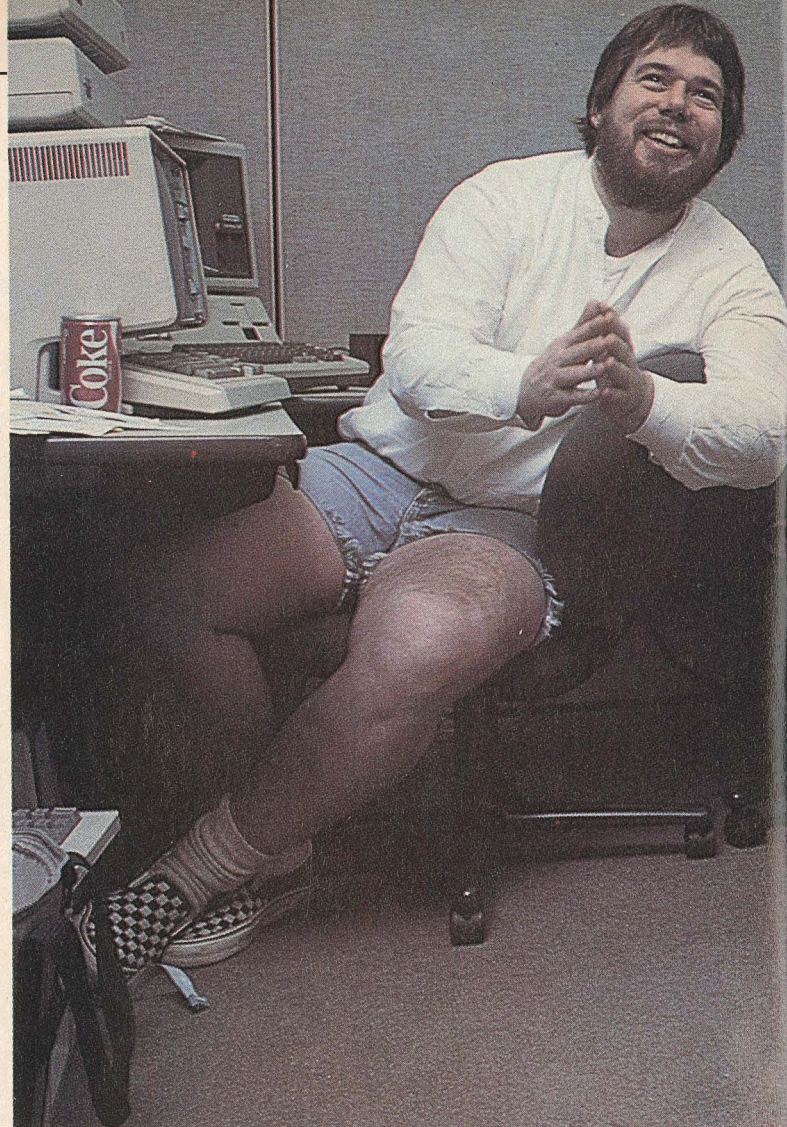
Deciphering MacSpeak

The high-tech, high-speed world of Horn and Capps often results in a flurry of words and phrases that may baffle even experienced programmers and advanced hobbyists. Here's a quick rundown of some of the more undecipherable words in the accompanying interview.

Alto and **Star**—Two legendary Xerox computers. The Alto was primarily an early development machine for in-house use at Xerox. The Star was a Xerox product that was, in many respects, the forerunner of the Lisa and Mac. It had an extremely high-resolution screen, a mouse, windows, and other Lisa/Mac-like attributes.

BitBlt—A technique developed at Xerox for fast manipulation of graphics on-screen.

Heap—An area in RAM memory that holds data from programs. The heap is the memory device that allows Mac to transfer (via the Clipboard) information between different applications programs.

Instant Pascal—The in-house name for the language that will be released by Apple as Macintosh Pascal. Instant Pascal is a revolutionary, interpreted version of Pascal that is lightning-fast and offers easy access to Macintosh's range of graphic manipulations.

the thing named "Fred," because that's where you're putting it! Now tell me, in one sentence, what's about to occur? That you can't do that.

ST.Mac: "Name conflict?"

Capps: No, it's different than a name conflict.

Horn: I think Steve's first one was "File folder not okay."

Capps: No, I think it was "Destination folder conflicts with source. Yuck!"

Horn: No one will probably ever see that one.

ST.Mac: The System and the Finder are entirely on disk?

Horn: Yeah. The only thing that's not on disk is the ROM. And the Mac is on the table, and....

ST.Mac: The operating system then, in Macish terms, is only doing the hardware stuff: mediating with the disks....

Horn: That's right. That's the file system: opening and closing files. That's the operating system.

Capps: That's like, say, one-fourth of the ROM. What most people put in ROM is a little piece of their operating system, and usually they get the rest of their operating system off disk. We put the whole thing into ROM, and four times more.

ST.Mac: How much of the System and Finder are moved into RAM when you turn on the machine?

Horn: All of the Finder. Well...no. Lots of the resources in the Finder are on disk until you need them. The dialog boxes for certain things don't come in until you need them. Those are purgeable. There are some purgeable things, but most of them aren't. So, yeah, about 45K goes into RAM. But when you do a disk copy, Steve purges [deletes from memory] parts of the Finder that aren't used, so you have more buffer space.

Capps: But for a running program to come in, there's X amount of overhead. So, when the program's running, there's maybe 50K of the heap used. Probably more than that, because fonts take up a lot of room. So the working set of memory, less the Finder, is maybe 70K.

ST.Mac: Would it be preferable to have all the System and all the Finder in ROM?

Capps: No, because then you couldn't fix the bugs!

Horn: Yeah, you couldn't change it. It would be preferable to have more memory. Then you could leave part of the Finder. Then if you were, say, changing applications, the Finder could sit in this little part of memory, then—boom—come right out. That would be nice, but we don't have enough memory. Maybe on the Big Mac.

Capps: It's just too high-level to put into ROM. Because, a year from now, we might not think it's that great. We might want to make it different.

Horn: That's why disks are so neat: You can give somebody a new disk, and their Mac will actually be different. It'll be better.

Capps: But someday that's going to happen. Somebody's going to write the ultimate word processor, the ultimate drawing program, the ultimate Finder, then stick 'em all into ROM.

ST.Mac: What's the history of the Finder?

Horn: Well, when I came to Mac, they were experimenting with things like a big picture of a disk on-screen, with little filenames on it. Bud Tribble invented the term *Finder*, because the Lisa thing was a "Filer." And Bill Atkinson basically convinced everybody, a few months before I came, that the "pictureFinder," which showed a large disk on-screen, wasn't the way to go. The original Lisa Filer was a standard filer, with lots of dialog boxes and text and things you type.

Capps: In the vernacular, it was the "twenty-questions Finder."

Horn: No, the "thousand-words Filer." So I came in and just started playing around with "classes," and I built a Finder mock-up using little pictures that you could open up and see inside.

ST.Mac: Is it fair to say that you "Xeroxized" it?

Horn: Yeah, pretty much. There were some new ideas.

Capps: That's with a capital "X."

Horn: Right. "Xerox is a trademark of Xerox Corporation."

ST.Mac: So the Finder as we see it today had its genesis in you, and you had your genesis for those ideas at Xerox?

Horn: That's basically it. But not the stuff I *did* at Xerox; the stuff I *saw* at Xerox. Anyway, after I did the Finder I showed it to Bill Atkin-

---

Purge—Macintosh's use of internal memory is a complex subject, even to experienced programmers. In Mac, program code is relocatable (able to be shifted around within memory). Certain programs or portions of programs can also be purged, or removed from memory, when not needed.

ROMs—In Macintosh, the 64K of ROM memory holds more than 480 tightly compacted routines. In general, the ROM contents are one-third operating system (routines that work with the disk drives and memory), one-third QuickDraw graphics routines that give Macintosh its incredible graphic capabilities, and one-third Toolbox. The Toolbox portion of ROM contains code that can be used by programmers to implement the Mac user interface: windows, menu bars, dialog boxes, mouse actions, and more. Having these routines in ROM, and thus readily available to programmers, will ensure that all future Mac applications programs share a consistent behavior.

Smalltalk—A new computer language, authored at Xerox PARC, that works, unlike traditional programming languages, with the concept of object classes. In Smalltalk, program segments are objects that transmit and receive messages from other objects. Some of the Smalltalk ideas are now found in Lisa's Clascal, a class-oriented Pascal.

Ted Kaehler—A computer scientist and Xerox employee.

Xerox PARC—PARC is an acronym for Palo Alto Research Center, the famous Xerox research laboratory that was home for many of the discoveries that now appear in Mac.

son, and he went off with some people and changed the Lisa Finder. Is that about right, Steve?

Capps: Yeah. We were getting close to shipping Lisa. It was the spring of '82 and it was, like, "You better not make any more changes." But they didn't like the twenty-questions Finder.

Horn: They especially didn't like it once they saw ours. But I didn't Xeroxize any of the original Finder stuff. I actually hadn't even seen the Lisa Finder, and I had only seen one or two of the previous attempts at Mac. I really started from what I thought was right, and what I thought Xerox had done that was right. So I really didn't "take anything else"; I really started from scratch.

ST.Mac: A lot of it had to do with your Smalltalk background?

Horn: Some of it. Some of it had to do with the [Xerox] Star, and Starware.

Capps: The Star is a very similar thing.

ST.Mac: How do you compare the Xerox Alto and the Star to the Mac?

Horn: The Alto was just a development machine that people used. It did everything; it wasn't specialized for one purpose. It didn't use the mouse like the Star did. The Star is the one that the Mac's more like, because it was more visual.

Capps: The Alto was like your first girl friend.

Horn: Yeah, the first girl friend that would do anything with you. You could go to the beach....

Capps: And it was terribly slow. There were a lot of things wrong with it, but it was just so neat at the time that it didn't matter.

Horn: Well, that was 1972. And it wasn't *that* slow, and you could run the world's best word processor on it, which was *Bravo*, and you could use different fonts and things. And not much later, the Ethernet [a networking system] was up and people could use that!

ST.Mac: We'll get back to that later, but I want to ask: How do you go about planning a project as vast as the System and the Finder?

Horn: We didn't plan any of it, hardly. You just do a little bit of thinking ahead and decide what you want it to do. The Finder was *somewhat* planned.

Capps: That's about the biggest chunk that would ever get planned at Apple. That's not Apple's style, to sit back and plan something like that. Now at Xerox, people will do throwaway systems for a year, then, three or four years later, they'll get it out. But Apple just doesn't have that...patience. It's a wholly different style. One could argue either way: Apple will get out the stuff that makes them rich, while Xerox will get out the stuff that's consistent, but may not be the best.

Horn: Also, I would say that Xerox will get things out that will be extensible, while Apple might make some shortsighted things and say, "Oh, well, we'll fix that later, but we'll get the thing out now."

Capps: Yes, but that's being too negative. I think both ways are viable.

ST.Mac: At school, they teach you to plan the problem, then plan the inputs, and the outputs, then....

Capps: And those people are cranking Cobol for Blue Cross right now.

ST.Mac: It's amazing that a company of Apple's size, with almost a billion dollars a year in gross revenues, would put only eight people on the Macintosh software team. Were you understaffed, considering the amount of software that had to be written?

Capps: That depends. If you want to ship and have every program done, and not be up the night before, then there were too few people. If you want to work your behind off, and have loads of fun, it was just fine.

Horn: That's true. But I also think it would have been nice if we'd had somewhat better tools. Our tools were kind of Dark Ages tools. Capps went and made our tools more usable, but it would have been nice if we had gotten tools from the Lisa group.

ST.Mac: Like what?

Capps: Nice debuggers, for one thing. When I was in Norway, I wrote a debugger that showed you all the registers on-screen all the time. You could say "Step-step-step" and watch the register changes.

ST.Mac: What a great learning tool.

Horn: Absolutely.

Capps: What you *really* want—I was thinking about this last night—is for Instant Pascal [the interpreted Macintosh Pascal] to run assem-

bler and to make your "Instant Assembler" your debugger.

Horn: That's just saying, "Write the language and let it debug itself," which is Smalltalk.

ST.Mac: But you're saying that there were and there weren't too few people?

Horn: Right. But actually I liked it. I like working with a very small group. I like working with Steve, because it's just the two of us. When you get more people involved with something, you start having to communicate too many things.

Capps: It also depends on the people. If you can get people that are self-motivated, you can have an infinite number of them; you don't care. But if you get people that need dependencies, it just starts taking more and more of your time.

ST.Mac: How long did it take to write the Finder?

Horn: I guess it took about nine person-months.

Capps: Longer than that.

ST.Mac: How many hours a week?

Horn: Unknown.

ST.Mac: Let's get the definitive explanation of the Resource Manager.

Horn: Okay, the Resource Manager is a set of routines that let you load and write out objects that have types, and IDs, and possibly, names. These objects are loaded in dynamically when you need them, by calling the routine. The Memory Manager works with the Resource Manager by purging them out. And that's basically all it does.

ST.Mac: Like overlays?

Capps: It does overlaylike things. The Segment Loader is written to use the Resource Manager, and the Segment Loader does the overlay stuff. But what the Resource Manager does is let you do the same thing with anything you like, not just code segments. But also, it's an interface to the inside of your file, from the outside. It tells people who want to look at it what's in it. So let's say you have a document, and you want to put a picture in it. You put the picture in the resource part of the document, and anyone else can look at that picture and change it. The idea of having applications use resources is that people from the outside can reach in and change all the text, or all the pictures, or completely change the language, without having to redo the entire program.

ST.Mac: And nobody's ever done this before.

Horn: Well, there are things like Pascal code libraries, but....

Capps: Nobody's ever done it throughout the entire system.

ST.Mac: It's really a programming achievement.

Capps: Really. For instance, there's a list of things in the Finder resource file that gives what should be brought into RAM when you swap disks. Because of the setup, you can sit outside the program and drastically tune up the performance of that program. You could have done this before, in the old days, by having a special file that had these tuning parameters in it, but nobody would ever do that, because it wouldn't be worth the effort. I mean you wouldn't say, "Well, I'll have this special file that will do the tuning parameters, and I'll have this special file that says how to do the fonts, and I'll have this special file for this, and so on." Nobody would ever do that.

What this does, because it becomes a way of life, once you get over the hump of learning it, is make it so you don't ever go: Writeln (Hello, Doug). You go: GetString, then Writeln (String). It's a little different programming than what people are brought up on, but once you overcome that hump, you've got it made.

ST.Mac: How difficult is it going to be for third-party developers to learn all this stuff?

Horn: It's going to be difficult for some. People that are reasonably smart will pick it up fast.

Capps: Also, we've got to get real good sample programs out there. That's the best way to learn.

ST.Mac: Are there any features you wanted to add, but didn't have time for?

Horn: It would have been nice to have had a little more time to look at it. There were some things we wanted to speed up—and we're going to work on that stuff.

Capps: In general, anytime you write software you want to throw it all out and write it again. That's a rule of thumb.

How to get your Macintosh programs to market.
Quickly.

Call Data Encore. We're already duplicating 3½" microdisk software for the Apple® Macintosh™ system. So we're up to speed and ready to go.

As a subsidiary of Verbatim, the world's leading producer of flexible disks, we're one of the few companies with 3½" duplication capabilities. From raw materials to disks to packaging to delivery to your customers. This total responsibility ensures you of a quality product you'll be proud to call your own.

Why not call us toll-free at 800-872-8778. In California or outside the U.S., call collect (408) 720-7400. Or write us at 585 N. Mary Avenue, Sunnyvale, CA 94086. Because if you want to harvest profits from the new Macintosh faster, you'll want to get your software to market fast.

Data Encore℠
a Verbatim Company
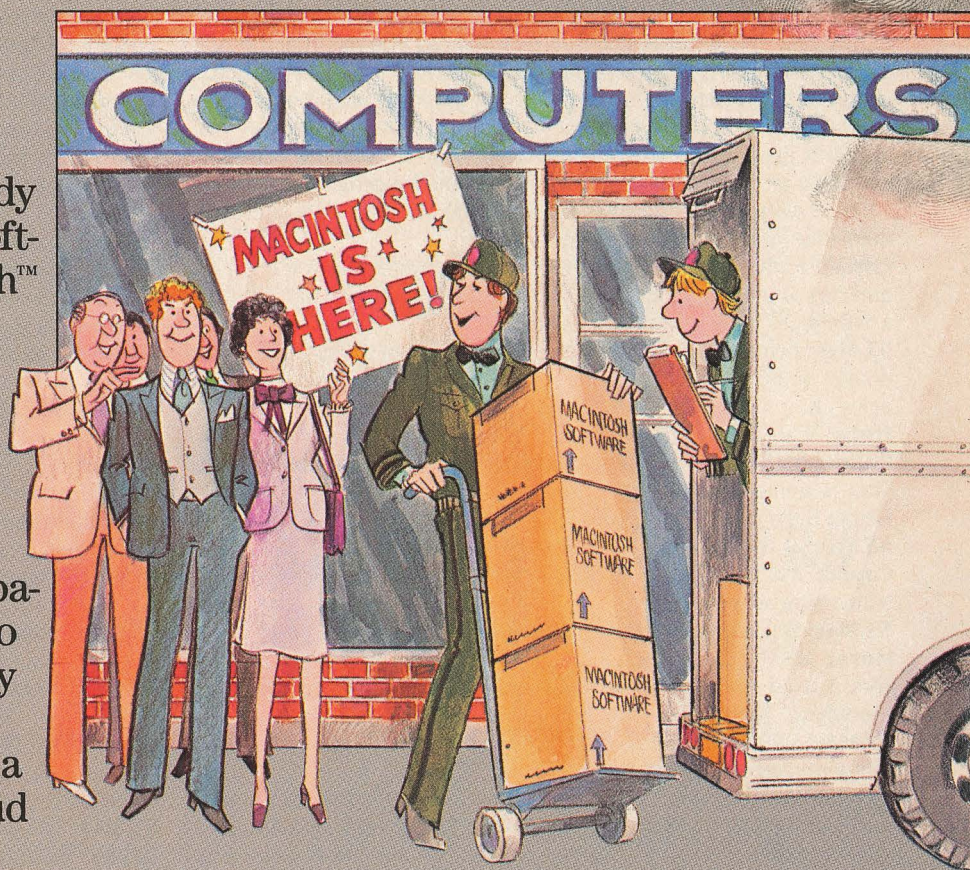
Horn: We did a pretty good job on the Finder. There are some pretty good concepts.

ST.Mac: Will we ever see more entries on the menu bar?

Horn: No.

Capps: There might be a few more entries on the Special menu; that's it. In fact, we might get rid of "Put Back." Put Back is kind of useless. Put Back doesn't make any sense, except on a multiprocessing system. Then you want to put back something, but you need to know what process owns it.

ST.Mac: How much of a constraint is the 128K limit?

Horn: It's pretty tight. In our disk-copy stuff it seems to be a problem, because you have to have a certain amount of system stuff around if you want to be in the Finder and do a disk copy, and maintain the visuals. That's what hurts us a little bit.

ST.Mac: So the memory limit is the wall you keep running up against?

Capps: It's hard to say. We've seen Lotus's *1-2-3*, and it won't run on a small machine very well. But it's also a huge, huge program that does a ton of stuff. One of the first things I did at Xerox was work on a collection of separate programs that all talked the same language. And we said, "That's not so great." So the next thing we did was write this *huge* program; and that required you to run it on a machine that had super memory and super disk-swapping—it was a big mainframe. And *that* didn't turn out to be good. So one might argue that you could have *1-2-3* be separate programs, and, if you had a quick way of jumping between them, you might get away with it on a small machine.

Horn: It depends on whether you want to do it [jumping between] explicitly or nonexplicitly. Whether you want the program to decide when to swap or whether *you* want to decide when to swap. Microsoft does that [allows you to decide].

Capps: Right. And I think you can tune the system so you'll get an amazing amount of function out of that small memory. It just requires a little more work—you can't be sloppy.

ST.Mac: Perception is reality. It doesn't matter what the system's really doing, or how much memory it's using, only how the user perceives it.

Horn: Yeah—only whether you get what you want.

Capps: It's like the way people surprise you daily with the Apple II. They're going to do that with the Mac.

Horn: Even more.

ST.Mac: Is the 68000 an easy processor to program on?

Capps: It'd be easier if it had a completely orthogonal [regular] addressing scheme.

ST.Mac: But it's more orthogonal than most processors.

Horn: Absolutely. It's incredibly good compared to a lot of processors. I used the 8086 for a while, and it had special registers for everything! The 68000 is lots, lots better.

ST.Mac: What were the hardest parts to program?

(Long pause)

Capps: With the Finder, the problem for me was that I'm just not very anal. And I can't do the things where you have to pay attention to a thousand details. I can pay attention to a thousand details as long as they're *romantic* details.

Horn: He's specifically talking about disk copy.

Capps: Disk copy I'm not good at.

ST.Mac: It's harder to do things that aren't inherently interesting.

Capps: Well, the scrap [the Scrap Manager] was just as compulsive, and it was probably equally as hard as doing the disk copy stuff, but it was more fun. The neat thing about the ROM is that you're always trying to crunch code, which is fun.

ST.Mac: Crunch code?

Horn: Making it as small as possible with the most utility. But I'd rather that we used *compulsive* than *anal*.

Capps: Larry [Kenyon, author of the Mac operating system] is very compulsive. He'd have to be that kind of person to do what he's doing.

Horn: That's right. We need that, but [what Larry's doing] doesn't require a lot of creativity. Sometimes it does, but....

ST.Mac: It's nuts and bolts stuff.

Horn: Definitely. You know what you need, you know what you have

to hold up, the airplane wing has to be this strong....

Capps: Well, Bruce, what was your hardest thing?

Horn: Not programming at all. The hardest thing has been, all along, convincing people that we need something. It was resources at one point, resource grouping at another point. It's been things all along—trying to convince people what's right, what'll make things sensible. Programming was much less difficult than that.

ST.Mac: So the politics were harder than the programming.

Horn: What I ended up doing was convincing Steve [Capps], and then the two of us would go and talk to people. I used his support for that; it was really useful.

Capps: I want to correct one thing. *Politics* is such a pregnant term. It's not politics in the usual, backbiting sense; it's more that there's an ingrown conservatism in everybody that says, well, if you've thought something through, then it must be right. Otherwise you wouldn't have come to the conclusion you came to. So, if somebody comes up to you with an idea that's a light year, or at least half a furlong, better, there's a natural tendency not to think it through right away. And the person that comes in with the idea, of course, has thought it out, so he's way ahead of you. It's just a normal thing. And Bruce really thinks ahead. More so than other people, he's just way out there in space, telling you about something that he understands perfectly. He's too impatient to let us catch up to him. Also, other people have just shot off to Saturn, and they don't want to switch courses.

Horn: What they did was this: They had the rocket already built, and the rocket could only make it to the moon, and that's all they thought they really wanted to do. And it wasn't a sensible rocket. And what I was trying to do was to say, "Hey, if you put these little attachments on the end, then you can put another booster on and really *get* to Saturn." Sometimes they've thought it out and sometimes they haven't. It takes a different way of thinking, rather than saying "Hey, we've got *this*, why don't we just add something to it?"

ST.Mac: It's easy to be complacent.

Horn: It's easier to live with what you've got than to try something new.

ST.Mac: Will there be a second software release and, if so, how will it be different?

Horn: Yes.

Capps: There'll be a couple of releases. One right away, just to fix the embarrassing bugs that we're surprised more people haven't discovered.

Horn: We have a few bugs to fix; we'd like to speed up some things and finish some things that we thought were finished but weren't. Also, we'd like to make another release to support hard disks, when they come out.

ST.Mac: Will Mac support more than two disks? Will it support a hard disk?

Capps: Yes. The *operating system* will (*pause*) well...you know the standard file listing in the Finder? Think if you had a hard disk with 500 files. Are you going to *scroll* through 500 files?

ST.Mac: So you could do it now, but it'd be painful.

Capps: Yeah, but something has to be done.

Horn: Right. Also, the Finder is optimized for the ejectable disk, so we made some decisions about how that should be done that aren't appropriate for hard disks. Some of that stuff will change.

ST.Mac: But, given a controller card, you could do it right now. It just wouldn't be easy to use.

Horn: Sure. Larry's operating system is completely geared for anything like that.

Capps: There's a concept built into the ROM of completely external file systems. So while our file system is optimized for these small disks, you can have a whole 'nother way of storing it out on a hard disk.

ST.Mac: A different driver.

Capps: Well, see, a "driver" is something that talks to disks. "File system" is the next layer up that says, "I want filename 'Doug' on this disk." There's a whole concept that says you can have external file systems, you can make external calls over networks, and stuff like that.

ST.Mac: Some skeptics say that command line–oriented systems, like MS-DOS, offer more flexibility than Mac's icon-oriented system. Well?

Capps: First of all, it's a *shell*, not an operating system. And anyone who would say that has probably never used ours, and if that's what they want, they can code one!

Horn: That's right.

Capps: That's the beauty of Mac.

Horn: You could have a desk accessory that was your command line operating system.

ST.Mac: That's a neat idea. You could have a little box with an A> prompt inside!

Horn: Well, we can't do EXEC files [batch files in MS-DOS] that say "Run this program, then this program, then this one." We're not optimized for that; we could have a little *routine* that does that, or a program that does that, or we could even put something in the Finder. But right now, that's not what most people want to do.

Capps: There's a different head. When you're a programmer, you want something that's different than what the average person off the street wants. And the average person isn't going to figure out how to use EXEC files, much less want them. But they'll be convinced they need them, if the salesman gets hot.

ST.Mac: Is Mac a "bet the company" machine?

Horn: I guess. It's the neatest thing that Apple's come out with lately. It's the *only* thing that Apple's come out with lately.

Capps: If Mac failed tomorrow, the company wouldn't die tomorrow, but it definitely wouldn't flourish.

ST.Mac: What's the competition for Macintosh?

Horn: The competition currently is the PC.

Capps: The competition is people's stupidity, I guess.

Horn: No, not people's stupidity, but their unwillingness to try something new.

Capps: Well, they're ignorant. They'll think that...well, *1-2-3*'s going to be on the Mac, so they'll think....

Horn: That if it says IBM on it, that's enough for me. That's what we're up against.

Capps: I feel that the main job of marketing is to get people in there using the machine. If they can do that, we'll sell all we can make.

ST.Mac: Will *1-2-3* be out before there are 512K machines?

Capps: Probably not, but about the time that *1-2-3*'s done, there'll be 512K Macs.

ST.Mac: Which would be when?

Capps: Soon. We have 'em in the lab right now.

ST.Mac: This summer? This year?

Horn: This year, probably.

Capps: Jobs would promise you by the end of this summer.

ST.Mac: But it's up to the chip manufacturers more than it's up to Apple.

Capps: You got it.

ST.Mac: Who will buy Macs? Apple says knowledge workers. Do you two agree?

Horn: I don't like the term *knowledge worker*, but, yeah, I agree. And I think a lot of college students will buy it, and a lot of people who would've bought PCs will buy it, I hope. I think that more people will buy it than we would ever suspect, or market for.

ST.Mac: Do you want to speculate on Big Mac?

Horn: What's Big Mac? Some people think that Big Mac is already here: the Lisa 2. And some people think that Big Mac is a 512K Mac with a double-sided Sony [disk].

ST.Mac: If you two could decide the changes for the next iteration, what would they be?

Capps: Change the software very, very little, so you'd have complete compatibility, big-screen, and big bits.

Horn: Big disk and big memory.

ST.Mac: How big?

Capps: As big as you could afford. When we say "big-screen" we don't mean a larger screen, we mean more pixels, more dots. But then we're talking high-tech, because then your video's scanning at ranges up into FM.

ST.Mac: Is 1,024 by 1,024 an achievable goal?

Horn: Oh, that's pretty big, that's pretty big. Well, the Star's 1,024 by 768, and that's very nice. That's what all the Xerox computers have now.

ST.Mac: Is that achievable?

Capps: Yeah, if you want to pay for it.

Horn: Gray scales are a good one, too.

Capps: It's a tradeoff. You might as well not throw more dots on the screen. You might as well use your memory for visual depth, instead of for width and height.

ST.Mac: More bits per pixel, rather than more pixels on the screen.

Capps: Right.

ST.Mac: Any chance that future Macs will use enhanced processors, like Motorola's 68020 [a full thirty-two-bit processor, running at twenty megahertz, versus the 68000's eight megahertz]?

Horn: That would be nice. The 68020 would really make things zip.

Capps: That machine isn't real yet.

ST.Mac: Have they shipped any samples?

Horn: I don't think so.

ST.Mac: I can't even imagine how fast a twenty-megahertz processor would be in Mac.

Capps: But in two years, it'll be too slow.

ST.Mac: What will Mac be like in five years?

Capps: They'll be around the same way as Apple IIs.

Horn: There'll be outrageous software on it. In five years, there'll be a lot of people who know how to program on Macs. It'll be pretty amazing, I think.

Capps: Five years from now, we're going to look back on the Finder and laugh. We'll look back at *Alice* [a real-time, 3-D chess program written by Capps] and say, "Boy, what a clunky game." We'll look back at *MacPaint* and say, "That was pretty good for its time."

ST.Mac: Will we ever see a color Mac?

Capps: Yeah...we'll make the case in teal blue.

Horn: And a polka-dot red, white, and blue keyboard.

Capps: You know the problem with color, don't you? It's the fuzziness caused by a black matrix; color is too expensive, and you can't stare at it all day. So when all those problems get solved....

Horn: When it's cheap, too. Our tube is incredibly cheap.

Capps: And you've got to get memory that's a lot bigger.

ST.Mac: What kinds of programs would you like to see written for Macintosh? What types of desk accessories?

Horn: I'd like to see some stuff that Steve would consider boring. I'd like to see some really great engineering stuff. You know, how to design your airplane on Mac. General-purpose design tools would be really exciting. Because I'd use 'em.

Capps: I think that chip design stuff would be great. Right now, to do chip design, you either have to use graph paper or you have to have big machines. And they're getting the silicon foundries down small enough where you could almost afford it on your own. So having design tools that were on Mac would be real fine. Then your average freshman in college, if he was motivated enough and had a couple of thousand dollars, could get a chip made.

ST.Mac: How about mass-market stuff?

Capps: A checkbook balancing program.

Horn: Riiight. (*Laughter*)

Capps: Seriously, teaching stuff. Education's going to be really big.

ST.Mac: And new desk accessories?

Capps: All the business...crap.

Horn: We're *really* excited about business stuff.

Capps: That'll be done [original desk accessories] by Lotus's *1-2-3*. To give you a great analogy: I was working on a phone dialer today, just a list of names. And somebody came up and said, "Aren't you going to make it look like a Touch-Tone pad?" There's a reality strangeness there. Why do we need to make a thing *look* like a Rolodex, if it doesn't serve a useful purpose. It's neat if it does. You don't want to teach people a new way of looking at a clock, you want a desk accessory clock to look like a clock.

ST.Mac: Objectizing things can get ludicrous if carried too far.

Horn: If *objectizing* means being the same as in real life. It's as if, in *MacPaint*, when you clicked on a pattern, you could paint for a while, and then it would dry up, and you'd have to click on a pattern again. (*Laughter*)

ST.Mac: Or the tip of your pencil broke when you were drawing.

Capps: Yeah. Then I suppose you'd have to bring out a sharpener, right? Let's model the real world, right? Well, that's silly.

Horn: That's why our disks look like disks. Mac doesn't come up with, "Ooh, it's a filing cabinet!" That's a case where the real world is right.

Capps: Then there's the Bic lighter desk accessory, where you click and this flame comes out.

Horn: And sets your window on fire.

Capps: You've heard about the paperweight ornament? Well, ornaments are allowed to be—it's not a hard limit—only about 8K. So you have this ornament that's 7,990 bytes and it does nothing. It just sits there, but it freezes all your ornaments from moving. But you're right: I think the world will gravitate toward being all ornaments; you won't have any central application anymore, just these really smart ornaments that know how to talk and interchange information.

Horn: But each application *can* have its own ornaments, because each application has its own resource file. So you can have ornaments that talk to each other. We've actually talked about doing some neat stuff that way.

Capps: That's going to be a neat way to sneak multiprocessing in.

ST.Mac: When we get 512K, will we have full concurrency?

Horn: The system's not set up to be concurrent.

ST.Mac: MacBasic is fully concurrent, and events [the way Mac handles inputs from the keyboard or mouse] are set up to make concurrency easy, aren't they?

Capps: I guess the point is that, because of the way the Event Manager is written, you can kind of do concurrency, but you've got to handle all the dispatching of events to processes. Get it? Anyway, you've got to do all the dispatching, whereas in other systems, the dispatching might be handled automatically. But that's very simple to implement. Then we get to storage management. That's another thing about multiprocessing: Who owns what storage? Do you just partition up storage, and say: "This is your chunk, thou shalt not cross"? That's pretty easy, and we have support for that: You could have multiple heaps. But when you start saying that these guys are sharing *all* memory, and so this guy wants to purge, should he be allowed to purge another process's stuff? Probably not.

ST.Mac: Those are the traditional problems of multitasking.

Capps: Right. We'll definitely do the...legerdemain, does that sound right? Well, we'll do that, and the average user will think it looks great! But it wasn't designed from day one. That's a polite way of saying it.

ST.Mac: Okay. You guys are hot-shot programmers. Lots of readers would like to become hot programmers. What advice would you give them to become good programmers?

Capps: Oh, God! Turn 'em on to Instant Pascal [released as Macintosh Pascal] on Mac, and take a book, or take a class, and just do what you want to do.

ST.Mac: You're in favor of Pascal as a first language?

Capps: Doesn't make any difference.

Horn: I think Instant Pascal.

Capps: Well, Pascal's great, but Instant Pascal—it's the environment that's the selling feature, I think.

Horn: I think it's clear—Steve doesn't agree with me—but I think that Basic might teach you bad habits, where Pascal really teaches you about data structures and control structures. So I think Pascal's much more preferable.

Capps: I agree with that. But what I'm saying is that it's the *environment*. You can go get Pascal or Basic for any machine in the world, but you won't have the Instant Pascal environment, which is an incredible aid to learning.

ST.Mac: So get proficient at Pascal, then learn assembly language?

Capps: No. Once you get good at programming, you can speak any language.

Horn: That's right.

Capps: There's only a few classes of languages. If you know Pascal real well, you're not going to pick up APL or Lisp. But if you know Pascal, you're going to pick up Basic and C. And if you know one assembler, you know them all.

Horn: That's right. And Lisp is in this other class, and Smalltalk's in even another class.

Capps: And APL's in a weird class. SNOBOL's kind of half in one class, and....

ST.Mac: Are these worthwhile languages to learn? I like Lisp, but I'm not very good in it.

Horn: I think Lisp is great. I think Smalltalk is incredibly great.

Capps: What was that quote we read? "Never learn a language that doesn't teach you anything." If you know Pascal, there's probably no reason to learn Basic, except for historical purposes.

ST.Mac: So Instant Pascal's the way to go?

CHICAGO NEW YORK

FORMERLY APPLEFEST & PC'83
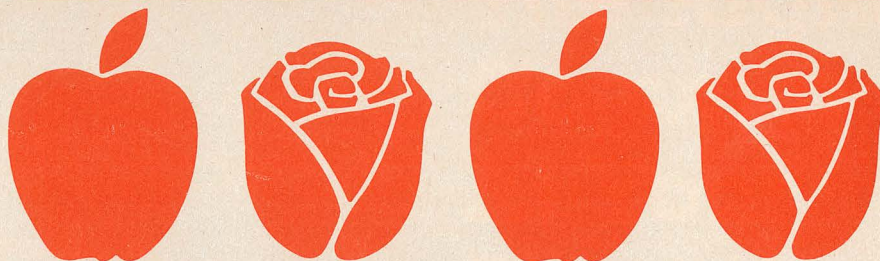
THE PERSONAL COMPUTER USERFEST

Plan now to attend the Personal Computer Userfests, the largest events ever . . . for Apple and IBM PC users.

Userfest brings together two of the largest, most successful shows ever conceived for personal computer users: Applefest and PC'83. Now that Apple and IBM can run each other's software, and with so many products adapted for both systems, the two shows merged beautifully.

At Userfest you'll see—and try out—all of the newest state-of-the-art products for your Apple, IBM PC or work-alike. Each Show has hundreds of displays and exhibits, and thousands and thousands of products including innovative new software, power peripherals, accessories, support services, books and publications. Products to help you explore the full potential of your computer for office, home and school applications.

Userfest features all the major makes of Apple and IBM computer compatibles. In fact, it's the largest display of these products, and biggest gathering of IBM and Apple experts, ever assembled in either city. Hence, you can learn more in two days at Userfest than you could in months of visiting computer stores or reading trade journals.

And best of all, everything on display at Userfest is for sale, usually at special show prices, so you can save hundreds, even thousands of dollars by making your purchases at the Show
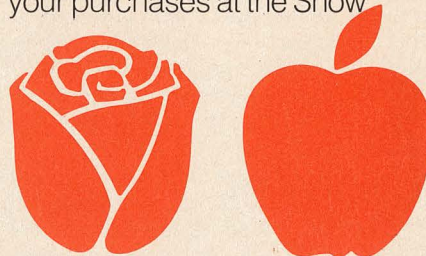
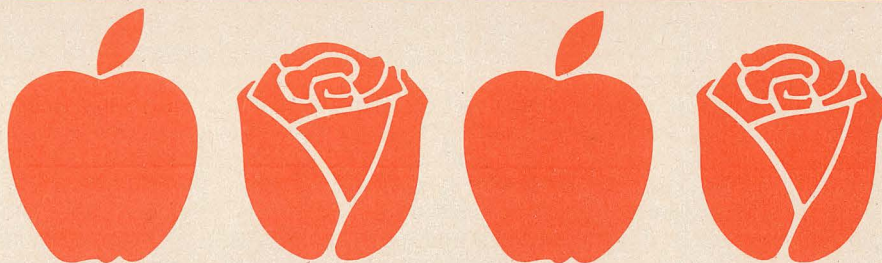So don't miss the Personal Computer Userfest when it comes to Chicago and New York in 1984. It's a once-only opportunity.

Order your tickets in advance and avoid long lines. Admission is $10.00 for a one-day ticket, or $20.00 for four days. Children's tickets (under 10 years of age) are $4.00 and $8.00. If you need hotel accommodations and/or airline reservations, check the line on the Advance Ticket form.

FEATURING IBM, APPLE & WORK-ALIKE COMPUTERS & COMPATIBLES

THE PERSONAL COMPUTER
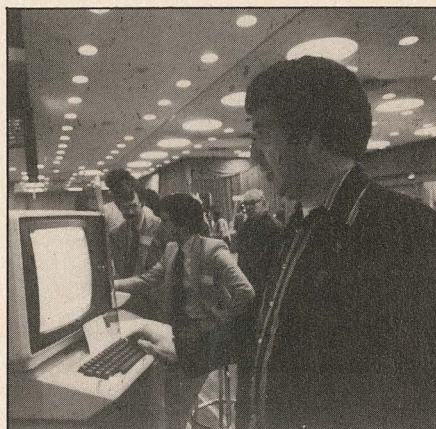# USERFEST

USERFEST/ CHICAGO

Thursday-Sunday
May 3-6, 1984
10:00AM to 5:00PM daily
O'Hare Exposition Center
9291 West Bryn Mawr
Rosemont, Illinois
(next to Chicago's O'Hare Airport)

USERFEST/ NEW YORK

Thursday-Sunday
September 20-23, 1984
Madison Square Garden
10:00AM to 5:00PM daily

For information about exhibiting at the Personal Computer Userfests, call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000.

For hotel information call or write Trade Show Department, Fox Travel, P.O. Box 498, Waltham, Mass 02254. Tel: 617-890-1770 or 800-225-8410 ext. 314.

Userfest (formerly known as Applefest and PC'83) is produced by Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167.

ADVANCE TICKET ORDER FORM

Mail this form (or a facsimile) with full payment to Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000. No ticket orders accepted 14 days or nearer to each Show. Your tickets will be mailed one month prior to the Show. Sorry, no telephone or credit card orders please.

Name:_____

Company (if any):_____

Address:_____

City:_____ State:_____ Zip:_____

Tel: Day (_____)_____ Evening (_____)_____

Enclosed is full payment for:

_____ adult one-day tickets @ $10.00 each   _____ adult four-day tickets @ $20.00 each
quantity   quantity

use this line for children's ticket orders

☐ Check here if you need hotel and/or airline information

Capps: I claim that, if he's smart, your average fourteen-year-old kid with Instant Pascal, *no* books, and good sample programs can teach himself how to program.

Horn: That's right. In a week.

ST.Mac: And the reason is that you don't feel like you're programming. You feel like you're *editing*. But assembly language is hard to learn, right?

Horn: No, it's not. You just play around with it, you learn the tricks, you learn what you can get around, and you just do it.

Capps: But the point is, if you don't learn it in six months, go to another field. My best friend in high school and I both got excited about computers at the same time but he just didn't get it. So I did all his assignments for him. And now he's running a circus—literally. He just didn't understand it. There's a certain brain that can't.

ST.Mac: Now let's do your history. How did you get involved in computers?

Horn: In the sixth grade, this guy, Nils Nilson, who's at SRI [Science Research Institute, a software think tank], brought over a terminal to the elementary school. And I got a little involved there; it was fun, I learned Basic....

ST.Mac: You were how old?

Horn: Eleven or something. After that, when I went to high school, they had HP computers, and I played with them, and programmable calculators.

ST.Mac: Modesty aside, you were a whiz at math, right?

Horn: Yeah, I love math. But Ted Kaehler came down to the school, and he wanted someone to do some work for him, so I went up to Xerox.

ST.Mac: What kind of work? You were only fourteen, right?

Horn: I guess. I was supposed to work on Smalltalk, and I did for a while, but then I got into writing assembly code for the Alto.

ST.Mac: Did this knock everybody out at Xerox? That a fourteen-year-old was writing assembly code for the Alto?

Horn: I don't know. There were lots of smart kids around. I worked at Xerox for a long time. It was neat. The Xerox group was really great. That's where I learned a lot of my stuff.

ST.Mac: What was the Notetaker?

Horn: I did a lot of the ROM for that. Other people worked on it, but I kind of felt like it was mine. It was a box about the size of an Osborne—this was before Osborne—and it had a screen about....

Capps: Bigger than the Osborne.

Horn: By a long shot. And a mouse, and a touch-screen, and everything, and it ran Smalltalk. And you carried it around!

ST.Mac: This was when?

Horn: Nineteen seventy-eight? Seventy-nine?

ST.Mac: How did the Xerox discoveries influence Macintosh?

Horn: The mouse. Windows. Selecting. Overlapping windows....

Capps: You've heard the "Bill [Atkinson] story." That he went there [to Xerox] and thought they were doing arbitrary updating [of screen images—a tough technical feat], but they weren't.

Horn: But they are now. I remember being at Xerox when the Apple crowd came in and saw Smalltalk. And they all saw Smalltalk, and I guess what happened is that Steve Jobs said, "I want that!" And at that point, the Lisa had function keys along the top, and it was totally different. And I guess Bill brought in all the ideas from Xerox. You know, *all* that stuff is Xerox!

ST.Mac: So what does QuickDraw [Bill Atkinson's graphic routines, seen on Lisa and Mac] do that Xerox hadn't figured out?

Horn: QuickDraw is a lot of things. Mostly, it's regions. Arbitrary [graphics] regions as a basic data structure.

ST.Mac: Xerox hadn't figured that out?

Horn: Xerox hadn't because they didn't really *need* to. They had stuff like that.

Capps: But they didn't have regions.

Horn: They had things *like* regions. It depends on what you think regions are. The point is that regions are unique to Mac and Lisa. That's what Bill did—and along with that, he added all these incredible things you could do. Bill took a lot of things, like ovals, and rectangles, and round-rectangles, and made *them* real citizens, you know. Ovals are superfast because he optimized for *that*, and all these other things. And he built that into QuickDraw.

ST.Mac: Steve, there's a story about you finding an Alto in a closet. Apocryphal?

Capps: Well, I was working at Xerox and we had an Alto in the closet. It wasn't a janitor's closet, it was a storeroom. Anyway, you had to go into this hot, stuffy room. And I brought it up, and it was fun. The first thing I discovered [on the Alto] was BitBlt [a graphics routine for high-resolution graphic manipulation]. This was when I was halfway through college, working at Xerox in Rochester, New York. So, instantly I got exposed to all this good stuff, at least what little PARC would let out. We were kind of like the black sheep. They wouldn't give us all the good stuff, but they'd give us a lot.

ST.Mac: Everyone at Xerox is hooked up onto a network of computers. Do you miss that?

Horn: I miss networking for sure. And I miss the Xerox environment. But I think we [at Apple] can build up something just as good—we just have to do it, and it's going to take a lot of work.

Capps: I think that, for the person who's not exposed to a computer, networking is harder to understand than the mouse. There's this head that says that if they've never used a mouse, then they'll pooh-pooh it. But networking, I think, is even more so than that, because...well, I just happened to be very, very lucky to fall into it at a tender, early age.

ST.Mac: It's quite conceptual.

Capps: If I tell you, "Yeah, you can just sit in your office and send a memo to Bruce in California," you'll say, "Oh, that's neat!" But you don't realize the power of that, and how much fun it really, really is.

Horn: You can leverage off it. It's not only fun. If you need an answer to a question you can—boom—find out in an hour, from anybody, anywhere. If you want to send mail, you can send mail. If you want to get source code, you can get source code. You can pull anything around.

ST.Mac: It's a quantum leap in brain amplification over using a stand-alone computer.

Horn: That's right, absolutely.

ST.Mac: The pressure of getting this software out must have been enormous. How did you cope with it?

Horn: We didn't. We just went crazy. We were sitting in the little cubicles there and we'd pop vitamin C pills all the time. We'd go wacky occasionally and just start babbling around, going crazy.

Capps: Video games.

Horn: His outlet was video games; I don't know what mine was. Screaming, I guess.

ST.Mac: Were you dreaming about code?

Horn: Oh yeah, all the time, all the time. Still happens. What's even more interesting is that when you do something different, like when I went skiing—and it was the first time I'd been out of Apple for more than a couple of hours, except for sleeping—it just purges: I had terrible dreams all night, as it purged from me. It was interesting, seeing all that come out. I'd just held it all in until the end, and it had to come out.

Capps: You get used to it. Andy [Hertzfeld, another Mac programmer] used to come up to us and ask how we stood the pressure. And once you've crossed the threshold, there *is* no more pressure.

Horn: You're just in a daze from doing it.

ST.Mac: How many hours a week were you working?

Capps: Easily over eighty.

Horn: The last couple of weeks, we were there all the time.

Capps: The last two days we basically stayed up forty-eight hours straight. Bruce'd sleep a couple hours....

Horn: Steve'd play video games. He stayed up the whole time.

ST.Mac: No caffeine?

Horn: Neptune and Pepsi. And vitamin C.

Capps: Everybody around us was getting sick, and I haven't been sick for I don't know how long. I attribute it to taking thousands of milligrams a day.

Horn: We'd pop 'em like candy.

ST.Mac: Steve Jobs gets a lot of criticism. How was he as a boss?

Capps: He was great. I wouldn't trade him, and all his faults, for any other boss I've ever had.

ST.Mac: Why?

Capps: Just because...this is where the action is. I mean, I could be back at Xerox, with normal blood pressure and a house and 2.1 kids, and I'd die. But out here [at Apple], you can die at thirty.

ST.Mac: And have a lot more fun doing it.

Capps: And have a *lot* more fun doing it.

LOOKING GOOD!
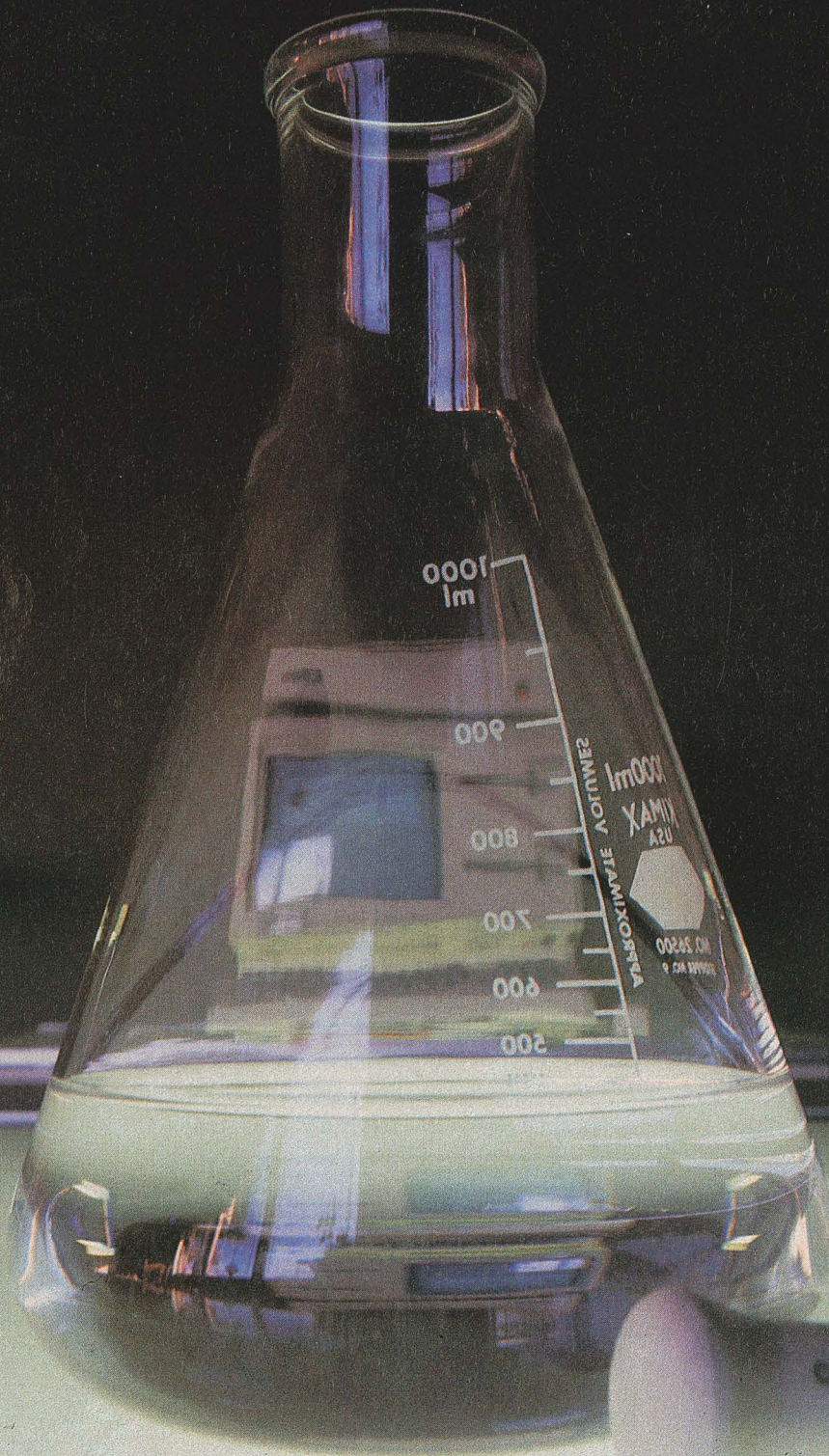
Lisa Accompanies McKesson Into a New Era of Growth

BY MATT YUEN

Lesson one of business basics: Make as much money as possible while spending as little as possible. Chapter one of that lesson says to give customers what they need at a price they'll pay.

Inside a thirty-seven-story building at the corner of Post and Market streets in San Francisco, McKesson Corporation is doing the quiz at the end of chapter one. One of the questions on the quiz is "How does a 150-year-old corporation use technology to increase productivity?" McKesson wrote on the answer sheet: Computers.

Having established itself as one of the nation's leaders in marketing, McKesson feels that its biggest challenge is to increase productivity. In meeting that challenge, McKesson is finding ways to deliver its goods and services to customers quicker and at lower cost. In all of McKesson's distribution businesses (drugs, chemicals, liquors), computers are clearing the way toward that goal.

Big mainframes handle the data processing, electronic order-entry, and computerized delivery schedules. But there are some tasks in every corporation that the giant computers just can't handle. For the little things that have big effects, for the small tasks that a large tool can't handle, there are microcomputers. And at McKesson, the new microkid swiping all the attention away from the veteran IBM PCs is Lisa.

They Needed Each Other

The teaming of McKesson with Lisa was a two-way effort. It started back in September 1982, when some of McKesson's upper-level managers were invited to a sneak preview of Lisa. Apple was considering a variety of corporations as potential beta test sites for Lisa. At the same time, McKesson Drug was looking for something that would give them the graphics capabilities they needed for publications and presentation material.

"We looked at the market," says Leslie Mastalarz, editor of sales publications, "and there wasn't anything on large computers that could do what we wanted. There were some packages that would produce a graph or a chart; and at the other end of the spectrum were the very expensive CAD systems." Other than Lisa, there wasn't anything on low-end computers that offered the friendly graphics system McKesson needed.

From the front door of the McKesson building, one need walk no more than a block or two in any direction to find at least a dozen other corporations that could use Lisa effectively. To understand why Apple picked McKesson as a beta site, take a look at what makes it one of the nation's leading wholesale distributors.

Time To Reorder Cat Mouthwash

Go into almost any chain retail drugstore and look at the drug shelf labels. They have the names of the products on the shelf, some odd-looking numbers, and maybe a computer bar code. Those labels are McKesson Drug's way of computerizing the task of taking inventory. Instead of having to look at and count each item in the store, a clerk can just scan the shelves with a hand-held Econoscan, a device that figures out how many of each item should be ordered. In addition, stores can use Econotone, a system that allows them to transmit orders directly to McKesson Drug's computer center via Touch-Tone telephone.

Two of McKesson Drug's newest services to retailers are Econoclaim and Econoprice. Econoclaim is a computerized program that helps pharmacists speed up the process of preparing prescription claims that need to be sent to state agencies or insurance carriers for third-party reimbursement. Econoprice is a service that helps pharmacists implement pricing strategies for prescription medications.

Two things happened to McKesson as a result of its elaborate computerized services' success; its wholesaler share of the pharmaceutical market grew to more than 50 percent, and it got the company noticed by Apple as a possible testing site for Lisa.

McKesson Drug's use of computers in retail drugstores was an innovative electronic order-entry system. Recognizing that, Apple
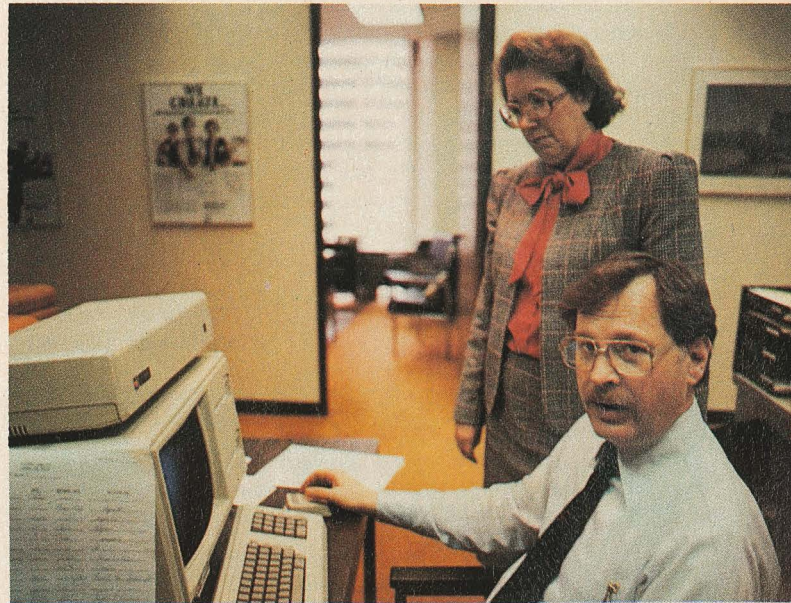
Who's watching whom? Technical support manager Kent Johns monitors McKesson Drug's electronic order entry system, while Leslie Mastalarz appears to be monitoring Johns.

approached McKesson Drug's sales department and said, in essence, "You're the kind of computer people who would be able to take big advantage of Lisa."

Apple's sales pitch wasn't necessary. By the time Apple asked the company to be a beta site, Mastalarz had long been considering implementing personal computers in the sales department. Two Lisas were installed, and a third was added shortly after.

Lisa fever spread through the drug company's sales department; other department managers saw the kinds of things Mastalarz was doing with *LisaWrite* and *LisaDraw*, and jaws hit the floor. They recognized immediately the kinds of charts, graphs, and presentation materials they could create with Lisa.

When you have something like Lisa that's possibly the most fun office tool since the push-button telephone, word gets around fast. McKesson bought some more. And then it bought some more again. Today, McKesson Corporation has nine Lisas, including one at its pharmaceutical card services department in Phoenix.

There Wasn't Always Air Conditioning, Either

With the sales department's Lisas bogged down from 6:30 a.m. to 6:30 p.m., it's hard to imagine how things ever got done there without them. Mastalarz points out that nothing new is being created

with Lisa; it's the same material they've always produced, but it looks better and is generated faster.

Mastalarz understates the changes Lisa has made in the sales department. For example, in the old days, sales management would draw diagrams of ongoing projects about once a year, usually at the midpoint, to show how much progress had been made and how far there was to go. Decisions on when various phases of a project were to begin were made mostly on the gut level. Not anymore.

"With *LisaProject*," says Mastalarz, "we're inclined to diagram projects up front to justify programming expenditures, plot time frames, and show when certain programs are to go into effect. Now we submit a project to the boss, and if he approves it, the project diagram is photocopied and included on correspondence; we're substantiating things more."

And *LisaProject* is the program McKesson Drug uses the least. *LisaWrite* and *LisaDraw* are used the most, and they're probably appreciated the most, too. One of the first tasks Mastalarz tackled was getting rid of loose pieces of paper that sales representatives had to carry from store to store.

Ten different services meant ten different contracts that sales reps had to keep organized. In addition to contracts, there was a load form (a form containing customer data that the sales rep hands over to McKesson for data entry) that had to be completed for each customer. Fumbling with separate sheets of paper for each service a pharmacist might want to buy (and risking the embarrassment of dropping everything onto the floor) is something Mastalarz wanted to help representatives avoid. The way to do this was to combine load forms and contracts into one pack. It was a noble idea; the typewriter, scissors, and tape were standing at attention, ready for work.

Lisa, the Miracle Drug

At the end of a week, Mastalarz had completed two contracts. Working at that rate meant the entire task would take just over a

Number-cruncher Dave Malmberg sits content with his Lisa.

month. Then came Lisa. "Four or five hours." That's how long it took Mastalarz to combine ten contracts into one consolidated contract, with the load form included.

With Lisa's dot-matrix copy in hand, Mastalarz could either duplicate the copy of the contract for use in the field or send the copy to a typesetter, who would retype everything and produce a sharper-looking version. An even better approach would be to connect Lisa directly to a typesetting machine to produce a contract of typeset quality. That's exactly what Mastalarz does now.

Last October, Compugraphic introduced its Personal Composition System, a package that consists of a Compugraphic output unit, the Lisa, and software. McKesson Drug was already using a Compugraphic unit for its in-house typesetting, and when Mastalarz found out that she could interface the Lisa with one, she knew it was just what the department needed.

"Being able to get typeset output of Lisa files means I don't have to worry about a typesetter keying information incorrectly," says Mastalarz. "I know that each word and number is correct, and that it will be that way when it comes out of the Compugraphic unit."

Before getting the Lisas, Mastalarz typed portions of contracts and sent them to a typesetter, who retyped them and turned out the final versions to be printed. A sample contract that took a few minutes to create with *LisaDraw* and just eight seconds to print with the Compugraphic system would have cost between sixty and eighty dollars if it had been sent to an outside typesetter. Because of the time and money involved, McKesson often sidestepped going to a typesetter entirely. With the Compugraphic system, the sales department will be able to produce typeset-quality reports quicker and cheaper.

"A key feature of the Compugraphic system is that I can print letterings in different colors," says Mastalarz. If she wanted certain parts printed in red, normally she would send the typeset copy to the printer for color separations. With the Compugraphic system, Mastalarz just cuts out the fields that are to be in color, holds them on the Lisa clipboard, and sends the rest of the copy through the Compugraphic unit. Then she brings back what's on the clipboard and sends that through.

"When we send out the two pieces of repro, we've done most of the printer's work already," she explains.

Thank You, Mr. Gutenberg

Some McKesson Drug employees who benefit significantly from Lisa probably aren't even aware of its existence; they're the drug force in the field. The bulletins that these sales reps receive from the sales department show how well (or not-so-well) various promotions are working out. To the drug force, they're the business bible; Lisa helps write the scriptures.

In the B.L. days (before Lisa), reports were divvied up among

several secretaries who would work on one or two apiece. Alice Faith Ray was one of the laborers in those days. Using calculator and typewriter, she would button-push and key-tap away to get her share of the work done in two and a half to three weeks' time.

Today, Ray generates all sales bulletins in a book form herself, and the only pushing and tapping she does is with a mouse and keyboard. Total time required: eight days. And most of that time is spent checking with regional offices to make sure figures are correct. Sometimes Ray receives the wrong information, or prices for certain products change. With Lisa, it means pulling up the old file, correcting the information, and printing out a new copy (ugh). In the B.L. days, that meant having to retype the entire document (ugh, ugh). Clearly, one ugh is better than two.

For the drug force, receiving sales bulletins in less than half the time it took before means dollar volumes are more up to date.

The only problem with Lisa, says Ray, is that it creates a paradox.

That the machine speeds up work is without question; but when managers see how fast things are being done, they want to pile more work into the "in" basket. "With such a powerful machine, it puts pressure on you to take complete advantage of it," says Ray. Unfortunately, "management isn't fully aware of its limits."

Henry Ford's workers might have felt the same way; the work is easier, but there's a lot more of it.

Selling services is one way McKesson plans to grow during the next few years. The other way is by buying companies and making them a part of McKesson—something that the corporation has been practicing, with varying success, for quite some time.

The Bumpiest Milk Route in the World

McKesson & Robbins merged with Foremost Dairies in 1967 to become Foremost-McKesson. The merged companies' early years were shaky, to say the least, looking more like a corporate version

Typesetting on Lisa with the Personal Composition System

Lisa's dot-matrix output is pretty good. It's not as sharp and detailed as the print in this magazine, but it's miles ahead of most other microcomputer output. However, it's still dot-matrix output. Now Compugraphic Corporation, the world's largest supplier of hardware and software for computerized typesetting systems, has developed a system that enables Lisa to produce typeset copy every bit as attractive as that prepared by a professional typesetting house.

The Personal Composition System (PCS) includes Apple's Lisa, a Compugraphic typesetting machine, and driver software to accept Lisa's input. Just think of the typesetting device as a superfancy souped-up printer and you'll have an idea how PCS works. Anything you can create on Lisa, including graphics, can be "printed" through the Compugraphic system on photographic paper.

In late 1982, Compugraphic saw what the Lisa could do with graphics and recognized immediately its potential to change the way reports, proposals, contracts, letters, and other written communication were produced. Compugraphic studied Lisa and explored ways to integrate it with the company's typesetting equipment. Phone calls were made, letters written, and lunches eaten. When the dust settled, Compugraphic had signed a contract to buy $20 million worth of Lisas and software over the next two years.

"Apple doesn't know what it did," says Compugraphic representative Mike Malcuit, referring to the effect Lisa graphics will have on typesetting. "We expect Lisa to transform the typesetting industry drastically. People in the business go crazy when they see what PCS can do."

Who can use a system like this? "Everyone," says Compugraphic account executive Dianne Capps. "The Personal Composition System can be used effectively for anything that needs to be produced in quantities of two or more." In other words, PCS will come in handy for law firms, engineering departments, sales and marketing organizations, accounting firms, and any other businesses that need professional-appearing written or graphic information.

That's a broad group, and Compugraphic recognizes that. When it needed beta testers for PCS, it chose as wide a variety of companies as possible. During the early stages, Compugraphic asked its testers what kinds of things they were doing with the system, expecting them to pull out a bunch of charts and graphs created with *LisaGraph*. Instead, the testers pulled out schematic diagrams of electrical circuits, computer program flow charts, street maps, and artistry far beyond what Compugraphic had expected—everything *but* charts and graphs.

The feedback has been encouraging; orders for PCSs have been placed by such companies as Bechtel, Boeing, Safeway, Standard Oil, and Crocker Bank.

Mike Malcuit praises PCS's contribution to business graphics. "Normally, a company will have a large composition center or outside graphics firm do whatever work it needs," says Malcuit, "but sometimes deadlines can't be met when you have to send things out to be done. What do you do when you need a bar chart by Friday morning, and it's already Thursday afternoon? You do without. All the executive needs is an easy way to convert numbers to graphs.

"With PCS, graphic artists can get back to doing things the way they used to do them—as an artist does them. It's the artist who can take a regular chart and pull out the things that make it talk.

"At Bechtel I met a graphic artist who had never run a machine in her life. With Lisa, now she's making 3-D horizontal bar charts, things that she didn't think were possible."

The Personal Composition System is available with a choice of three typesetting machines: the 8216, which generates output at fifteen characters per second, the 8400 (125 characters per second), and the 8600 (225 characters per second). Most customers choose the 8400 system, which goes for $39,995. The 8216 and 8600 systems are priced at $26,895 and $61,795 respectively.

Later this year, PCS will be configured with an electronic, or nonimpact, printer, which will allow users to create output directly on sheets of paper in multiple copies, thus sidestepping the photochemical process now performed by the typesetting machine. Additionally, a summer release of software will add capabilities now missing from PCS: the ability to print from a selection of Compugraphic's own typefaces (current PCS setups are limited to using only the type styles and sizes available in *LisaWrite*), better screen resolution, and full text-and-graphics merge capability. Multiple columns, true hyphenation and justification of text (a feature of typesetting systems that is lacking in Lisa), proportional spacing, and a wider range of point sizes will make PCS more suitable for setting large amounts of body text. Compugraphic plans a no-charge upgrade to the new software for current owners of PCS systems.

If the Personal Composition System becomes as widespread as Compugraphic envisions, it will become the next office automation tool, as commonplace as the photocopy machine is today. ✒

of a boat drill than a business merger. Foremost-McKesson's operating units were left to fend for themselves, slowly losing their grip on respective market shares. Top-level executives spent most of their time seeking out new companies to acquire, only to chuck them into the bottom drawer with little effort to integrate them with the rest of the company.

Things turned around in 1976 when Foremost-McKesson began trimming the fat and getting back to the distribution business that made it a powerhouse back at the turn of the century. Slow-growing businesses with small profit margins were dropped in favor of those offering a prosperous future. The company was on its way to becoming a lean, mean business machine.

Corporate acquisition wasn't entirely eliminated from Foremost-McKesson's game plan. However, it was limited by the Federal Trade Commission in 1967, when the FTC threw a decree in Foremost-McKesson's face requiring it to get approval from the FTC before it acquired any drug-related companies.

In March 1983, that decree was lifted, opening up a whole new area for strategic planning. Ironically, one of the first things the corporation did was divest the Foremost dairies and sell its own foods group, which led to dropping the Foremost part of its name.

Over the next few years, a lot of McKesson's growth will come from expanding its current moneymakers—drug and health care, wine and spirits, and chemicals. Whenever McKesson makes corporate acquisitions, board chairman Neil Harlan and chief executive officer Thomas Drohan are the names that usually appear in the media. But the person doing all the legwork and number crunching is David Malmberg, director of new technologies.

Besides, a Calculator Can't Draw

Malmberg is well-equipped to handle the seven- and eight-figure sums that are at the root of McKesson's buying and selling decisions. The survival pack includes two Lisas, two IBM PCs, *LisaCalc, Lisa-Draw, LisaWrite, VisiCalc*, and Lotus Development's *1-2-3*. Malmberg,

who considers Lisa to be an alternative to the IBM PC, rather than a replacement, prefers Lisa because "it provides a very personal, friendly environment to work in."

"Coming from Dave," says Mastalarz, "that's a very high compliment. Dave sits on boards of software companies and tells them in a down-to-earth manner what's good and what's bad. For him to call Lisa 'user-friendly' is extremely complimentary."

Evaluating prospective acquisitions isn't much different from figuring out which car to buy. After deciding that you need a new car, you'd look at manufacturers, compare mileage ratings, and make sure the car matches your image. In McKesson's case, Malmberg tries to find the answers to such questions as: Who are the major companies in a specific industry? If it's a public company he's considering, what is its price-to-earnings ratio? What kind of market does it serve? Is it consistent with McKesson's interests? Who are the competitors?

After all facets have been analyzed, Malmberg takes to the board of directors what he thinks is an attractive candidate for acquisition and recommends it, along with what he considers to be a reasonable price.

Analyzed, *thinks*, *recommends*, *considers*, and *reasonable*. They're all words that imply judgment—judgment that's usually preceded by the examination of conditionals that start out, "What if...." Calc city.

Even though much of Malmberg's analysis involves comparing potential dollars spent against potential profits, *LisaCalc* isn't the software he uses the most. On the contrary, it's *LisaDraw*. When all the number crunching is done, Malmberg needs something to illustrate the results of his work; that's where *LisaDraw* comes in. So far, like everyone else, he's been working with Lisa's dot-matrix printout. With the Compugraphic-Lisa setup, he'll be able to get presentation-quality output and get it faster.

Still Can't Beat Rush Hour, But....

Before getting Lisas, Malmberg built his financial models with

It used to take several secretaries to turn out the sales bulletins and reports that Alice Ray now does by herself with LisaWrite *and* LisaCalc.

VisiCalc on the IBM PC. Now he prefers to use *LisaCalc*, but not because it helps him work faster.

"Lisa still hasn't sped up the work that much," Malmberg says. "I still spend about the same amount of time; it's a matter of thinking time versus computing time." Nonetheless, he often chooses Lisa because it's easy to use, and just in case he needs to use *LisaDraw*. Being able to go directly to *LisaDraw* cuts the additional time and personnel needed to prepare diagrams and graphs. In the old days, Malmberg would give the art department a rough idea of what he needed and then an artist would create it. Now he's the artist.

Releasing McKesson from having to seek FTC approval before buying drug-related companies is almost like letting a teenager loose in a record store with a credit card. Not that McKesson is going on a reckless shopping spree, but it can shop more freely, without having to run to the FTC saying, "Mom, is it okay if I buy this?"

New buying freedom means more activity for Malmberg. McKesson is acquiring companies that produce pet supplies, first-aid kits, cosmetics, and other products found in retail drugstores. Its goal is for drugstore managers to think of McKesson Drug first whenever they need something. With a tall order like that to fill, Malmberg will be quite busy.

While Malmberg scans the outside world, looking for ways to increase McKesson's hold on the middleman market, two other people look inside the corporation, helping to find ways to maximize growth. John Billington, manager of corporate planning, and Eli Ortega, manager of strategic planning, spend a lot of time figuring out how McKesson spends a lot of its money.

It Was Pollen Season; Facial Tissues Profited

In financial planning, it's hard to know where to go unless you know where you've been. Billington and Ortega look a lot at where McKesson has been. These two are the ones responsible for bringing the board of directors up to date on where McKesson's money is going and what it's doing. Dollar volumes, growth rates, profit increases, and sales volumes are part of what they consider in their analysis. *LisaCalc* comes in handy, they say, but like Malmberg, they also recognize the program's limitations.

"We told Apple that we need a more complex modeling package within a spreadsheet," says Ortega. But he and Billington agree that all packages are a compromise in one aspect or another. Ultimately, they'd probably like to see something with *1-2-3*'s integration quality implemented on Lisa for its simplicity and presentation-quality output.

What it comes down to is the same old story: "Sure, we'd like to use the most powerful software available, but is having that power worth the time and money needed to create the graphics for board presentations?" Ortega thinks not.

"We prefer to go with Lisa for two reasons. First, its ease of use makes it simple for someone who's not a computer operator. Second, we can get graphics quality that's good enough for presentation without having to be retyped," says Ortega. "Dot-matrix output from the IBM PC isn't something you hand to the CEO. Lisa's output is also dot-matrix, but the quality is super for presentations."
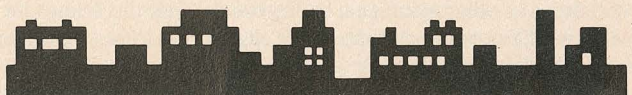
The limitations imposed by software might not be a drawback to Lisa much longer; Macintosh is part of the answer. Billington is speculating on the possibility of adding Macintosh to the current setup, not necessarily because of the machine itself but because McKesson will be able to run Macintosh software; *Multiplan* comes to mind first.

Creating charts and graphs isn't as easy as Billington would like. It's simple to take data from *LisaCalc* and generate a graph with a few clicks of the mouse, and it's simple to enhance the graph by throwing it into *LisaDraw* and cleaning it up or adding art. The problem is that graphs enhanced by *LisaDraw* become *LisaDraw* documents; it's not possible to put them back into *LisaGraph*. This means that if more data is added to the original graph, the entire graph must be modified again, not just the new parts that were added after the graph was enhanced by *LisaDraw*.

Vitamin-Fortified Computer

Some of the words that crop up a lot when talking about McKesson are growth, expansion, increase, growth, prosperity, market leadership, growth, productivity, and growth. The drug and health care division, which spearheads the corporation's growth, accounted for more than half of McKesson's revenues during fiscal 1983.

With little doubt, Lisa plays an important role in many areas at McKesson. They're not the most visible areas, but they're important nonetheless. Whether it's making charts, writing contracts, keeping customers on database, or juggling numbers, Lisa is an integral part of McKesson's plans to grow.

Mac Learns a Language:
A First Look at Microsoft Basic

BY DAVID DURKEE

Microsoft Corporation has been providing versions of the Basic language for microcomputers for years. Some have come as "store brands," like IBM Basic on the IBM PC. These versions are usually modified to suit the capabilities of the individual machines and are often pared down to an essential subset of the mother tongue to fit entirely in a limited amount of memory. And for many machines, Microsoft has offered a standard version of its Basic on disk with its own label.

Mac's designers didn't put a store-brand version of Microsoft Basic (or any other Basic) in the Macintosh ROMs. Apple hasn't commented specifically on the reason for this, but it may have something to do with the radical difference between the Macintosh's capabilities and those of other machines. In any event, Apple is working on Macintosh Basic, to be sold on disk this summer.

In the meantime, Microsoft has wasted no time in producing its own brand of Basic for the Mac. One of the first software packages available for the Mac, Microsoft Basic is not only the first Basic, it's also the first language of any kind offered for Apple's newest machine.

Linguistically speaking, a computer language is a pretty simplistic code. The human brain is capable of remembering much more "programming" than the most powerful computer. Computers haven't even begun to approach the complexity of English or any other spoken language.

A lot of people think it's the other way around. They can speak their native language beautifully, and yet they think that some conceptual block will prevent them from learning a computer language. If, because of that perceived obstacle, they refrain from learning to program, restricting themselves to running commercial software, then they're missing part of a whole new world. Programming can make the difference between using the computer as an appliance and enjoying it as an avocation.

To get an idea of where Microsoft Basic fits into the overall scheme of things, let's look at programming languages in general for a moment: what they are, how they work, and how they differ. Then we'll look at Microsoft Basic itself.

The Berlitz of Computer Programming

Basic is the language that has taught more people to talk to computers than any other. The name is an acronym that stands for Beginners' All-purpose Symbolic Instruction Code. The fact that it's for beginners is why so many people learn Basic first. The fact that it's in the permanent memory (ROM) of so many computers promotes its popularity. Being all-purpose, or nearly so, Basic has won the loyalty of a lot of people. *Beginners'* and *all-purpose* are the part of the longer name that you need to remember; *symbolic instruction code* just means language.

What Can You Say?

Given Basic, or any other language, you can tell the computer what to do. You can give it tasks as complex as you like. Performing complex tasks requires some degree of fluency in the language, but the language itself isn't complicated. The complexity must come from you. If you want to tell the computer to do something in Basic, you have to know how to do that thing yourself. You also have to know enough of the language to express the task to the computer.

To write the numbers from one to ten on a piece of paper, you have to be able to count and you have to be able to write. Those are the two components of the task: You could also think of them in terms of an ordered sequence and say that those are the *steps* required to write the numbers from one to ten.

That's how programming works: in steps. Sometimes a step tells the computer to go back and reexecute the same sequence of steps it just did (such as a step that tells the computer to count and to repeat an action ten times). Other steps will tell the computer to do something only if a certain condition is true (if, say, one number is greater than another number). This is how decisions are made by computers, from the simple kinds—like the decision to stop counting when it reaches ten—to the more complicated ones that can make computers seem intelligent.

Linguistic Skill Levels

Most of the programs you can buy for Macintosh were written in *assembly language*. The language that the computer's processor chip, the 68000, understands inherently is called *machine language*. There is a very close correspondence between machine language and assembly language. Before a program written in assembly language can be used, it must be *assembled*, or translated into machine language. Because machine language is the language that the machine understands implicitly and assembly language bears so close a correspondence to machine language, they are both called low-level languages.

Basic, on the other hand, is a high-level language. That statement can be taken two ways. Basic is closer to the language humans understand than it is to the language computers understand. It is also a program itself—like *MacPaint* or *MacWrite*—that was written in assembly language and runs in machine language.

Examine the ancestry of any program that runs on the Macintosh and you will eventually find the 68000. Programs written in assembly language and translated directly to machine language, the 68000's native tongue, will always be considerably faster than programs written in high-level languages like Basic. A program that's written in Basic looks something like this:

```
10   CLS
20   FOR X = 1 TO 10
30   PRINT X
40   NEXT X
50   END
```

The 68000, if asked to look at this program directly, will see nothing but a collection of meaningless characters. Basic, when asked to run this program, calls upon a program that is part of it, called the *interpreter*. The interpreter looks at the program one number or word at a time—more like the way a person would. It translates what it sees as numbers into what the 68000 will see as numbers. It looks at the words and calls upon machine language routines that are also a part of Basic to perform the actions called for by those words.

Interpreting is analogous to "reading" a book

3D Bridge

View of Bridge from (10.0126 ,-2.76357

Wait...

Break in 7210

List

```
50 REM *** MS-BASIC Program "3D Bri
60 REM *** Draws a bridge from many
110 DEFSNG A-Z
120 DIM XP%(220), YP%(220), T(4,4), T
130 PI=3.14159265#
140 XCENT=PI/2: YCENT=0: ZCENT=-1
150 CLS
160 XVIEW=RND*60-30: YVIEW=RND*6
170 PRINT "Wait..."
180 GOSUB 5000 'Init 3D
190 IP=0
200 RESTORE
202 READ DEN
```

Command

---

that's written in a foreign language by having a translator read it to you in your own language. There is one other way that high-level languages can be used to talk to the microprocessor. The other method is called *compiling*, which is more like having the book translated into your language and then reading the translation. Naturally, you can read the translation a lot faster than an interpreter can translate and read to you.

So it is with compiling as opposed to interpreting a high-level language. The compiled version will run a lot faster than the interpreted one. Of course, the compiled version still won't run as fast as a machine language program. A Mac running a machine language program is like you reading a book that was written in your own language in the first place; it doesn't require any translation at all. It tends to read easier and you don't have to wait for the translated version to come out.

The Language Gap

The analogy doesn't work as well when applied to programs written in assembly language. A book written in your own language is only a little easier to read than a book translated from some foreign language. A program written in assembly language runs much faster than one written in a high-level language and compiled into machine language. Both compilers and interpreters use generalized routines to do tasks in a variety of ways. It's virtually impossible to write generalized machine

language routines that will run as fast as specialized routines designed to do the task in one particular way.

The comparison between interpreted and compiled languages works so well, however, that we feel compelled to milk it for all it's worth. When you have a translator (interpreter) read a foreign book to you, you are able to start right away. If you insist on reading it yourself, you will have to wait for a translated version to be published: You have to wait for the translator (compiler) to finish the whole book. On the other hand, once the book has been translated (compiled), you can read it as often as you like or you can give it to a friend who knows the same language you do.

That's why interpreted languages are used at all, even though they are almost always slower than compiled languages: They are immediate. You can write a program in an interpreted language and run it right away. You can stop it and ask questions of the interpreter, then tell it to continue. You can see immediately if something is wrong with the way the program runs. You can stop it and fix it, then run it again immediately and see whether the fix worked.

When you write a program in a compiled language, however, it has to be translated into machine language before it can be run. This process can take a few seconds for a short program or several minutes for a long one. Then you can run it, and if you see something wrong, you can stop it and fix it. But before

you can run it again, you have to compile *the whole program* over again. This is why an interpreted language is a lot easier for a beginner to write in, which is why Basic was designed as an interpreted language.

Microsoft's Basic

If you've never used Basic before, reading this section of the article should at least give you an idea of the language's capabilities and what it is like to program in Microsoft Basic on the Macintosh. If you're experienced with Basic on any other machine, you'll see how Microsoft Basic on the Mac compares to your version of Basic.

A programmer writes a program using a part of Basic called the *editor*. Using an editor to write a program can be a little like using a word processor to write a letter. Basic is organized around *program lines*, which are sections of code that begin with a line number and contain one or more commands. These lines go together into a program listing, falling into order based on the line numbers.

Microsoft Basic places three windows on the screen: the command window, the listing window, and the output window. The *command window* is where you type new program lines or commands directly to Basic. The *listing window* displays the program listing. When you type a line in the command window, it is automatically inserted in the proper place in the listing window. If a line already exists with its line number, the new line replaces the old

one. If that line number hasn't been used before, the new line will go between the lines with the closest numbers above and below it.

The listing window can be scrolled up, down, left, or right the same as any window on the Macintosh can be. If the listing is too long to view all at once, you have to scroll up and down to see all the lines in it. If any line is longer than the listing window is wide, you have to scroll right to see its end and left again to see its beginning. Even if the listing window is extended to the full width of the screen, a program line can be too long to see all at once.

You can change a line in a program without retyping it. Just click on it in the listing window (you may have to click the listing window to activate it first) and it will be copied to the command window. There, you can edit the line and hit return to insert the new version into the listing. When you're editing a line in the command window, you can make use of all of Mac's normal editing features. You can click an insertion point anyplace and insert characters or delete them there. You can highlight a part of the line by dragging the mouse over it and then either delete the section or cut or copy it to the Clipboard. From the Clipboard, you can insert the section into another place in the same line or into a different line altogether.

When you're editing a program, each line you type is copied from the command window to the *output window* when you hit return. The difference between the line copied to the listing window and the line copied to the output window is that the former is inserted in the listing window at a location determined by its line number, while the latter appears in the output window immediately after the last thing you typed. It also appears in the output window exactly as you typed it, although it goes into the program listing with all of its commands converted to all-capital letters. In effect, while you're editing a program, the output window provides you with a sequential record of the last few lines and commands you typed.

The output window also displays the program's output, whether it's graphics or words and numbers. Graphics are positioned relative to the upper left corner of the output window and may be placed at any dot position within the window according to an X-Y coordinate system. Text is normally written to the screen sequentially. That is, it is printed at some position after that of the last thing printed: either to the right of it on the same screen line or below it on the screen. However, a call to the Mac's ROM routines allows text to be positioned anywhere by dot coordinates.

If printed output comes in contact with the bottom line of the window, the contents of the window scroll up one line. Text and graphics scroll together, and there is no way to recover anything that has scrolled off the display.

Now you know how Microsoft Basic's windows facilitate interaction between you and Basic. The editing system seems awkward when compared to other editing systems on the Mac, like *MacWrite*'s or even the system used in the Note Pad, because of its line-oriented nature. Be that as it may, it's at least as good a system, if not a better one, than the systems used by the vast majority of the Basics that run on machines other than Macintosh.
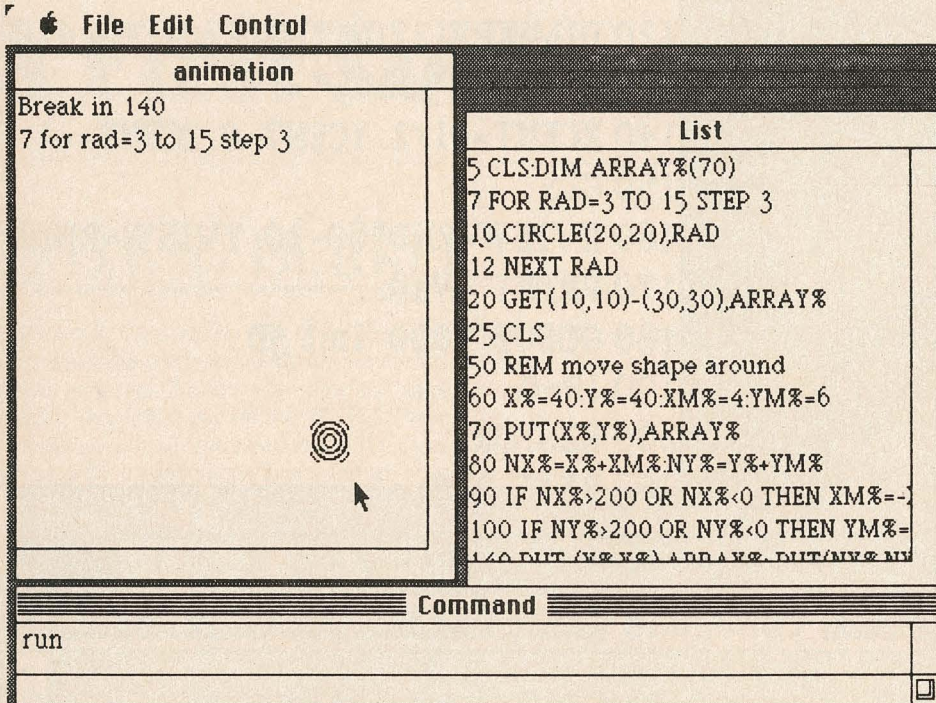
The three windows of Microsoft Basic. Clockwise from upper left: the output window (headed "Animation," the name of the program in memory), the listing window (headed "List"), and the command window (headed "Command"). Line 7 was just typed into the command window. When the programmer hit return, the command window was erased and the line was copied into the output window exactly as it was typed, as well as into the listing window—in the proper place and converted to upper case.
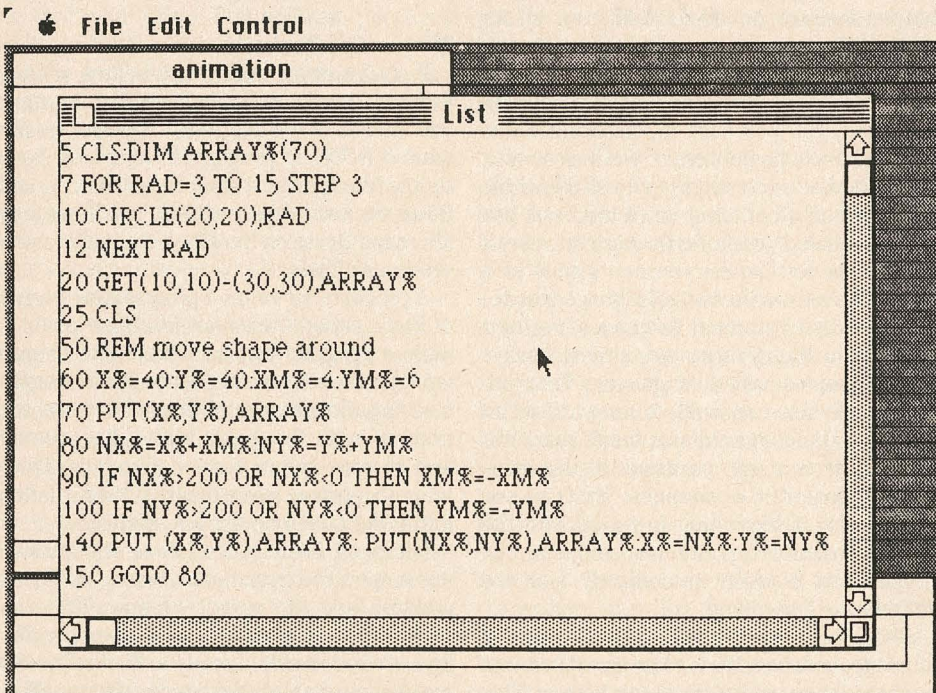
This is the entire listing of the program titled "Animation," which you can type in and run if you have Microsoft Basic. If the program were any longer, you wouldn't be able to see it all on the screen at once like this. Also, if line 140 were longer, it would run off the right-hand edge of the window. In that event, you would have to use the window's bottom scroll bar to see the end of the line.

What Do You Know?

The vocabulary of a language is what determines what the language is capable of doing. To go back to the "write the numbers from one to ten" example, if the language has no word that means *write*, there is no way it can perform that task. That example's a little extreme, though. Only a very specialized language, like one used exclusively for controlling robot arms, might lack such a command. Nevertheless, plenty of versions of Basic lack functions that you might need to do a given thing, or at least to do it the way you'd want to. Some versions of Basic have no provisions for doing graphics, which makes it hard to draw convincing pie charts.

Microsoft Basic for the Mac has a large vocabulary. It can do things that are commonly left out of ROM versions of Basic on other computers, like *if-then-else*, *while-wend*, *instr*, and *print using*. It can take advantage of the Mac's graphics capabilities with commands to draw circles, boxes, and lines on the screen and to move drawn objects around very quickly. Anything the Mac can do that isn't directly possible from Microsoft Basic can be done by calling a QuickDraw routine in ROM. Microsoft Basic has syntactical rules built in to pass parameters from the program to those ROM routines. The Basic manual refers the programmer to the *QuickDraw Programmer's Guide* for instructions on the ROM routines, although that guide isn't available yet.

Basic is a language for beginners, but beginners will probably find Microsoft Basic difficult to learn. Its documentation is a straight reference manual; it makes no pretensions to being a tutorial. Until a Mac-specific tutorial comes out, the best hope for the hopeful self-taught programmer is in tutorials written for versions of Microsoft Basic on other computers.

Microsoft Basic comes with only a few sample programs, so we threw in two of our own. If you have Microsoft Basic for the Mac, you can type these in and run them. The first program is reproduced exactly as it will appear on-screen if the window is opened large enough. The second program is too long to fit on the screen all at once, so it is presented as a separate listing.

The first program is a simple one that draws a small design in the corner of the output window (lines 7 through 12), copies part of the pictures into an array (line 20), and animates the shape (lines 60 through 150). Once you get it running, try changing some of the numbers in lines 7, 10, 20, and 60 and see what happens. The second program displays a real-time clock on the screen.
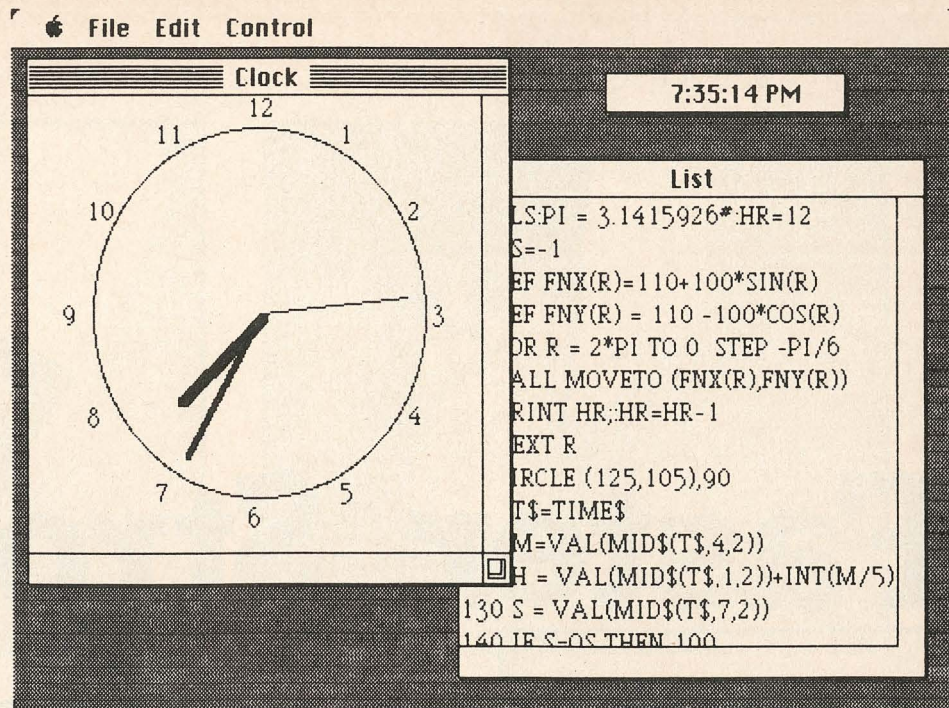
```
10   CLS: PI = 3.1415926#: HR = 12
20   OS = -1
30   DEF FNX (R) = 110 + 100 * SIN (R)
40   DEF FNY (R) = 110 - 100 * COS (R)
50   FOR R = 2 * PI TO 0 STEP -PI / 6
60   CALL MOVETO (FNX (R), FNY (R))
70   PRINT HR;: HR = HR - 1
80   NEXT R
90   CIRCLE (125, 105), 90
100  T$ = TIME$
110  M = VAL (MID$ (T$, 4, 2))
120  H = VAL (MID$ (T$, 1, 2)) + INT (M / 5) / 12: IF H >= 13
     THEN H = H - 12
130  S = VAL (MID$ (T$, 7, 2))
140  IF S = OS THEN 100
150  H1 = H * 2 * PI / 12
160  M1 = M * 2 * PI / 60
170  S1 = S * 2 * PI / 60
180  CALL PENSIZE (3, 3)
190  IF OM <> M THEN LINE (125, 105)-(125 + 80 * SIN (EM), 105
     - 80 * COS (EM)), 30
200  CALL PENSIZE (5, 5)
210  IF OH <> H THEN LINE (125, 105)-(125 + 60 * SIN (EH), 105
     - 60 * COS (EH)), 30
220  CALL PENSIZE (1, 1)
230  LINE (125, 105)-(125 + 80 * SIN (ES), 105 - 80 * COS
     (ES)), 30
240  LINE (125, 105)-(125 + 80 * SIN (S1), 105 - 80 * COS
     (S1))
250  CALL PENSIZE (5, 5)
260  LINE (125, 105)-(125 + 60 * SIN (H1), 105 - 60 * COS
     (H1))
270  CALL PENSIZE (3, 3)
280  LINE (125, 105)-(125 + 80 * SIN (M1), 105 - 80 * COS (M1))
290  OS = S: OM = M: OH = H: EM = M1: EH = H1: ES = S1
300  GOTO 100
```

This analog clock was created with the Microsoft Basic program listed above. It gets its information from the Mac's internal clock, so it displays the same time as the alarm clock desk accessory. The time for the Basic clock can be reset through the alarm clock while the Basic clock is running.

Faster, Stronger, Sleeker
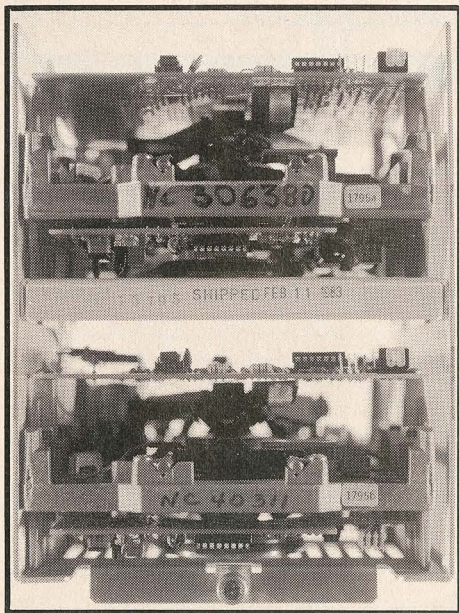
Apple's Upgrade Gives Older Lisas The Power of a New Machine

Good news travels fast, but if you're a Lisa owner and you haven't heard yet, Apple has just made you an offer you shouldn't refuse. At the same time that Apple announced Macintosh, they announced three new models of Lisa to round out the Apple 32 family: Lisa 2, Lisa 2/5, and Lisa 2/10. The new Lisas feature a Sony 3 1/2-inch disk drive that replaces the less reliable twiggy disks, a new, faster version of the Office System, and new ROMs to drive it all. The Lisa 2s will be able to run Macintosh software thanks to the Sony disk drive and a Mac emulation program Apple will be releasing.

The Lisa 2 is a plain vanilla version with one Sony disk drive, 512K of RAM, and no hard disk. The Lisa applications programs have been unbundled from the package, and the price has been dropped to a mere $3,495 for the basic system. The Lisa 2/5 ($4,495) is a 2 with an external five-megabyte ProFile, and the 2/10 ($5,495) is a 2 with a built-in ten-megabyte hard disk where the upper twiggy used to be.

Until June 1, Apple is offering a free upgrade of older Lisas to Lisa 2/5s; after June 1, this upgrade will cost $595. Current Lisa owners who want to upgrade to a 2/10 can expect to pay $2,495 until June 1 and $2,795 thereafter, although Apple does not plan to begin 2/10 conversions until May. Of course, owners of upgraded Lisas will wind up having more than those who buy Lisa 2s. They'll still have a full megabyte of memory and all the software that came with the system, and those who upgrade to Lisa 2/10 will still have their ProFiles, in addition to the new hard disks.

Lisa dealers will perform the hardware upgrade, which involves replacing the front panel, the disk drives, and an I/O disk controller ROM on the I/O board, as well as removing a resistor. At the hands of a technician who's done it a few times, the whole procedure should take about half an hour.

Updating the software will take a little longer. Before the hardware is upgraded, any data saved on twiggy disks will have to be moved onto the ProFile, because afterward there will be no twiggy drives in the Lisa to read those disks.

After the hardware upgrade is complete, the new 2.0 Office System must be installed on the ProFile and the documents on the ProFile (the user's data files, not the tools, stationery pads, or examples) must be backed up onto Sony disks. This can require as many as eight disks if the ProFile is full.

The next step is to erase the ProFile and reinstall the new Office System. This step allows all the files to be copied back to the ProFile in the new file format, which is one of the reasons the Lisa 2 runs faster than the original Lisa. Then the Lisa tools can be reinstalled on the ProFile from the new 3 1/2-inch disks provided by Apple as part of the upgrade kit.

Lisa owners will probably find that the dealer gets to do the easy part of the upgrade and that they get stuck with the tedious part. Nevertheless, getting an improved, faster Lisa absolutely free and adding the ability to run Macintosh software will make up for these momentary hassles in the long run.
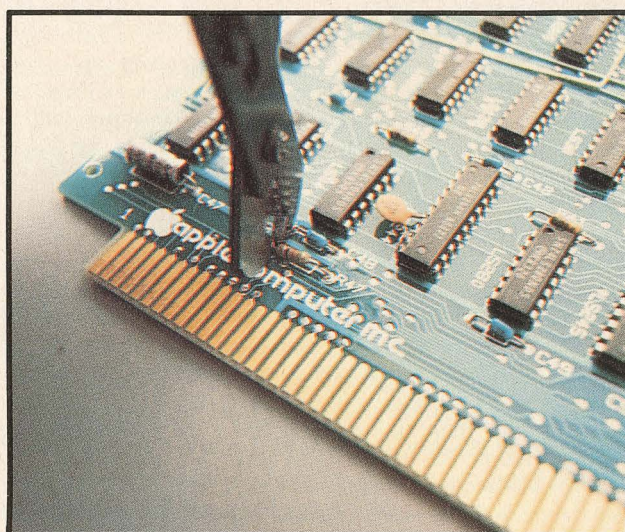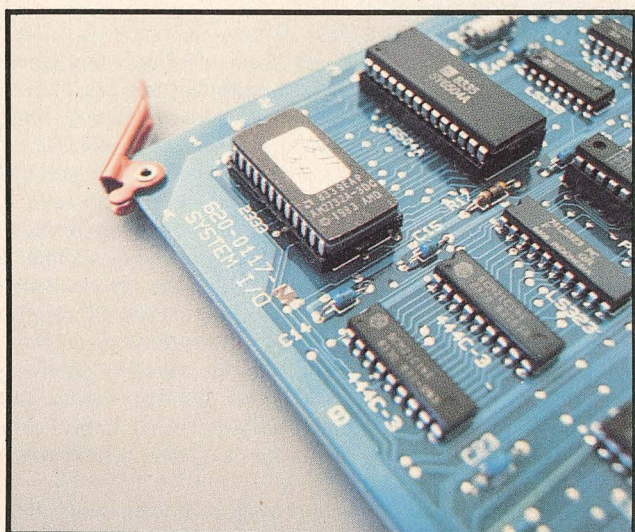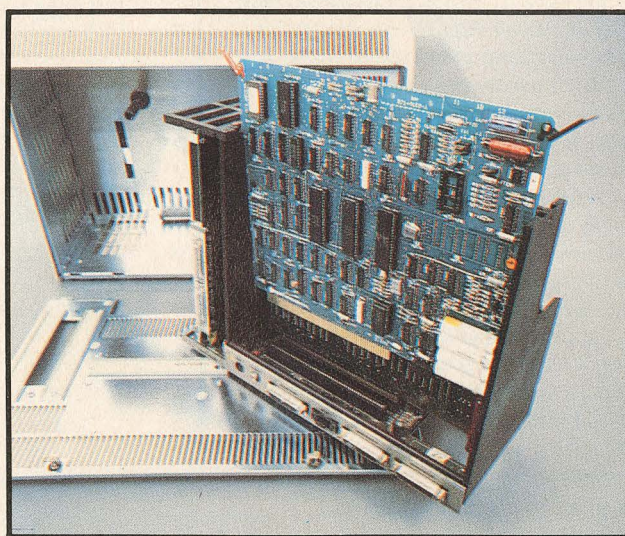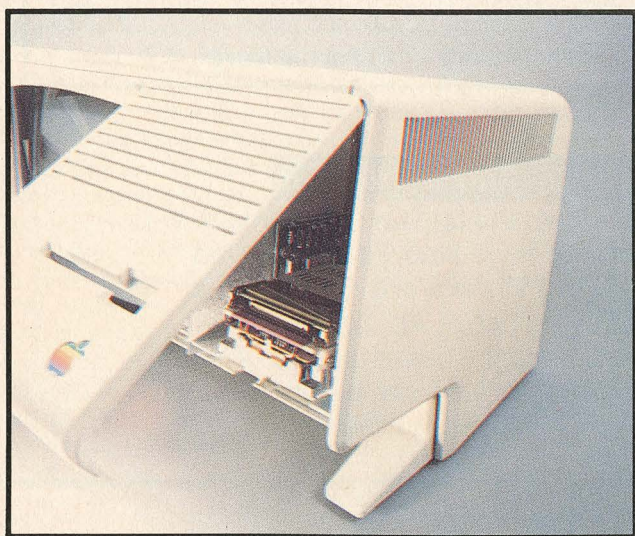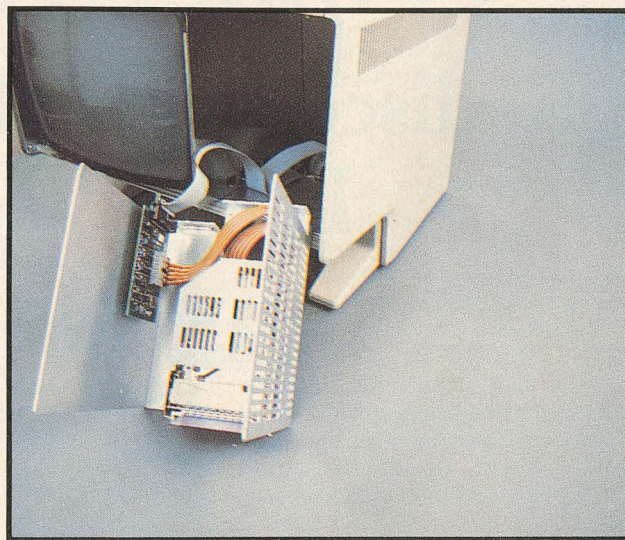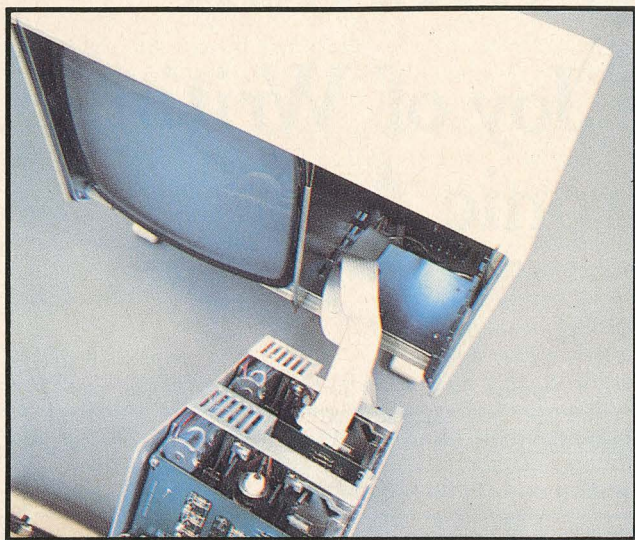
A MacWrite Memoir
Rediscovering the Joy of Writing In the Electronic Age

BY ALDEN WOODARD

A friend of mine summed up *MacWrite*. "This is the first word processor that's easier to use than a typewriter," he said.

Easier to use than a typewriter? Yes, that's about it: *MacWrite* is easier to use than a typewriter.

At first it seemed like a good idea to discuss all the various *MacWrite* features, then spend some time noting the program's limitations: the size limit of about ten pages single-spaced (or twenty pages double-spaced), the fact that a spelling checker isn't available, and other quibbles.

We'll get to all that later, but that's merely objectivity. Objectivity is prized by journalists everywhere, and objectivity is fine and good. But sometimes objectivity can be misleading. Here's a totally personal and subjective opinion of *MacWrite*: Upstairs I've got an IBM Personal Computer and an Apple II and an Apple III. Name a popular word processor and I probably have it and have used it, if only briefly. Previously, I did all my writing on the IBM; it was fast and easy, I thought. For two years I hammered away exclusively on the IBM using my (then) favorite word processor.

Now, you couldn't pay me to go back. Well, you *could* pay me, but it would take lots of money. *Lots* of money.

As I type this, the words are displayed in New York font and (let me check...) twelve-point type. I selected "Bold" from the Style menu for extra pizzazz. The letters are big and black on a clean, white background. The margins are generous. The words wrap around automatically at the ends of lines. When I end a paragraph by hitting return, the insertion point moves down and the first line of the next paragraph is properly indented.

Because editors are fussy about their manuscripts, I set the top and bottom margins (called headers and footers in *MacWrite*) five lines deep. This involved calling up the header and footer windows, then hitting return five times in each window. For the header, I also dragged down little icons that represent page number and time (a third icon is available for date: day, month, and year). When this is printed, the top left of each page will display the page number and the top right will note the time of day when this document was completed.
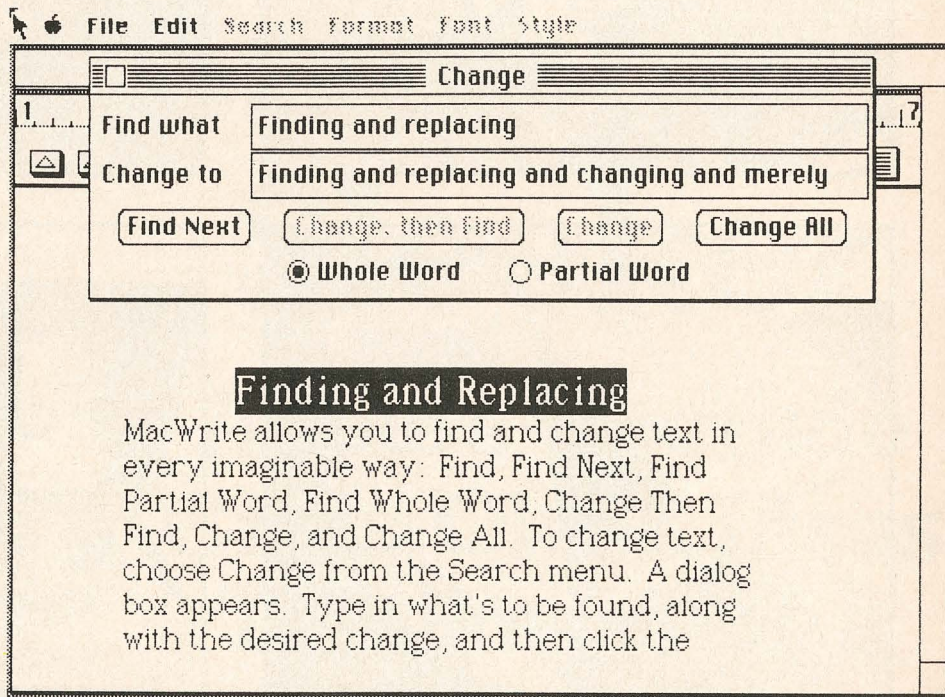
The Change command lets you search for text in your document and replace it with whatever new text you specify—pretty handy when you've misspelled someone's name throughout a letter and face the dreaded task of finding and correcting each occurrence.

But wait! I just typed into a new page. *MacWrite* displayed the page break and the header. As I watched, the "9:42 am" changed to read "9:43 am." Cute.

If cute isn't your thing, choose "Hide header" and "Hide footer" and "Hide ruler." All that's left on the screen is text. One person's cute is another person's distraction.

Getting in Form

Word processors use a variety of ways to format lines, paragraphs, and entire documents. The concept of formatting is often difficult for computer newcomers. For example, say you're using a traditional computer to create a document that's fully justified (straight left and right edges of text). Now you go back a few lines and insert a phrase. Time to reformat: The lines in the remainder of the paragraph must be recalculated and moved around to fit. For a more extreme example, imagine that you decide to change from fully justified to ragged right margins, after you've completed the document.

In both cases, a format command needs to be given: control-key-this or alt-key-that, or maybe function-key-something-or-other.

In contrast, *MacWrite* works like all word processors *should* work: When you make an insertion, text beyond the insertion point automatically shifts—fast. To reformat a document, just move the mouse pointer to the ruler and click whatever strikes your fancy. Four boxes represent ragged-left, ragged-right, fully justified, and centered text. Three more boxes offer the equivalent of single, single-and-a-half, and double spacing (actually, the amount of spacing between lines depends on the size of your characters, which only makes sense).

Decide what you want and click. The Macintosh way.

Margins and tabs are just as easy. Use the mouse to slide black triangles on the print ruler to set left and right edges of text. Another icon is used for indenting the first line in a

paragraph. Often, you'll want the first line of a paragraph indented five or more spaces. But hanging indents (where all lines in a paragraph are indented *except* for the first line) are also possible. Just drag the indent marker to the left of the "left edge of text" marker—the opposite setting from normal, first-line indentation.

Any number of print rulers can be inserted in the text, each with a different set of attributes for the text that follows. The rulers can be visible or hidden.

Entering words is a matter of typing, hitting return to end paragraphs, and using Backspace to correct errors just made.

The Mouse Arrives

More extensive editing demands the mouse. Click anywhere in the document to select a new insertion point—the point where new text will be inserted. Double-click to select a word, then press Backspace to delete the entire word in one whack. Or merely type: The selected word vanishes and the new characters take its place.

Using the mouse is a joy. Editing becomes a matter of quick clicks and slashes. If there's an easier way to edit, it hasn't yet been discovered. And don't clear your desk for room to roll; the mouse requires only a few inches of space. In a pinch, a nearby thigh will suffice as an ad hoc mouse platform.

The fundamental notion of *MacWrite*, and all of Macintosh, is this: First you select information, then you manipulate information. With *MacWrite*, selecting is done by clicking, double-clicking, shift-clicking (holding down the shift key while clicking), and dragging: the same actions used by every other Macintosh program. After text has been selected, the font and style menus can be plundered until you're giddy with power. Choose one of nine type fonts, one of five type sizes (from petite nine-point to massive twenty-four-point), and one or more style variations: plain, bold, italic, shadow, or outline.

If you wish, every line, or every word, can appear in a different font, size, style, or combination of styles. Just remember that, as Apple guru Scot Kamins is fond of noting, "The aesthetic police are watching." As always, your good taste is your best guide.

Finding and Replacing

MacWrite allows you to find and change text in every imaginable way: Find, Find Next, Find Partial Word, Find Whole Word, Change Then Find, Change, and Change All. To change text, choose Change from the Search menu. A dialog box appears. Type in what's to be found, along with the desired change, and then click the appropriate box: Change All, for instance.

Zip—done. Not a control key in sight.

Clipboarding

How about the often dreaded task of block moves—grabbing a chunk of text and moving or copying it to a new location?

No sweat.

Select the text as always, by dragging, double-clicking, or shift-clicking. Next choose Cut or Copy from the Edit menu (Cut removes the text, Copy doesn't). Either way, your text is now on the Clipboard, ready to be pasted where you wish, as often as you like, until replaced on the Clipboard by another selection.

But why stop there? When a selection is on the Clipboard, it's there until replaced by a new selection. Quit *MacWrite*, open another program, and paste away. Put your text into *MacPaint*, or *Multiplan*, or *Chart*, or Microsoft Basic. To be truly clever, use *MacWrite* to create a list of commands for The Source, CompuServe, or your favorite computer bulletin board. Next fire up *MacTerminal*. When your modem announces "CONNECT," open the Edit menu and paste an entire sequence of commands into the unsuspecting *MacTerminal*. Automation has never been easier, or more fun.

Printing

It's tough to write about Macintosh without succumbing to blathering, "Gee, this is wonderful" over and over again. When it comes time to print your document, however, the experience is, well, wonderful.

After choosing Print from the Edit menu, you're greeted with the box that begins "Quality." Pick a quality.

More boxes, more choices. Print only part of a document or print multiple copies of some, or all, of the document. (The Copies box accepts entries up to 9,999, but let's not put it to the test. Still, it seems safe to say that 6,000 to 7,000 copies of your document can be printed out, one after the other, until your printer, your patience, or your paper gives out.)

All done? Click "Okay" and a box appears to inquire about page size and such, unless you've chosen Page Setup earlier and clicked in your demands.

Next, your document is saved to disk, just as it will be printed—a considerate safety measure on the part of *MacWrite*. Your document then effortlessly and quietly streams from the Imagewriter printer. If you've chosen Draft quality, you get blinding speed but only one type font and size (Mac decides which). Standard and High print quality are "what you see is what you get," although High is a tad more gorgeous and a bit slower than Standard.

If you notice a world-class boo-boo in your document, printing can be stopped by holding down the command key and typing a period.

Your first printout may surprise you. The appearance of text is even better on paper than on the Mac screen. The reason is that gaps between dot-matrix dots "fill in" during printing.

Workstyles

Unlike other computers, Macintosh gives you freedom. Not only the freedom of fonts and styles, but freedom of workstyles.
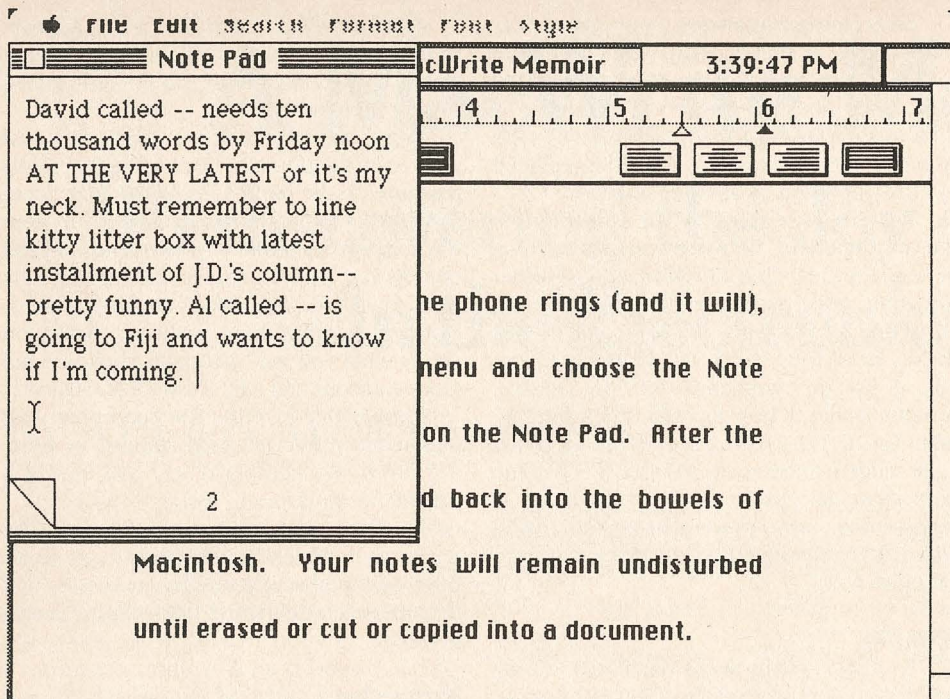
Here's one workstyle. Open *MacWrite*, then grab the alarm clock desk accessory. If you wish, set the alarm to signal the next momentous moment in your life. I set mine for lunch.

Now pull the *MacWrite* window down about a quarter inch, place the clock in the upper right corner, and click the *MacWrite* window to make it active. The clock is still visible, calmly noting the passage of time.

Begin typing. When the phone rings (and it usually does), pull down the Apple menu
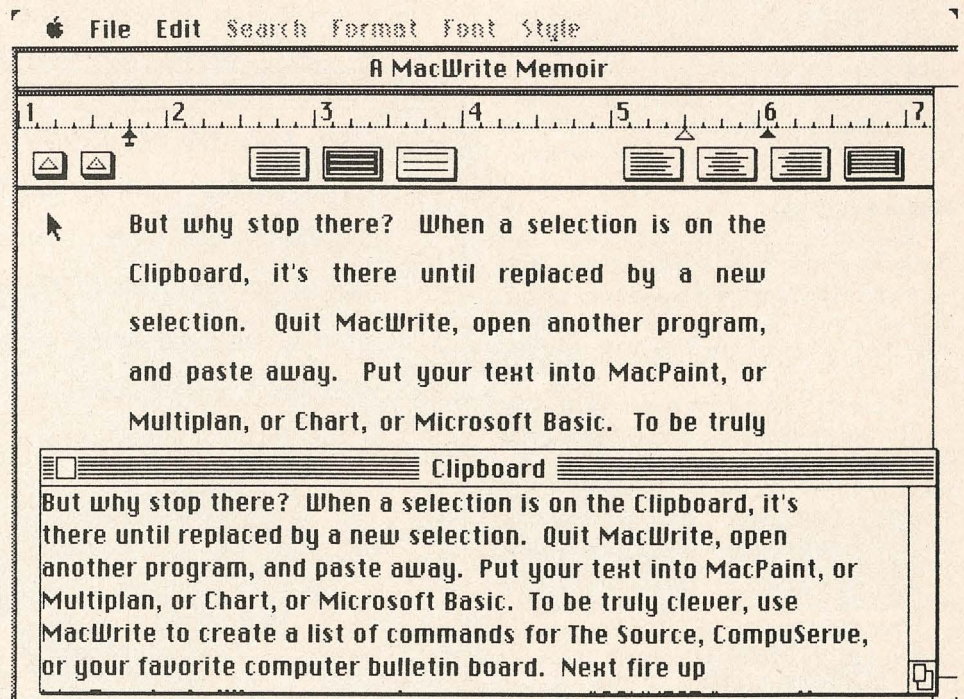
Text cut or copied from a document automatically goes to the Clipboard. The exception is text that is cut by backspacing over a selection. Whole chunks of text can be pasted from the Clipboard back into your document.

File Edit Search Format Font Style

Note Pad   cWrite Memoir   3:39:47 PM

David called -- needs ten thousand words by Friday noon AT THE VERY LATEST or it's my neck. Must remember to line kitty litter box with latest installment of J.D.'s column-- pretty funny. Al called -- is going to Fiji and wants to know if I'm coming.

2

...e phone rings (and it will),

...menu and choose the Note

...on the Note Pad. After the

...d back into the bowels of

Macintosh. Your notes will remain undisturbed

until erased or cut or copied into a document.

The Note Pad is a handy place for jotting down random thoughts as you work. Up to eight pages of Note Pad scribblings can be saved automatically in the Note Pad file on the startup disk.

and choose the Note Pad. Take some notes on the Note Pad. After the call, click the Note Pad back into the bowels of Macintosh. Your notes will remain undisturbed until erased or cut or copied into a document.

Keep typing. If you're getting paid to type, choose the calculator, another desk accessory, and determine just how much you're getting paid. If the figure is of historic importance, cut or copy from the calculator into the appropriate historical document.

Keep typing. If words don't properly convey your feelings, open up the Scrapbook and paste in something that does. I've got a picture of a fish in my Scrapbook, but if I pasted the fish into this article, *ST.Mac* would probably excise it. Editors don't appreciate fish.

Limits Revisited

About this limit on document length. Let's check: Open the Apple menu and drag down to "About MacWrite." Release the mouse button. Up pops a dialog box. It reads (among other things): Memory used: 66%. Memory free: 34%.

I can keep typing if you can keep reading.

The length limit isn't a major drawback. Few people will create documents that nudge *MacWrite*'s limit. For everyday matters such as letters, memos, reports, lists, and the like, *MacWrite* has all the capacity you need. If you're writing a novel, break it into chapters. You can live with that, right?

What else does *MacWrite* lack? Widow and orphan control, for two things. Widows are made when the first line of a paragraph appears, all alone, at the bottom of a page. Orphans are made when the last line of a paragraph (that began on the previous page) becomes the first line of a new page. Widows

and orphans are unsightly, at least in the world of word processing.

Widows and orphans mean nothing to *MacWrite*, a cruel attitude that requires adding blank lines here and there (just hit return) to produce tasteful print.

No one, of course, is ever satisfied. How about a word counter, and a speller and grammar checker, and a thesaurus? How about the ability to split the screen and view and edit two portions of a large document? The Macin-

tosh version of *Multiplan* allows split screens, and it's great. Or, how about the ability to edit two different documents at once? How about merging one document into another?

Strike that last request. It *is* possible to merge a document into another document; just place the first document—the entire document—onto the Clipboard, then open the second document, select an insertion point, and paste. Instant merge. But how about true footnote capability, not merely footers? How about a mail-merge facility for mass mailing? How about, how about, how about....
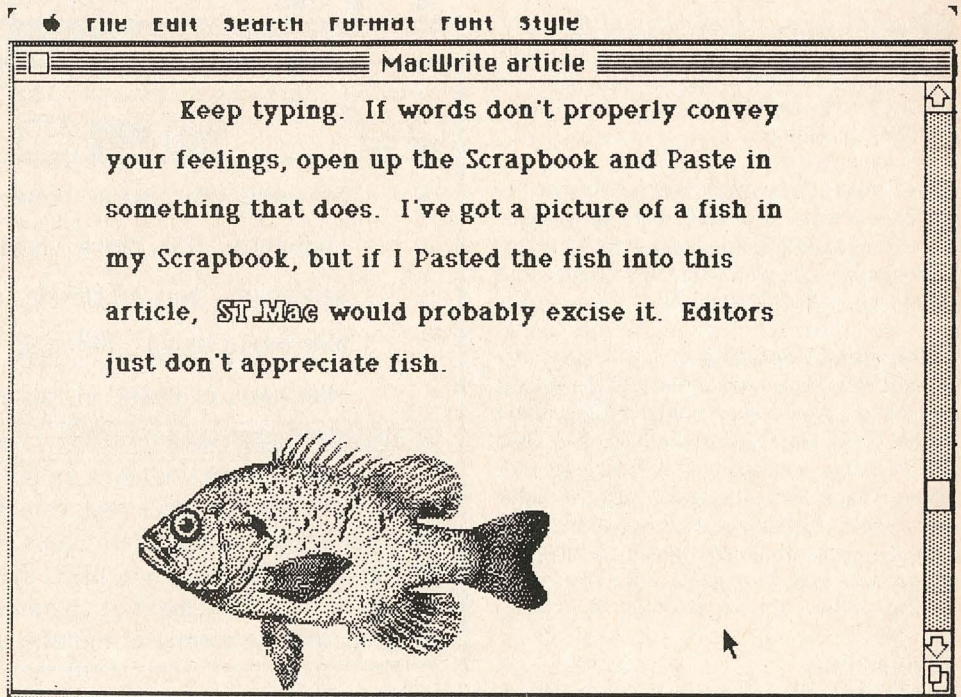
The Next Word

This isn't fair. To be honest, most people will never use all the *MacWrite* features, let alone specialized features like footnotes and mail merge. More features are usually that: more features. Features that make programs bigger, slower, harder to learn, harder to use, and more prone to bugs. And more expensive.

Apple wanted to market a word processor that the majority of Mac owners would appreciate fully and use regularly. Not one with the *most* features, but the *best* features.
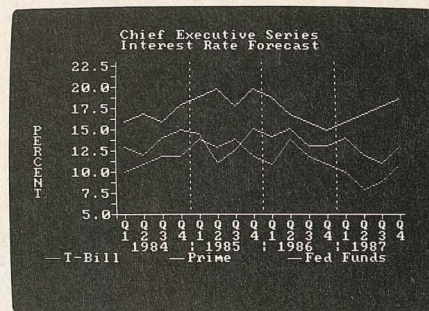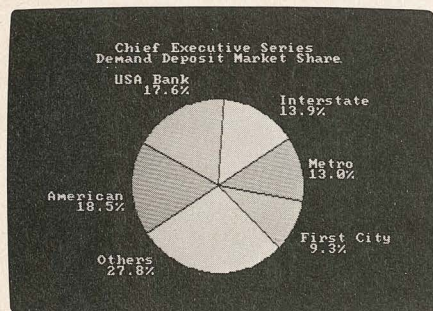
MacWrite fulfills those intentions. It is, if anything, even easier and more delightful than words can convey.

But *MacWrite* is only the first of many word processors for Macintosh. Next comes *Word*, Microsoft's elaborate word processor that, at least in the IBM version, offers everything but the ability to slice meat so thin your relatives will *never* come back. If anything, *Word* will be even more elaborate on Mac.

Word may be a socko wordsmith, or it may be a confusing dud, but it will certainly be only the second of many word processors for Macintosh.

File Edit Search Format Font Style

MacWrite article

Keep typing. If words don't properly convey your feelings, open up the Scrapbook and Paste in something that does. I've got a picture of a fish in my Scrapbook, but if I Pasted the fish into this article, ST.Mac would probably excise it. Editors just don't appreciate fish.

Your favorite illustrations can be kept on permanent display in the Scrapbook, ready to be pasted into an appropriate MacWrite *document.*

ADVENTURE
Our graphics put you there!
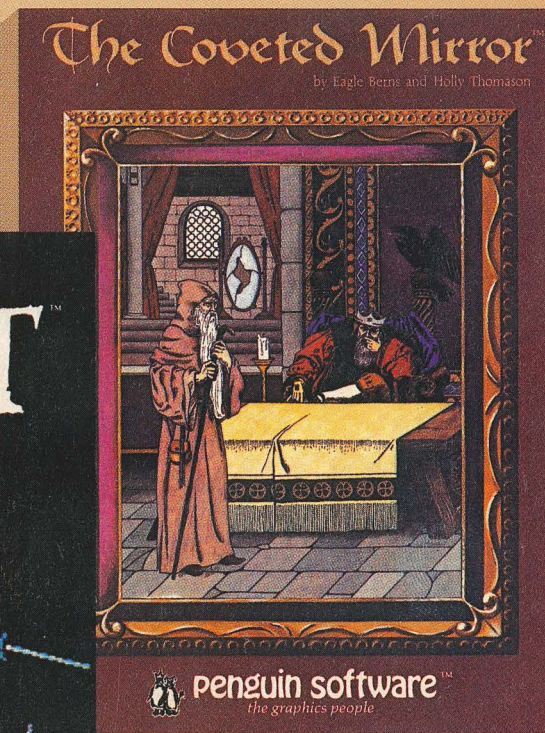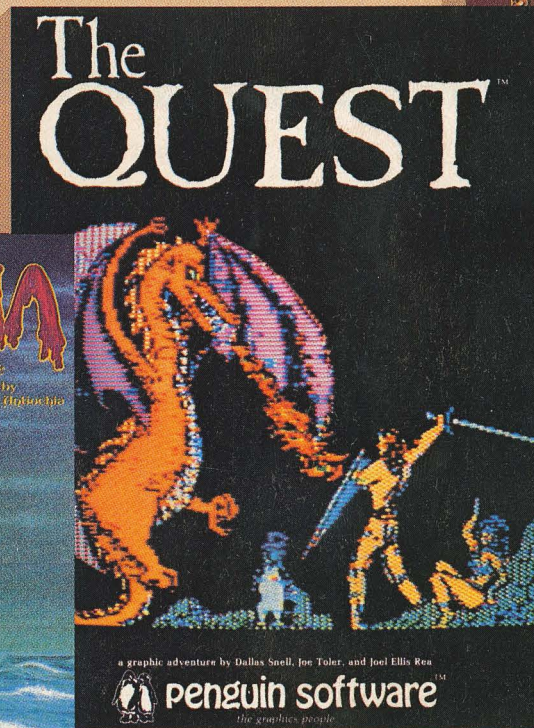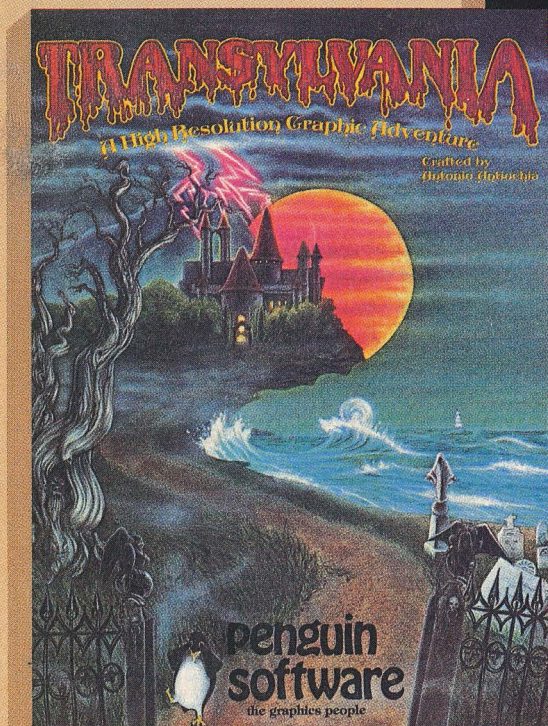
Thoughtful prose and stunningly detailed graphics by some of the best computer artists bring life to the Penguin Software adventure series. Whether in the beautiful detail of Transylvania, the expanse of The Quest, or the animation throughout The Coveted Mirror, our graphics make you feel like you're there, experiencing the adventure as you play.

The Penguin adventures are or will be available on disk for the Apple II series, Atari home computers, Commodore 64, IBM PC, and Apple Macintosh. And since they're from Penguin, price in 5¼″ disk format is only $19.95 each. Other formats may be higher. Graphics created with The Graphics Magician.

penguin software™
the graphics people

830 Fourth Avenue
Box 311   (312) 232-1984
Geneva, IL 60134