Williams = Apple = Williams = Apple = Williams = Apple = **Steve** = Williams = Apple = **Richard** = Williams = Apple = Williams = Apple = Williams Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = **Becky** = Apple = Williams = Apple = Williams = Apple = Williams = **Ana** = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = **Gerald** = Apple = Williams = Apple = Williams = Apple = **Sharon** = Williams = Apple = Williams = Apple = Williams = Apple = **Jerry** = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = **Betty** = Williams = Apple = Williams = Apple = Williams = Apple = Williams = **Jay** = Apple = Williams = Apple = Williams = **Nora** =Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = Williams = Apple = **Doug** = Williams = Apple = Williams = Williams = Apple = Williams = Apple = Williams = **Ken** = Apple = Williams = Apple = Williams = Apple = Williams = Apple =

MACINTOSH

DESCRIPTION	RETAIL	SALE
Mac the Knife Vol. 2	49.00	36.75
Mac Advantage UCSD Pascal	295.00	175.00
Kid Talk	79.95	59.95
Dinner at Eight	59.95	44.95
Video Works	99.95	74.95
Smart Alarms	49.95	37.49
Laser Serve	125.00	99.95
Mac Serve	250.00	189.95
Mac Accountant	275.00	175.00
Mac Gammon	49.95	37.49
Mac Checkers	49.95	37.49
Mac Attack	58.50	43.89
World Builder	79.95	59.95
Word Play	44.95	33.79
Math Flash	49.95	37.49
Dow Jones Straight Talk	79.00	59.00
Graphics Works	79.95	59.95
Crunch	195.00	99.95
The Home Accountant	150.00	75.00
Omnis 3	495.00	349.00
Ensemble	275.00	150.00
1st Base	195.00	99.00
Omnis 2	249.00	149.00

APPLE //

Disk Repair Kit	12.99	9.75
Quark Catalyst 3.5	34.95	25.99
Haba Merge	29.95	22.99
Super Calc	195.00	149.00
Sensible Speller	125.00	93.75
Mouse Calc	149.00	109.95
Superbase	99.95	74.95
Party Ware	14.95	11.95
Certificate Maker	49.95	37.49
Print Shop Companion	39.95	29.95
Graphics Department	124.95	93.95
Type Faces	95.00	71.95
Speed Reader	69.95	49.95

ACCESSORIES

Speed Pads	19.95	14.95
Power Director	99.95	74.95
//c Carry Case	39.95	10.95

Your one stop computer store.

Williams COMPUTER CENTER

Texas Talker BBS
367-8206 700+ Members
25770 I-45 North
Spring, Texas 77386   367-1600

Rayford-Sawdust

FM 1960

Dec. 22, 1986
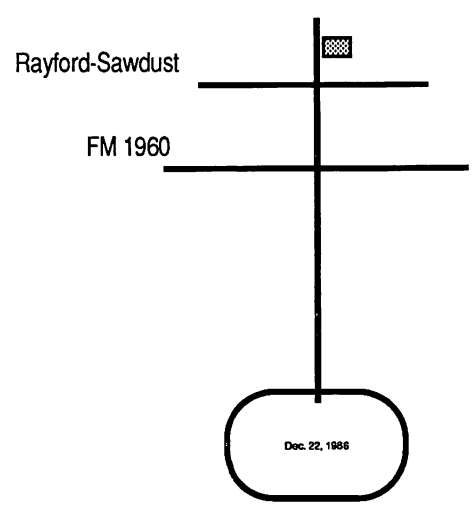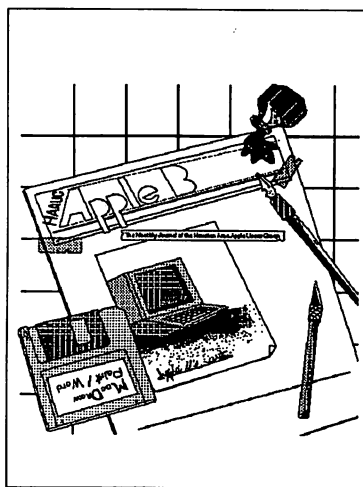
About the Cover

This month's cover repre-sents the ongoing changes to the *Apple Barrel*. Artwork was done by Chris Flick using Macro-Mind's *ComicWorks* (a.k.a. *Gra-phicWorks*) package.

Be sure to look at page 26 for "The Cover That Was To Have Been."

Contents

Volume 10   Number 1 — January 1987

Advertisers Index

Overview

The Houston Area Apple Users Group provides a forum for the exchange of ideas and information about Apple Computer products and their associated hardware and software. Meetings, newsletters, and special interest groups are just a few of the ways this forum is maintained. Below are listings of the club officials and descriptions of the groups' functions.

Club Officials

President
Robin Cox   778-1635
First Vice President
Jeanne Boucher   723-9519
Second Vice President
Mike Barnaba   527-9953
Secretary
John Marek   782-3898
Treasurer
Jack Cowart   467-4215
Directors
Cleland Early   941-7247
Chris Flick   363-3153
Jim Huck   496-9508
Mike Kramer   358-6687
George Pierce   526-5103
David Scheuer   464-1228
Senior Advisor
Rudge Allen   622-3979
Membership
Neal Scott   890-0532
Apple Barrel **Editors**
Tom Engle   774-9055
Chris Flick   363-3153
Publicity Coordinator
Dennis Bilbe   HotLine
IAC Representative
Ina Levinson   270-0137
Vendor Coordinator
Jeanne Boucher   723-9519
Program Coordinators
Mike Kramer   358-6687
Steve Bass   847-4407

The club officials meet on the first Thursday of the month at the Marriott Astrodome Hotel at 6:30 pm. These meetings are open to the membership. Members interested in the inner workings of the group are encouraged to attend.

Membership Information

A one-year membership to HAAUG is available for thirty dollars and includes the Starter Kit, a subscription to the *Apple Barrel,* and a membership card giving access to all the club's services. Memberships are available at the Answer Desk during the meeting, through brochures located at local dealers, or by calling the HAAUG HotLine and having a brochure mailed to you. Renewals are twenty dollars per year.

Meeting Dates

The HAAUG monthly meetings take place at the Marriott-Astrodome Hotel, 2100 South Braeswood at Greenbriar. The meetings are scheduled for the third Saturday of the month and begin at 11 am. All new members should attend the New Member Orientation.

Meeting Schedule

10:00   New Member Orientation
11:00   Special Interest Groups
12:00   HAAUG Club Business
12:30   Main Presentations
 1:30   Special Interest Groups
 2:30   Special Interest Groups
 3:30   Special Interest Groups

HAAUG HotLine (713) 522-2179

The HAAUG Hot Line provides a means for **members** to learn of meeting topics, news, etc, and to obtain answers to computer-related questions. If you want a return call, leave your name, telephone number, and membership number. If the question pertains to a particular commercial product, please check the list of volunteer specialists in this issue.

Special Interest Groups

SIGs focus on one topic of common interest during these meetings within a meeting. Meeting locations and times are contained in the meeting handout.

SIG Leaders

Apple II SIG Coordinator
Tom Dillon   376-6502

AppleWorks
John Slack   491-1747
Advanced Topics
Bill Hensley   980-4993
Jim Record   353-1119
Advanced BASIC Programming
Richard Lemay   749-3135
Best of the Public Domain
Robin Cox   778-1635
Education
Ann Petrillo   489-7535
Games
Thomas Carson   861-5425
John Newell   531-9139
Green Apples
Richard Goss   463-0640
Hardware
Dave Simek   496-3304
Home Use
David Jaschke   461-2450
Mike Stoops   242-1312

Mac SIG Coordinator
Steve Bass   847-4407

Communications
Mike Conway   495-2292
Desktop Publishing
Cleland Early   941-7247
Tom Engle   774-9055
Developers
John Pence
Games
Bob Lewis   242-0990
Helix
Larry Stage   358-1105
Omnis
Buddy Jacks   299-1555
Programming
Mike Martin   333-6405
Tips & Techniques
Steve Bass   847-4407

Public Domain Software Libraries

The Public Domain Libraries are divided into two sections: Macintosh and Apple II.

Macintosh Library

The Macintosh library contains over ninety disks of public domain software. Copies may be purchased at the meeting for five dollars apiece. In addition, there are two directory disks which may be copied for free. These disks contain the listings of what is on the other disks. Also, the latest disk submissions are available for copying.

Macintosh Librarian
Larry Saphier   496-0507

Apple II Library

The Apple II library may be copied at the meeting for one dollar per side. Books are available at the meeting describing what is available in the library.

Apple II Librarian
John Marek   782-3898

Hardcopy Library

The Hardcopy Library maintains an archive of back issues of the *Apple Barrel*, issues of other magazines, and other paper material. It is open to the membership and is available at the main meeting. At present, no provisions for loaning the material have been made.

Hardcopy Librarian
Les Brown   795-2741

"HAAUG Heaven" Bulletin Board System

HAAUG Heaven is HAAUG's computerized Bulletin Board System. The BBS's telephone number is (713) 664-9873 and operates 24 hours a day. The recommended protocal setting is 8-N-1-Full.

Writer's Guidelines

The HAAUG *Apple Barrel* exists to provide the membership with timely information about their Apple computers. We at the *Barrel* hope that readers will write articles on subjects such as reviews, computer techniques, and casual observations of computerdom. For those of you interested in writing for the *Barrel* (hopefully most of you), we have established helpful guidelines to follow when writing and submitting articles.

1. Try to include some form of graphic or table which helps to explain points made in the article. The saying that a picture is worth a thousand words certainly applies here.

2. For those including graphics and tables, please make references to them within your text. This will help both the reader understand what you are saying and assist us in the paste-up of the article.

3. Include references to other sources of information if possible. By doing so, the reader is directed to other useful material beyond that of the actual text.

4. For Apple II users, we will accept text created either in *AppleWriter* or in the *AppleWorks* word processor. We prefer ProDOS, but can handle DOS 3.3 as well.

5. For Macintosh users, we accept text created in either *MacWrite* or *Word*. Please do not include graphics within the actual text. Graphics accompanying the articles should be saved as *MacPaint*, *MacDraw*, or *Chart* files.

6. Articles should be provided on disk as well as printed hardcopy. The hardcopy is primarily used to allow us to see where you as the author place emphasis. (A backup copy for yourself would be advisable)

7. Article submissions can be given to either of the editors or left at the Apple Answer Desk. Include both your name and phone number with the article to help us communicate with you if problems arise.

We hope that these guidelines will not only help you but also make it easier for us to provide a more consistent and classy newsletter.

Advertising

Rates

Inside Covers	$100
Full Page	$80
Half Page	$60
Third Page	$40
Business Card	$15
Other	Contact Editors

Ads are to be prepaid, camera-ready, and received at least one week before the first of the month in which the ad is to appear. Contact either of the Editors for further information.

NEXT ISSUE

*   Cleland Early continues with more Random Thoughts and a review of *Ready, Set, Go 3.0*

*   Phil Booth reviews *MultiScribe* for the II.

*   "What the User Gets" by William Linsley

*   Chris Flick takes a look at Filter Procedures on the Mac.

*   Plus the usual SIG News (if any of the SIG people bother to write anything) and the Specialists page.

The Club Report Card

I had trouble starting this editorial. Actually, I have trouble starting all of them, but the deadline is tomorrow. I hope to get this one in on time. But what should I write about? It is the beginning of a year and therefore it seems appropriate that I should comment on the things that were accomplished during last year. Hmmm. Well, there were a lot of things accomplished, there were a few that were not and a couple of things that just got dropped.

I thought I would just jot down all of the things that were brought up and grade them. Please remember, the grades aren't for me or the board — they're for the whole club. After all, this is a group effort.

Perhaps the biggest problem of the year was not actively pursuing and using volunteers. To paraphrase a previous president, "Sometimes I'm just too damn busy chopping down trees to stop and sharpen the ax." (My apologies to Steve K.) Occasionally in the rush to get things done, it just seemed like too much trouble to help a new volunteer learn how to do it and get it off and running. I think that most of us are guilty of this at one time or another.

Now to the grades:

Coordinating and delegation (That's me) (B-): Too many late (!) night board meetings and undone tasks. Oh well, I promise to use the gavel more this year. As far as delegation and getting things done, it's hard to threaten a volunteer with a pay cut. (Chop Chop)

Apple Barrel **(B):** I have to give the editors, Tom Engle and Chris Flick, an "A" for the new magazine, but I have to give the rest of us a "C" for the lack of support in writing articles or helping in other ways. Actually, there have been some excellent articles and Jim Huck has recently volunteered (read drafted) to be the *Apple Barrel* Manager. Just another instance of chopping trees. Starting with this issue, the *Apple Barrel* will be published 10 times a year. I know, I've gotten burned on this one before, but here I am out on this tiny limb again. Be patient, we're getting there.

HAAUG Heaven Bulletin Board System (A+): With over 250 members actively participating on the system this has to be one of the big pluses of the year. The only complaint I've heard is that the system is so busy it is hard to sign on. The BBS has been such a success due largely to the efforts of Clark Johnson and Rick Oshlo

Macintosh Public Domain Library (A): This has really been a plus for the Macintosh members. The only problems I know of are the typical ones for a large library. Larry Saphier, Dick Lee and the whole gang are doing a great job. Now if we can just get the members to stop taking the catalogs.

Apple II Public Domain Library (C): We've had a lot of problems getting the II library up to date and organized. John Marek and Dave Jaschke recently volunteered to clean it up. Again, be patient.

Membership (A): Neal Scott took over for Midge Kocen at midyear, who is now spending her days in the sunny Carribean. (P.S. Midge, we miss you at pizza.) Neal has arranged for some new membership cards and a more informative Starter Kit.

Hard Copy Library (B): Les Brown has made our collection of club newsletters available at the last several meetings. This is really a treasure trove and worthwhile browsing through.

Publicity (C): We need help on this one. Steve Bass has been putting a dealership publicity group together but it needs more volunteers to get the news out. If you regularly visit a computer store then get in touch with Steve.

Overall I think the volunteers have done an excellent job. However there is still a lot of work that needs to be done. If you would like to help or tackle a new project then please volunteer. Just remember the ol' ax and be persistent.

—Robin Cox
Lumberjack and
sometime president.

Coordinating/delegation	B-
Apple Barrel	B
HAAUG Heaven	A+
Macintosh Library	A
Apple II Library	C
Membership	A
Hard Copy Library	B
Publicity	C

With 1987 Comes Changes

As might be evidenced by the cover of this issue, we the *Apple Barrel* editors are planning some changes for the newsletter. Although the cover shows only a cover in the works, I felt that the whole magazine needs some changes. I felt that the beginning of a new year was a good place to start.

First, we plan to have an issue more regularly. Put simply, we hope to produce at least eight issues this year. However, although my "All Right, I'll Write" editorial of June 1986 appeared half a year ago, neither Tom Engle nor myself have been swamped with people seeking to have their words put into print. You might be surprised how much easier it is to produce a magazine when the number of articles from which to select exceeds the space in a month's issue. When a month's deadline rolls around and there is not enough material to fill that issue, I don't feel that we should put out a mediocre magazine. I hope that you agree with this dedication to quality.

As for articles, let me say that we are not expecting full length articles from everyone. We are looking for short reviews of that new program or new piece of hardware you just bought or saw. If you don't have the time to write a half page or one page review, share a tip or two with the rest of your fellow members. In fact, a paragraph would probably more than suffice in most instances. We also would like to have gossip, rumors, puzzles, artwork, and general club and computer happenings. **If we have the material to fill a *Barrel* monthly, then you will receive a monthly *Barrel*.**

Second, as for changes, we are planning to go to a looser format. That format will be developed over the coming months. This is why we are asking for more tips, gossip, rumors, art, and other material from you. We are planning to use photographs in the newsletter as well as more art. Also, club news is a must. We would like to expand the "Software Specialists" page and would therefore appreciate your telling us about your secret software skills. We would also like to include a rumor section every month. Since we are hoping to get huge amounts of submissions, you will be getting a quality magazine each month.

Third, I think I'll stand on my soapbox for another minute or two. The production of this newsletter is a lot of work. For the past six months I have been working full time, doing some professional programming, attending each main and Board of Directors meeting and putting together the *Apple Barrel*. Now I am running for the Board of Directors with a willingness to devote even more time to HAAUG. Somewhere in there, I have the opportunity to sleep and to eat. The situation is no better for Tom. Therefore, we would appreciate your help in publishing the *Barrel* through an occasional submission. Let me close by saying that I enjoy doing the *Barrel*. However, both Tom and I are seeking your help in producing an issue each month.

*—*CHRIS FLICK

Random Thoughts

by Cleland Early

Some Musings on Language and Computers

"To create graphics that are both precise and pretty requires...a firm mathematical foundation for the features you have," reads the *Macintosh Pascal Technical Appendix*. "If the mathematics that underlie a graphics package are imprecise or fuzzy," we are told, "the graphics will be, too."

I suppose these assertions are really not that remarkable; after all, computers are at heart mathematical engines. Nevertheless, these two sentences have stuck in my mind. I guess I find it somewhat incongruous that the manual writer could care so much about the precise and pretty mathematics underlying the QuickDraw constructs and yet use such utilitarian language to describe them.

Mathematics and language are at best uneasy allies. Scientists and engineers who spend their time dealing with mathematical disciplines like computer science and theoretical physics have difficulty expressing themselves verbally. Computer scientists and theoretical physicists are not unimaginative, but their imagination is different from that of, say, A. A. Milne or James Joyce. A mathematical formula meaningless to someone like me may describe effectively to the physicist the world of subatomic particles and allow him to visualize tiny fuzzy balls whizzing through inner space.

As society has become more complex, it has become increasingly difficult for any one person to know enough about math or writing to work effectively in both fields. Society has become divided into three groups—the mathematical, the verbal, and none-of-the-above. I don't know how to help the last group, but I believe there are ways for the first to communicate well with the second.

Making Computers Accessible

Last issue, I complained that computer language manuals tended to be impossible to read and understand, particularly for the non-technical user. I find this situation troublesome because more and more programs are being written by people other than computer scientists. The great popularity of Borland's *Turbo Pascal* demonstrates that a significant number of people, most of whom are not technically sophisticated, want to write small or specialty computer programs.

Microcomputer manufacturers such as Apple are striving to make their computers accessible to those of us who are not computer scientists. The wonderful Macintosh user interface is one result of this effort. However, if computers are to be truly accessible, they must be easier to program. In this sense, the Macintosh may actually be a step backwards.

Though the idea sometimes seems strange, high-level languages were created to make computers easier to manipulate. Because users have trouble with high-level languages, some software companies have developed so-called fourth generation languages like FOCUS and Ramis II, which allow users to ask English-like questions of the computer. Unfortunately, even fourth generation languages require the user to phrase his questions in certain structured ways. They too have rules, a syntax.

For a time in graduate school, I worked as a tutor for undergraduate management information systems (MIS) classes. It amazed me to see so many otherwise bright people coming to me with blank looks on their faces, struggling to write COBOL programs. I had mixed success with my tutorees; some I helped, some were beyond help.

Most eventually experienced that light-bulb-over-the-head feeling so familiar to student programmers everywhere. Learning computer programming is a process of the epiphany, the revelation. Computer language concepts and syntax often seem so clear in retrospect that one wonders why learning them was so difficult.

Jargon and Buzzwords

I think computer languages are hard to learn because the [English] language used to describe them tends to obscure rather than illuminate. Like most mildly esoteric professions, computer science and MIS have their own language. Computer jargon can be quite intimidating to the uninitiated.

The term "jargon" is usually used in a pejorative sense, but technical jargon does have a legitimate purpose. For members of a scientific, technical, or business discipline, jargon serves as a sort of conceptual shorthand, allowing people to communicate complicated ideas economically.

Problems arise when members of some jargon-speaking group try to talk to outsiders. Jargon can act as a barrier to entry for new members. For example, computer jocks may dismiss as dense someone who doesn't know what "RAM" is. On the other hand, a person whose speech is thoroughly peppered with acronyms or other argot may be using jargon as a shield for ignorance. By using a lot of jargon, someone can sound a lot more knowledgeable than he or she really is.

In this last situation, jargon stops being useful and instead becomes pretentious. Like clichés, those familiar, overused figures of speech, jargon also can lose its power to summon up the underlying concept. When this happens, potentially useful jargon turns into buzzword. Speech and

writing filled with such buzzwords becomes dull, lifeless, and uninformative.

For the technical writer, jargon can be a dangerous trap. It's easier to toss off a phrase like "dynamic memory allocation" than to think of a good, original metaphor to describe the process.

Serious ≠ Stiff

Jargon is not the only enemy of understandable prose. Too often, writers consider "serious" (and/or "scholarly") to be synonymous with stiff. Of course, most of these writers probably don't consider the way they write a problem, nor would they call their writing "stiff." Instead, they might say they have written in a tone appropriate to their material and intention.

People whose background is in science or business are likely to consider "content" more important than "form." In some quarters, there's a curious, even perverse prejudice against good writing. Scientific writers who care about form are often dismissed as "popularizers." If ideas are good it doesn't matter how the ideas are expressed. In fact, some readers are actually thrown by writing which shows a modicum of wit, imagination, or style.

My feeling is that the attitudes suggested by the last couple of paragraphs are merely rationalizations for a general lack of writing skill in the scientific and business computing communities. It has to be kind of embarrassing for people whose heads are filled with reasonably profound ideas to be unable to express those ideas a whole lot better than the average college freshman.

The Prescription

Effective writing is probably almost as hard to learn as assembly language programming, so it's probably not reasonable to expect all technical writers to write like, say, Steven Levy, author of the marvelous book *Hackers* and my favorite computer writer. However, Levy managed to learn enough about computers to write *Hackers*; it seems reasonable to expect technical writers to learn enough about writing to make their work intelligible.

Most technical writing would be

Computer Writing: Some Selections (Good & Bad)

Two of the selections reproduced below are, according to my criteria, examples of good (but not necessarily terrific) computer writing. The other two are pretty terrible. How to tell the difference? When you read bad technical writing, your eyes will start to glaze over, and you'll have to read the selection three times before you have even a clue about meaning. A hint: the bad examples are by a couple of computer science giants.
— Cleland Early

The newest traditional languages (such as Modula-2) bend over backwards to ensure that modules hide internal routines and data structures from other modules. The goal is to achieve module independence (a minimum of coupling). The fear seems to be that modules strive to attack each other like alien antibodies. Or else, that evil bands of marauding modules are out to clobber the precious family data structures.
— from **Thinking FORTH** by Leo Brodie

The preceding chapter has introduced a formalism by which the structures of well-formed programs will subsequently be defined. It defines, however, merely the way in which programs are composed as sequences of symbols, in contrast to sequences of characters. This "shortcoming" is quite intentional: the representation of symbols (and thereby programs) in terms of characters is considered much too dependent on individual implementations for the general level of abstraction appropriate for a language definition.
— from **Programming in Modula-2** by Niklaus Wirth

The process of importing procedures is like that of using prefinished products when building a sailboat. Theoretically, you could build a sailboat completely from scratch using a hammer, saw, nails, and raw lumber. To save time and labor, though, you'll probably piece parts of the boat together from prefinished components. The prefinished components typically include canvas sails, aluminum masts, glass portholes, galley cupboards, and other items common to all sailboats and available at shipyards. In Modula-2, the supply shipyard is the module library, which contains prefinished commonly used procedures.
— from **Modula-2: A Seafarer's Guide and Shipyard Manual** by Edward J. Joyce

In practice, most of the intersections between base logical files are empty. Furthermore, the majority of the possible operational logical files have no use and, in the present state of technique , a large number of the operational logical files which would have been useful cannot be implemented. As a matter of fact, only an ideal computer would be capable of accessing all the possible sub-files and thus allow the implementation of any useful operational logical file. Consequently, several base logical files are often amalgamented through a logical union to create one operational logical file. The individual sub-file identifiers should be retained to preserve the identity of the sub-files.
— from **Logical Construction of Systems** by Jean-Dominique Warnier

better if it contained less jargon. If the writer makes an effort to avoid familiar buzzwords, he'll have to think harder about what he has to say. I think the result is likely to be clearer than it otherwise might be.

Somewhere along the line, probably in miserable English classes, most of us have learned to equate serious with solemn. We learn to eliminate any direct reference to the author or the reader. "I" and "you" are out. We learn to avoid colloquial language. We learn to excise humor. Such bunk. Most writing, even technical manuals, would be better if the writer had adopted a conversational tone. Think about programming. When you write a program, everything about it must be in order. One misplaced number or word or a missing procedure and the program won't work right. Mathematical disciplines like programming require great precision.

Writing's not always like that. Precise language is not always enough. Writing often must be evocative to be effective, and effectiveness, it seems to me, should be the goal of every writer.

More Randomness...

Winer Knocks IBM

An interesting letter appeared in the August 11 issue of *InfoWorld*. Dave Winer, president of Living Videotext (creators of *Thinktank* and *More*), praised the Apple Macintosh, criticized IBM for failing to keep its industry leadership, and said "IBM cannot stop Apple." "I don't think the marketplace will ever forgive the basic mistake that IBM made in failing to keep its technology, both software and hardware, close to the leading edge," the letter reads. Winer claimed to have sensed a new wave of enthusiasm for the Macintosh.

Those of us who attend HAAUG meetings know that this new enthusiasm is not all that new, but Winer is right that the Mac and its software keeps getting better and better. You hear less and less about how horrible the Mac is because it won't run IBM software. After all, who wants to run IBM PC software?

Winer is, I believe, a little generous in giving credit IBM credit for industry leadership. IBM has never really been an innovative company (see last month's column). IBM-created software, in particular, has always been sort of a joke.

Shatter

Remember *Shatter*, the comic book drawn on the Macintosh using *MacPaint* and printed on a LaserWriter before being colored by hand? Excerpts from the first issue of *Shatter* appeared in the June 1985 issue of *MacWorld*. *Shatter* is now up to issue number 6.

I've had a look at five of the six issues and have to say that *Shatter* is a little bit disappointing. The artwork, though pretty remarkable for *MacPaint* files, simply can't compare to the art in hand-drawn comic books. Worse, the storyline and dialog are pretty simplistic compared to other new adult-oriented comics (yes, there are such things). I'm not sure that Shatter will ever be more than a curiosity.

Incidentally, the original *Shatter* artist, Michael Saenz, no long draws the comic. I suppose he has been spending his time helping MacroMind design the nifty drawing program, *ComicWorks*.

Clancy Uses Mac

Tom Clancy, the best-selling author of *The Hunt for Red October*, uses a Macintosh for word processing. Clancy, who has made $400,000 in royalties from *Hunt*, has just published his second novel, *Red Storm Rising*. Clancy made news about a year ago when he received a three-book contract from Putnam which called for a $1,000,000 advance per book.

Who says you can't do serious word processing on the Mac?

What Happened to *MultiWrite*?

I received a copy of the Fall 1986 Icon Review Catalog the other day, and was surprised not to find any mention of *MultiWrite* in it. According to the last IR catalog and advertisements which have appeared in Macintosh magazines as recently as July, *MultiWrite* was intended to be a major improvement over *MacWrite*, with integrated outlining, the ability to open 12 documents at once, and some other neat features. *MultiWrite*'s absence from the Icon Review catalog is interesting because the word-processing software is supposedly published by MindWork Software, a sister company of Icon Review.

In the first Icon Review catalog, a small paragraph appears on the inside back cover describing something called *PreWriter*, "an integrated outliner, index card filer, text processor, and idea manipulator that is designed to greatly amplify creativity during the early idea formation stage of the writing process." The next catalog featured a full-page description of *PreWriter*, and solicited advance orders at a cost of $99. The next thing I knew, *PreWriter* is gone, replaced (I guess) by *MultiWrite*, now called a power word processor with integrated outlining. The advance-order price dropped to $79.

And now it's gone all together. I've ordered stuff from Icon Review before, and was reasonably satisfied with their service. I also like getting their catalogs. But...what's going on here? Has anyone actually seen a copy of *MultiWrite*? Has *MultiWrite* been dropped? Is *MultiWrite* vaporware or were the advertisements merely premature? Has anyone had their advance orders refunded? What about those people who ordered *PreWriter*? Icon Review's inconsistencies raise lots of questions.

Two from Borland

Borland International, of *Turbo Pascal* and *SideKick* fame, has introduced *Reflex for the Mac*. *Reflex for the Mac* is not really a new product. Borland bought *Interlace* from Singular Software and renamed it. I don't understand why Borland changed the name. Naming a product is a problematical decision, but it seems to me that in this case the negatives of the name change outweigh the positives. Interlace had received almost overwhelmingly positive notices; Borland risks losing that good will with the name change. *Reflex*, though well-regarded, is known as IBM software; does Borland really want to imply that *Reflex for the Mac* has been ported over

from the IBM PC?

Then there's *Turbo Pascal for the Mac*. I wonder what effect the release of Mac *Turbo Pascal* will have on smaller companies like Think Technologies (developers of Apple's *Macintosh Pascal* interpreter, the new *Lightspeed Pascal*, and *Lightspeed C*) and TML Systems (*MacLanguage Series [TML] Pascal*). It'd be a shame if Borland's Pascal put these other companies on the rocks.

By the way, *Lightspeed Pascal*, formerly known as *Quicksilver*, looks to be pretty neat at first glance. The programming environment is a lot like that of *MacPascal*, only compiled instead of interpreted.

FullPaint Gets an Upgrade

Saying "we listened," Ann Arbor Softworks has dropped copy-protection on *FullPaint*, their fine *MacPaint* update. In addition, they are adding color-printing capability. Regi-stered owners of *FullPaint* will receive a free, automatic upgrade. My upgrade disk showed up only a few days after I first saw the announcement in *MacWorld*. I guess that nasty note about copy-protection I wrote on my registration card had some effect.

Though I'm happy Ann Arbor Softworks did what they did, I'm not sure that their action was purely in response to user's gripes. Silicon Beach's *Superpaint* is on the way, and advance word says that *Superpaint* blows away both *MacPaint* and *FullPaint*. Besides, *Superpaint* is not copy-protected and costs the same as *FullPaint* .

Is Copy-protection dying?

Consumer bitching about copy-protection has gotten louder, and more and more companies are dropping copy-protections as a competitive strategy. Even *Time* magazine (October 20) has noted the trend. If you have any doubt that copy-protection is a dead duck, the news that Microsoft has or will remove protection from all of its business products should change your mind.

MacWorld in Hot Water?

Since Apple is now so enamored with non-disclosure agreements, I guess *MacWorld* may have angered Apple with a premature announcement of the existence of the new Apple IIGS. Everyone has known for six or eight months that Apple was planning a new Apple II, and the official announcement was made September 15. *MacWorld* spilled the beans a few days early. I received my October *MacWorld* on September 12, and it contained a picture of the IIGS and a small article describing the new computer.

The new Apple II has some pretty impressive capabilities and looks very sharp. Just think — if the new Apple II is this neat, what will the new Mac be like?

Computer Jargon and Buzzwords: A Quiz

Test your knowledge by taking the following simple quiz.

(1) The "Universal Woz Machine" is:
a. the infrared remote control unit Steve Wozniak's new company is developing;
b. the Apple II disk drive controller;
c. a nickname for the Apple I computer;
d. Wozniak's high school rock 'n' roll band;
e. none of the above.

(2) "SCSI" stands for:
a. Small Computer Systems Interface;
b. System Communications Standard Inventory;
c. Shugart Company Systems Interface;
d. Smart Controller System Isolation;
e. none of the above.

(3) In the Macintosh, a "heap" is:
a. the same as the "stack;"
b. a special type of power transformer;
c. an area in memory that may be allocated in relocatable and non-relocatable pieces;
d. a and c;
e. none of the above.

(4) A "power user" is:
a. what computer owners sometimes call themselves so they can feel more important and clever than they are;
b. a non-programmer who manipulates huge spreadsheets and programs in dBase III and Turbo Pascal;
c. someone who gets paid to use a microcomputer;
d. more than one of the above;
e. who knows?

(5) A "handle":
a. points to a pointer;
b. points to the address of a relocatable object;
c. lets you carry the computer;
d. all of the above;
e. none of the above.

(6) "VLSI" means:
a. Vacuum Limited Semi-conductor Initiation;
b. Versatle Linking and Systems Integration;
c. Very Large Scale Integration;
d. Vertical Line Scanning Interface;
e. none of the above.

(7) A "macro" is:
a. a mainframe computer;
b. a series of commands and functions that instruct the computer to take an action for you;
c. a computer chip that substitutes for several chips;
d. a and c;
e. none of the above.

(8) "M.I.P.S." is short for:
a. Mindless Idiots, Perverts, and Scumballs;
b. Millions of Instructions Per Second;
c. Microcomputer Interface Prototype Scheme;
d. Manhattan Institute of Programming Science;
e. none of the above.

(9) "Requirements Analysis":
a. is what someone considering a computer puchase should do;
b. is part of the systems development life cycle;
c. answers the question "What must be done to solve the problem?"
d. b and c;
e. none of the above.

(10) In "asynchronus communication":
a. the remote computer does not echo characters typed on the keyboard back onto the screen;
b. the sender signals the receiver each time he or she starts or stops sending information;
c. the most recent information appears on the bottom of the screen and moves upward;
d. one modem is in answer mode, the other in originate;
e. none of the above.

What makes a good Computer Term?

Computer jargon will likely always be with us; our goal should be to make new computer terms less confusing. Computer people are fond of metaphors, but often the metaphors they choose are not helpful. For example, Macintosh programmers talk about the "stack" and the "heap." These terms describe different ideas, but the images they summon up are very similiar, creating a potential for confusion. The same thing goes for "handle" and "pointer." An example of a good computer metaphor is "toolbox," used to describe the Macintosh user-interface routines stored in the Mac's read only memory. You use a toolbox to build something; similarly, a programmer can use the ROM routines to build a Mac application.

—*CE*

Answers

1. b	6. c
2. a	7. b
3. c	8. b
4. e	9. d
5. d	10. b

Have You Heard

TML tools for the IIGS

TML Systems, currently best known for the *TML Pascal Compiler* for the Mac, is readying three IIGS products for early 1987 release. They are a Pascal compiler, a source library, and resource editor. According to Nick Fotheringham of Analytical Solutions and author of *Family Origins*, a genealogy program for the IIe and IIc, *TML Pascal* code should be transportable between the Mac and IIGS. Nick, by the way, is working on versions of *Family Origins* for the Mac and IIGS.

SuperPaint is here (finally!)

A full year after having Charlie Jackson of Silicon Beach demonstrate an early version of *SuperPaint* for the Mac to HAAUG, you can now get copies of the "final" version. *SuperPaint*, if you are not familiar with it, combines the best of *MacPaint* and *MacDraw* plus a whole lot more into a single application that lists for $99, compared to about $350 for both *Paint* and *Draw* together.

SuperPaint allows multiple windows (only two on a 512K Mac or in a 512K *Switcher* partition) and provides two drawing planes in each document—one for painting and one for drawing. Features include rotation, distortion, slant, and perspective (in the paint layer), bidirectional window scrolling, color printing on an ImageWriter II, three levels of magnification, ability to read and write *MacDraw* PICT, *MacPaint*, and StartupScreen files, and LaserBits, which allows editing at 300 dots per inch resolution. The LaserBits graphics retain 300 dot resolution when pasted into other documents.

In the words of Charlie Jackson, who used to write "What's Hot and What's Not" for *Macazine*, this one's HOT! By the way, I had the honor of being a Beta tester.

New Brøderbund Offerings

Brøderbund has just released *On Balance*, a new money management package for the Apple II to help you manage your checkbook and know where your money goes. You can search by date, check number, payee, or dollar amount. *On Balance* will even remind you of when payments are due and flag all tax related items. Capacity is provided for up to 800 transactions per month and a year's worth of information can be saved on one diskette. Data compatibility with *AppleWorks* is provided, permitting data to be manipulated in the AppleWorks spreadsheet. List price is $99.95.

Where in the USA is Carmen SanDiego, a follow-up to *Where in the World is Carmen SanDiego*, is now available. These are educational geography games that let the kids have fun while learning. If *Where in the World...* is any indication, *Where in the USA...* will be very popular. List price of *Where in the USA...* is $44.95.

Brøderbund has offered the Science Toolkit Master Module for some time now. This $59.95 package includes light and temperature probes, an interface box that connects the probes to the computer through the joystick port, a manual full of experiments, and software for timing intervals, measuring temperature, and light levels and recording and graphically displaying up to 24 hours worth of data.

Two new experiment modules are now available. Module 1, Speed and Motion, allows experimentation with thrust, acceleration, win resistance, and centrifugal force. Included are photo cell and a balloon-powered car and software for measuring speed and revolutions per minute. Module 2, Earthquake Lab, includes a seismoscope, onscreen seismograph,

and seismographic plotting capability. A suggested use for this module is determination if anyone entered the room while you were out. List price for each module is $59.95.

Broderbund has also released *Toy Shop* for both the Apple II ($59.95) and for the Mac ($64.95). *Toy Shop* lets you create mechanical models of the *Spirit of St. Louis*, a steam engine, and lots of other toys by decorating them on the screen, printing them on the printer, and then assembling them.

Broderbund has released a new version of the popular *Print Shop*. It is a new Apple II version that will print in color on several color printers including the ImageWriter II. *The Print Shop* is now available for the Macintosh. As with *Print Shop* on the Apple II, it lets you create greeting cards, flyers, and banners. You can print them in either black and white on most printers or in color on the ImageWriter II. *The Print Shop* for the Mac features true WYSIWYG (What You See Is What You Get), permits use of any MacPaint-compatible graphics, and lets you place them anywhere on the document. List price is $79.95.

Accelerate with *Excellerate*

Personal Training Systems of San Jose has recently begun marketing a series of *Excel* training aids called *Excellerate*. Each package consists of a professionally done audio cassette, a disk full of *Excel* templates, and an instruction booklet. The series currently consists of:

1A   Fundamentals of *Excel* for the First Time User
1B   Fundamentals of *Excel* for the Experienced User
2   Creating Business Graphs
3   Adding Power to Your Spreadsheet

I've had a chance to review the Unit 6 and found it to be a very effective aid in learning to use *Excel*. For those busy folks who don't have the time to sit down and read the manual, this might be just what is needed. Initial pricing is $29.95 per module, although there was an indication that they would be higher later. Address inquiries to Personal Training Systems, P.O. Box 24431, San Jose, CA 95154, or call (408) 559-8635.

Turbo Pascal for the Mac

Turbo Pascal for the Mac is finally out. Among its features are:

- Compilation speed of 12,000 lines per minute.
- "Unit" structures for creation of modular programs.
- Up to 8 editing windows.
- Compatibility with *MPW Pascal*.
- Ability to use all available memory.
- "Units" for calling all Toolbox routines.
- HFS Compatibility.

List price for *Turbo Pascal* for the Mac is $99.95. It will be interesting to see what the impact of Turbo will be on *TML Pascal* sales. Personally, I'm rooting for Tom Leonard and crew since (1) they are a smaller company and (2) have been doing an outstanding job of providing programming tools while Apple and Borland just kept announcing theirs. [See Dan Podwall's article in this issue.]

DataFrame XP Available

SuperMac Technology has begun shipping standard and XP versions of the DataFrame 20 and 40 SCSI hard disks for the Mac. The XP version gives up to double the speed of standard SCSI drives. List prices for the DataFrame 20 and 40 standard version

are $1099 and $1799. The XP versions list for $200 more. Conversion kits are offered for upgrading existing standard drives to the XP version. Information can be obtained from SuperMac at (415) 964-8884.

Borland buys *Interlace*

One of the recent surprises was acquisition of the Mac relational data base *Interlace* by Borland International. For the most part, this appears to be a good thing, since Borland is well established and has a reputation for producing reasonably priced, well-supported products that are a good value. The only negative thing about the acquisition is changing the name of the product to *Reflex*, which is the name of their MS-DOS database, which currently has no resemblance to *Interlace*. Borland has stated, however, that the MS-DOS version will receive some of the features of the Mac version and vice versa.

Borland has announced a major update to *Interlace* (aka *Reflex*) which will cost registered owners $15 and their original master program disk. Since the current release asks for the original disk every 14 days, you should boot *Interlace* disk just before sending it in to time stamp your backup copy. Borland assures that there will be a 24 hour turnaround.

Some of the new features are:

- Copy protection has been removed.
- The font, font size, style, etc. can now be changed for multiple text blocks when defining forms or reports.
- Variable and text blocks can be moved as a group when defining forms or reports.
- List price has been reduced to $99.95.

Using the ImageWriter II's slot

The ImageWriter II offers a number of new capabilities that the older Imagewriter I didn't have, such as improved speed and color printing. I suspect that many ImageWriter II users do not realize that it also has a single "slot" for which three optional cards are available. When you bought the

printer, the salesperson might have mentioned Apple's optional 32K buffer board which has a list price of about $99. For a while, this was the only card available. I bought one as soon as they were available. Hopefully you delayed your purchase, since there are now more alternatives.

The second ImageWriter II card to become available was Orange Micro's Image Buffer. Listing for $99.95, this card comes with 64K of memory and is user-expandable to 128K by merely plugging in RAM chips. Since Orange Micro is a well-established, reputable company, this, in my opinion is a much better buy.

The most recently available ImageWriter II expansion card is Apple's AppleTalk Card. Listing for $139, this card permits sharing of an ImageWriter II by several computers in a Macintosh AppleTalk network without the need for physical switches. Thus it is possible to kick off a printing job on multiple computers and have them wait in a queue until their time to print. I removed my 32K buffer card in order to install an AppleTalk card as soon as I could get one. I really enjoy not having to unplug cables all the time.

So what's next? Maybe someone (Orange Micro? Applied Engineering? Apple?) will introduce a buffered AppleTalk card so the computers aren't tied up waiting their time to print. Print spooling software such as *MaxRAM/MaxPrint*, will drive the printer from the background while you continue to work, but this solution can slow the system's overall performance.

Print Sideways

For some time, Funk Software has marketed a package for the IBM that permits printing Lotus spreadsheets sideways on the paper. Now they offer the same capability for Apple users. *Sideways* lets you print documents sideways on your printer in six character sizes in single or double density. You can print documents produced by *Visicalc*, *Multiplan*, *Flashcalc*, *Magicalc*, *AppleWriter II*, *Wordperfect*, *PFS:Plan*, *PFS:Report*, *Mouse Calc*, *And Practicalc II*. With the exception of *AppleWorks* files, they must have been saved as ASCII text files, but can

be on either DOS 3.3 or ProDOS formatted disks. A wide range of printers and printer interfaces is supported. List price of *Sideways* is $60.

MacLightning spell checker

Target Software has begun marketing a new $99.95 desk accessory spelling checker that's a real winner. It boasts an 80,000 word Merriam Webster dictionary, both interactive and batch spell checking modes, and user modification of the dictionary. In the interactive mode, it beeps lightly to indicate a misspelled (I just turned it on because I always misspell "misspell") and lets you ignore it if you want. It is very quick and does not interfere with typing. It even has a phonetic mode that will tell you that the correct spelling for "fone" is "phone". By the way, *MacLightning* works no matter what you are doing. It even caught a spelling error when I was naming a new folder.

Graphics News

Ann Arbor Softworks has recently released a new version of *FullPaint*, having removed their tough copy protection scheme and added color printing

If you've been impressed by Mindscape's *ComicWorks*, but didn't see how it would help you much in your business, you'll be pleased to know that Mindscape has just released *GraphicWorks* ($79.95). It boasts the following features:

- The power of the best selling paint programs
- Full text editing capability
- Multi-page documents
- A full library of templates for newsletters, reports, etc.
- Independent, object-oriented, bitmapped graphics
- Ability to mix and edit text and graphics anywhere on the page
- Professional-quality airbrush
- Full LaserWriter support
- Custom four-fold and poster printing capabilities
- Not copy-protected.

Low Cost Power Spreadsheet

Bravo Technologies has recently released *MacCalc*, a $99.95 power spreadsheet for those who want more than *Multiplan* but can't come up with the price of *Excel*. Reviews in the magazines give it high marks for speed and features. It has a few unique features, including the ability to let each cell have its own fonts, style, and format and the ability to read files from *Multiplan*, *Excel*, and *Lotus 1-2-3*. *MacCalc* is not copy protected.

Upload vs. download

In a previous article, I wrote a glowing report about using my new NEC laptop computer as as substitute for my Mac. I used it the other day to take notes in an important meeting at work with the intention of producing the minutes. When I got ready to transfer the file into my real Mac, I inadvertantly pressed the download function key instead of the upload function key. When it asked for the file name, I typed the name of the file into which I had typed the meeting notes. Without warning that I had specified an existing file name, the little beast eradicated my 10K file. I wrote a quicky BASIC program to PEEK all the memory locations and print the ASCII character stored to the screen. I found a lot of interesting strings, such as the version number of the Microsoft BASIC in the ROM, but never did find the file. Sigh...

—*M*IKE *K*RAMER

Snowy Evening

(Poetry à la 6502)

I have been a member since the days that HAAUG and HAM had much more in common than today. And, I am now, as then and always: a hacker. Are there no hackers left — on the II at least? My offering this month is in that vein.

This program is an entertaining display of one of Robert Frost's best poems. To me this New Englander is the Sousa of American poets. The poem is properly called "Stopping by the Woods on a Snowy Evening" — too long for even DOS 3.3, so I call the program: *SNOWY EVENING (FROST)*. It is unusual in several ways besides being a computerized poem.

- It is, for what it does, very short; the program is short and the poem is short. Compared to "The Night Before Christmas" and "The Twelve Days of Christmas", over 100 lines each. Its so short that even YOU might key in this marvel!

- It teaches you a new meaning for the word billboard. Not to be confused with bulletin board, it is a term used to describe an electronic message of moving characters. The poem appears in a window in the snow, yes, in the middle of a wilderness miles from the closest glacier. The characters shift to the left in an almost endless stream — that's billboarding.

- The snow fills up the woods in a text animation taking license with * as a snowflake, keeping the snow out of the window until the very last; then the remnants of the poem are buried in light powder.

- It continuously monitors the keyboard location to see if you have dared to press ESCape; if so it thinks you want it to start over (a tricky feature it really doesn't need).

- It works on any Apple II and doesn't need a color monitor, unless you prefer white snow.

- It's easily modified to run on any machine that speaks BASIC, and the hardware and dialectic changes should be a minimum and not too difficult.

The chart shows the data strung-out to help you get this keyed correctly.

The program counts characters (spaces, too) to make processing decisions. If you change the number of spaces, in particular, it can throw the counter off; so key in the data very carefully!

Program Description

This is the mainline of the program:

10 Clears the screen, reads the first data string, (and clears the keyboard strobe).
20 For the first 593 characters do the following:
30 If the working string, A$, is too short, read the next, B$, to lengthen it.
40 Set to print on line 12, print in subroutine, and repeat until 593

```
0   REM  SNOWY EVENING (FROST)
10  HOME : READ A$: POKE 49168,0
20  FOR I = 1 TO 593
30  IF  LEN (A$) < 20 THEN  READ B$:A$ = A$ + B$
40  VTAB 12: GOSUB 110: NEXT I
50 A$ =  MID$ (A$,20,200)
60  FOR I = I TO 613
70  VTAB 13: GOSUB 110: NEXT I
80  FOR J = 1 TO 200: GOSUB 130: NEXT J: RUN
90  :
100  REM  SUBROUTINE
110  HTAB 10: PRINT  LEFT$ (A$,20)
120 A$ =  MID$ (A$,2,200)
130  FOR K = 1 TO 2
140 Y = 24 *  RND (1) + 1:X = 39 *  RND (1) + 1
150  IF X > 9 AND X < 31 AND Y > 11 AND Y < 15 AND I < 613 GOTO 130
160  VTAB Y: HTAB X: PRINT "*";
170  IF  PEEK (49152) = 155 THEN  RUN
180  NEXT K: RETURN
190  :
200  REM  DATA
210  DATA "                              WHOSE WOODS THESE ARE I THINK I KNOW.    HIS
         HOUSE IS IN THE VILLAGE, THOUGH.    "
220  DATA "HE WILL NOT SEE ME STOPPING HERE    TO WATCH HIS WOODS
         FILL UP WITH SNOW.    "
230  DATA "MY LITTLE HORSE MUST THINK IT QUEER,    TO STOP WITHOUT
         A FARMHOUSE NEAR    "
240  DATA "BETWEEN THE WOODS AND FROZEN LAKE    THE DARKEST EVENING OF THE YEAR."
250  DATA "HE GIVES HIS HARNESS BELLS A SHAKE    TO ASK IF THERE IS SOME MISTAKE."
260  DATA "THE ONLY OTHER SOUND'S THE SWEEP    OF EASY WIND AND DOWNYFLAKE.    "
270  DATA "THE WOODS ARE LOVELY, DARK, AND DEEP    BUT I HAVE PROMISES TO KEEP    "
280  DATA "AND MILES TO GO BEFORE I SLEEP,    "
290  DATA "AND MILES TO GO                BEFORE I SLEEP.    *"
```

Listing for *SNOWY EVENING (FROST)*

```
            1         2         3         4         5         6         7         8         9         0         1         2
LEN 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
120                                         WHOSE WOODS THESE ARE I THINK I KNOW.    HIS HOUSE IS IN THE VILLAGE, THOUGH.    <
 77 HE WILL NOT SEE ME STOPPING HERE    TO WATCH HIS WOODS FILL UP WITH SNOW.    <
 76 MY LITTLE HORSE MUST THINK IT QUEER,    TO STOP WITHOUT A FARMHOUSE NEAR    <
 73 BETWEEN THE WOODS AND FROZEN LAKE    THE DARKEST EVENING OF THE YEAR.    <
 74 HE GIVES HIS HARNESS BELLS A SHAKE    TO ASK IF THERE IS SOME MISTAKE.    <
 68 THE ONLY OTHER SOUND'S THE SWEEP    OF EASY WIND AND DOWNYFLAKE.    <
 40 THE WOODS ARE LOVELY, DARK, AND DEEP    <
 31 BUT I HAVE PROMISES TO KEEP    <
 35 AND MILES TO GO BEFORE I SLEEP,    <
```

Use the above chart to help key in SNOWY EVENING.

Quick CONNECT

WHAT'S A FELLOW DO?

Apple's Technological Wizards

Just who are these five Apple employees with the title of Apple Fellow? And what is it that they do? In this month's issue of Quick Connect, we'll meet them and find out.

STEVE WOZNIAK

As most Apple® users are well aware, Steve Wozniak engineered the original Apple computer and co-founded Apple Computer, Inc. in 1977.

Steve is a familiar face to most User Groups. Chances are good that you've heard him speak, or met him face-to-face at one of your meetings. The "Woz" is a regular on the User Group Circuit. He makes about twenty visits a year to User Groups, both those inside the United States and in other countries. A visit to a User Group in Wellington, New Zealand, for example, is on his schedule for the early part of 1987.

"I'm their patron," says Steve, "I go every year."

User Groups, he says, were key to the launch and success of the young Apple. Consequently, they will always be special to him and, in fact, the Woz maintains a high profile in the User Group community.

Steve--who is Apple employee number 1--contributed greatly to the development of Apple's newest computer, the Apple IIGS.

Currently, Steve is busy with his own company, CL9 (known affectionately as "Cloud 9"). The company is soon to release its first product, CORE (Control Of Remote Electronics), which Steve describes as a "smart remote control." Woz is a frequent visitor at Apple headquarters, and continually stays in touch with the various product development teams.

ROD HOLT

Rod is the hardware wizard who served as Apple's chief engineer in the company's early years. He was instrumental in developing the computer power supply and the automatic clamping and ejecting disk drive.

At present, Rod is taking some time away from computers to sail the Pacific Ocean from Australia to Japan. But who knows--there may be a computer stowed away somewhere on board!
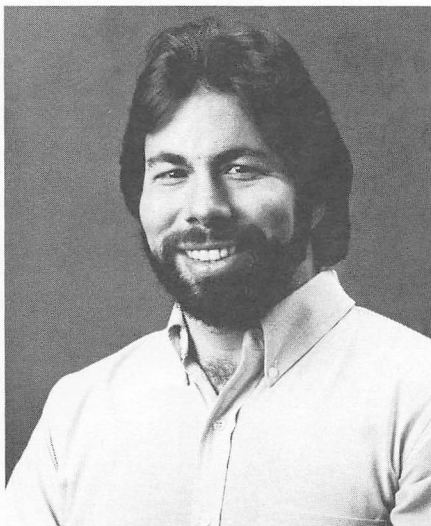
> "User Groups were key to launch and success of a young Apple."

ALAN KAY

Alan is considered one of the personal computer industry's most creative and accomplished scientists. At Apple, Alan's contributions were instrumental to the development of the Macintosh® user interface, with its pull-down menus, windowing and graphics interface.
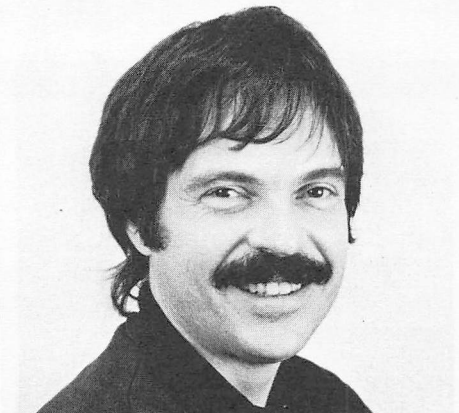
Continuing to focus on interfaces, Alan is currently working in Los Angeles on an Apple-funded project called Vivarium. Vivarium is a unique educational project for children. Working with special Macintosh computers, children create and run simulated environments or ecologies using 3-dimensional color graphics.

AL ALCORN

Al is Apple's most recently appointed Fellow. As some of you "old-timers" may remember, he co-founded Atari back in 1972 with Nolan Bushnell.

According to Al, it was the risk-taking experience he gained at Atari, combined with an enthusiasm and drive that makes products happen, that got him the title of Apple Fellow.

Right now, Al says, he's involved at Apple "in a challenging project that of course I can't tell you about." He did go so far as to say that the project has to do with "increasing the market potential of the Macintosh user interface."

You'll have to use your imagination to fill in the rest.

"I'm probably the only Apple Fellow who actually shows up at the office," he says.

Not surprisingly, Al can't imagine wanting to be anywhere else. Apple, he says, provides him with the most exciting work environment in Silicon Valley.

"Apple encourages risk-taking," he says, "The team I lead has a truly entrepreneurial feeling." Apple is one of the few companies that expects and encourages that attitude, Al says.

Al feels User Groups are instrumental in communicating important issues in a rapidly changing technological environment. User Groups provide a network of support, he says, that is absolutely essential to helping users unleash the full potential of new developments in technology.

"Apple encourages risk-taking."

BILL ATKINSON

Bill, the fifth Apple Fellow, was not available for comment in this month's Quick Connect, but rumor has it he's busy working on an incredible new product that will more than justify his silence. You may know that Macintosh graphics lovers owe heartfelt thanks to Bill. Author of MacPaint, he's the computer graphics wizard who developed the QuickDraw graphics routines for Macintosh. You can be sure Bills next product will be every bit as astounding as the tools he developed for the Macintosh back in 1984.

Apple Kicks Off New Ad Campaign

If you're a Red Sox or a Mets fan, you're sure to have witnessed the debut of Apple's new television ad campaign.

The first of 11 new television ads were aired October 18 during the opening game of the 1986 World Series.

The theme for the new ads, *The Power To Be Your Best*, reflects the "special sense of personal satisfaction that so many Apple users report receiving from using Apple products", says Apple chairman and CEO John Sculley.

The campaign's style, adds Apple advertising manager Bruce Mowery, projects "the warm, human approach to computing that has made Apple successful for ten years."

In fact, the focus of the new ads is so people-oriented that you won't even see a computer in four of the Macintosh ads!

The ads feature the Apple II product line, as well as the Macintosh Plus and LaserWriter® printer. Apple II ads, as you might guess if you haven't seen them yet, focus on the education and home markets, while Macintosh ads target business users.

The ads are the creation of Apple's new ad agency, BBDO.

As users already familiar with Apple products, what is your impression of the new ads? Drop us a line and let us know!
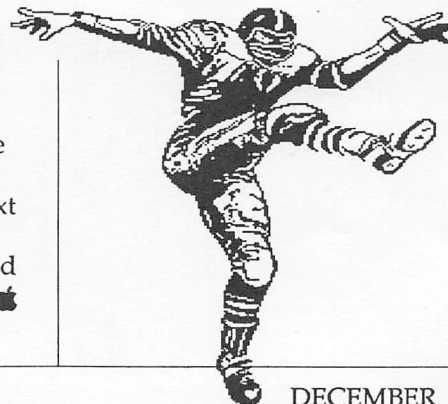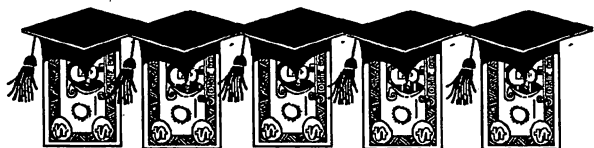
THE BEST IN ACADEMIC COMPUTING

Wheels For The Mind Awards

Apple Computer will honor developers of outstanding educational applications for Apple computers by presenting the *Wheels for the Mind* awards in June of 1987.

The grand prize winner will receive $20,000, with $7,500 going to each of four runners-up.

Students, faculty, and staff from colleges and universities all over the United States are invited to compete for the prizes.

Bud Colligan, Apple's manager of higher education marketing explains, "The *Wheels for the Mind* awards are Apple's way of demonstrating our appreciation for these individuals and their dedication to producing products that make a difference in teaching and learning."

Entries will be accepted in the following categories:

* In Class Instruction: Applications associated with demonstration and presentation materials used by an instructor in a classroom.

* Teaching Tools: Applications associated with a specific course but used outside the classroom (tutorials, programming aids, simulations, lab and data analysis).

* Study/Research Tools: Application tools for helping students, faculty, or staff in areas not associated with a specific course (library searches, word processing, productivity, spreadsheets, desktop communications, desktop publishing, and networking).

* Development Tools: Applications supporting the development of courseware and other educational tools (authoring tools, expert systems, and presentation aids.)

Entries must be received no later than February 28, 1987. Winners will be announced at the Apple University Consortium (AUC) meeting in June 1987.

For official submission form and guidelines, write to:

Wheels for the Mind Awards
Apple Computer, Inc.
20525 Mariani Avenue MS 23-E
Cupertino, CA 95014

VOICE/DATA NETWORKS

Telephone For You, Mac

The Macintosh recently got a boost into the expanding world of voice/data integration.

New products provide high-speed communications and office applications for businesses over ordinary telephone lines between the Macintosh and computers from other manufacturers.

Developed through the joint efforts of Apple Computer Inc. and Northern Telecom Inc.–a leader in telecommunications and integrated information services–new hardware and software products establish telephone links between the Macintosh and computers from IBM, DEC, and Hewlett-Packard.

This joint venture represents Apple's commitment "to integrate Macintosh technology into the business environment," says Charles W. Berger, vice president for business development at Apple. It will provide business users with a range of applications, such as electronic mail and desktop publishing, in the most economical way possible.

Don't hang up, Mac. Business users like what they're hearing!
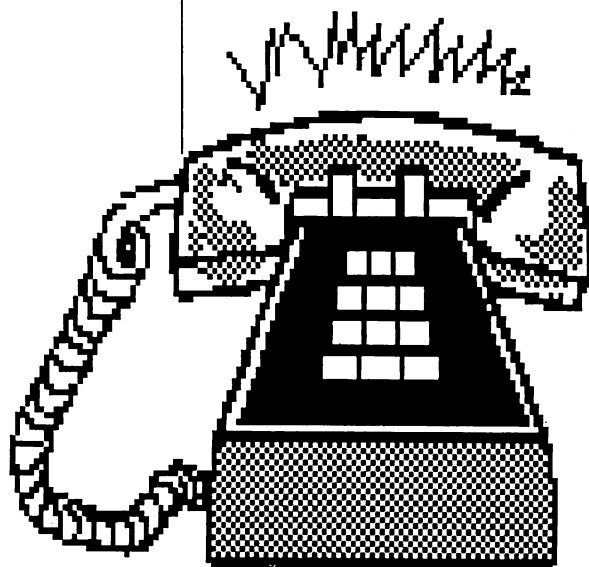
Quick BULLETINS

The Write Stuff: LaserWriter® Tips

Looking to get the highest quality output from your LaserWriter?
Follow these tips:

1. Before inserting cartridge, make sure the toner is evenly distributed by holding cartridge horizontally and slowly rocking it 45 degrees back and forth several times (see manual for drawing).
2. After inserting cartridge, remember to remove sealing tape to release toner.
3. Some cartridges print with a light print density for the first100-200 pages. If print density is too light, adjust print density dial on back of LaserWriter. You may want to run some pages to "break in" the new cartridge.
4. Clean toner cartridge wire as necessary, following instructions in manual.
5. When density begins to decrease towards end of cartridge life rock cartridge as noted above to prolong its usefulness.

Are we getting your news?
If we're not receiving your newsletter--we'd like to!

Please put us on your mailing list:

Apple User Group Support
20525 Mariani Ave, 23-G
Cupertino, CA 95014
Attn: *Quick* CONNECT

Take the New MacDraw™ Guided Tour

A Guided Tour disk for MacDraw is now available for the Macintosh Plus and Macintosh 512K Enhanced. If you want to upgrade from the previous version, send your original MacDraw Guided Tour disk to the Media Exchange Program along with a completed Media Exchange Card. See your local authorized Apple Dealer for more information and Media Exchange Cards.

Staff

Editor: Cathy Hoolihan

Contributors: Ellen Leanse
              Phyllis Farnam

Coordination: Apple Creative Services

Production/Copy: Susan Janus

HAPPY HOLIDAYS
&
BEST WISHES
FOR THE NEW YEAR

From
Apple's User Group Connection Team

characters are processed.

50 The 593rd character is processed, so pickup the last string to display.

60 For characters 594 to 613 do the following:

70 Set to print on line 13, print in subroutine, and repeat until the 613th character is processed.

80 The poem is complete; fill out the snow by making snowflakes in subroutine, and repeat for 200 pairs of snowflakes, and then start the program over.

This is the print and snowflake subroutine:

110 Set to print beginning with the 10th character position of the line, and print the first 20 characters of the working string, A$.

120 Shorten the working string, A$, by deleting the first character.

130 Make two snowflakes (per character of poem).

140 Get a random y-coordinate for VTAB, 1 to 24. Get a random x-coordinate for HTAB, 1 to 39 (to prevent scrolling).

150 If coordinates are inside window, go back to 140, if not at end of poem.

160 Print snowflake.

170 If ESCape pressed, start program over. Omit this and POKE 49168,0 in line 10 if you do not want this feature.

180 If 2 snowflakes are made, then return.

The program runs for just over two minutes with the last half minute spent in the ending snowfill. Enjoy!

—JOHN M. WHITE

From *Appleworks* To BASIC and Back

AppleWorks is justly known for its convenience; the ability to switch between three integrated programs provides ease of use that software for more powerful computers often lacks. Nevertheless, the individual programs are often quite noticeably weaker than stand-alone applications. In particular, the spreadsheet lacks mathematical functions which one finds in better spreadsheets such as *Supercalc 3a*. Applesoft BASIC, on the other hand, has much of the flexibility which is lacking in the *AppleWorks* spreadsheet, but the trouble of setting up files for each use would easily overcome any ease in "massaging" data.

Would it not be wonderful if one could enter the data in a file set up in *AppleWorks*, read it into a basic array to manipulate it, and then return it to a spreadsheet or database in *AppleWorks*? It would be and it is. The programs listed will take a DIF file, read it as a two dimensional array, and then return the data to a DIF file to be read by either a spreadsheet or a database.

The first program, *DATE-READER*, takes the data from a DIF file. I find that a database-produced DIF file works best, as it does not have titles to worry about. The database lacks calculating power, but since the number manipulating will be done in BASIC, that doesn't matter. The program listed assumes the DIF file comes from a database. The DIF file is created when it comes time to print, as printing to a DIF file is one of the options. When the user is asked for the pathway, he lists the disk name followed by a slash and then the name he wants to give the new DIF file.

The program uses a subroutine in line 3600 to manipulate the data. I have included a version in which the subroutine subtracts one date from another and provides the number of years and months between them, and and whether the gain (on a stock) is

long or short term (more or less than six months under the old tax code) as an example. This particular subroutine will accept dates from the data base either in the form "Sep 12 86" or "9/12/86." The program can distinguish between text and numbers. This program happens to turn text into numbers. The subroutine can easily be modified to do something else with the data, of course, but here is an example of an *AppleWorks* database which the subroutine in *DATEREADER* will take

as input and then subtract dates. (See Figure 1)

The program *DIFWRITER.DB* takes the array and rewrites it as a DIF file which can be read by *AppleWorks*. If new columns or fields are required, it is a good idea to create them in advance by leaving blank fields or columns in the original file read by *DIFreader*, as in the example above. When the new DIF file is printed out it will have the calculated values in these fields.

These programs are modifications

STOCK	BOUGHT	SOLD	YEARS	MONTHS	TERM
Comm Psych	Apr 7 86	Jun 24 86			
Halliburton	Sep 9 85	Jun 24 86			
Toys R US	Jun 2 86	Jun 24 86			

Figure 1. A sample portion of the database.

```
10   HOME
20   D$ =   CHR$ (4)
30   PRINT "WORKING WITH A DIF FILE"
50   PRINT : PRINT : PRINT
60   INPUT "NAME OF FILE? ";F$
65   PRINT
67   PRINT D$;"CLOSE"
68   PRINT D$;"PR#3"
70   GOSUB 1000: REM **OPEN FILE
80   GOSUB 300: REM  ** READ HEADER
90   DIM T1$(T,(V + 3)),T1(T,(V + 3))
95   PRINT : PRINT "READING FILE": PRINT
110  GOSUB 500: REM  ** READ NUMBERS
115  GOSUB 1200: REM  ** CLOSE FILE
167  GOSUB 3600
168  GOSUB 3710
169  GOSUB 5001
299  END
300  REM  ** READ HEADER
310  GOSUB 2000
320  IF T$ = "VECTORS" THEN V = V2
330  IF T$ = "TUPLES" THEN T = V2
340  IF T$ <  > "DATA" THEN  GOTO 310
350  RETURN
500  REM  ** READ NUMBERS
510  FOR I = 1 TO T
520  GOSUB 2100: REM  ** BOT - GARBAGE
530  FOR J = 1 TO V
540  GOSUB 2100
550  T1(I,J) = V2:T1$(I,J) = S$
560  NEXT J
570  NEXT I
580  RETURN
1000 REM ** OPEN FILE FOR READING**
1010 D$ =  CHR$ (4): REM  **CONTROL-D
1020  PRINT D$;"OPEN ";F$
1030  PRINT D$;"READ ";F$
```

Listing 1. The *DATEREADER* program.

```
1040    RETURN
1100    REM ** OPEN FILE FOR WRITING**
1110    D$ =  CHR$ (4): REM  ** CONTROL-D
1120    PRINT D$;"OPEN ";F$
1130    PRINT D$;"WRITE ";F$
1140    RETURN
1200    REM ** CLOSE FILE**
1210    PRINT D$;"CLOSE"
1220    RETURN
2000    REM ** READ HEADER ITEM**
2010    INPUT T$
2020    GOSUB 2100
2030    RETURN
2100    REM ** READ DATA ITEM**
2110    INPUT V1,V2
2120    INPUT S$
2130    RETURN
3000    REM  **WRITE HEADER ITEM**
3010    PRINT T$
3020    GOSUB 3200
3030    RETURN
3200    REM ** WRITE DATA ITEM WITH QUOTES**
3210    PRINT V1;",";V2
3220    PRINT  CHR$ (34);S$; CHR$ (34)
3230    RETURN
3600    REM  DATEREADING
3610    DIM D(T),DD(T),DT(T),M(T),MM(T),MT(T),Y(T),YY(T),YT(T)
3620    FOR I = 1 TO T
3630    LET Y(I) =  VAL ( RIGHT$ (T1$(I,3),2))
3640    LET D(I) =  VAL ( MID$ (T1$(I,3),5,2))
3650    LET YY(I) =  VAL ( RIGHT$ (T1$(I,2),2))
3660    LET DD(I) =  VAL ( MID$ (T1$(I,2),5,2))
3670    GOSUB 4000
3699    NEXT I
3700    RETURN
3710    FOR I = 1 TO T
3720    LET YT(I) = Y(I) - YY(I)
3730    LET MT(I) = M(I) - MM(I)
3740    IF MM(I) > M(I) THEN MT(I) = MT(I) + 12
3750    IF MM(I) > M(I) THEN YT(I) = YT(I) - 1
3760    LET DT(I) = D(I) - DD(I)
3770    IF (M(I) = 1) OR (M(I) = 3) OR (M(I) = 5) OR (M(I) = 7) OR (M(I) = 8) OR
        (M(I) = 10) OR (M(I) = 12) THEN  LET X = 31
3780    IF (M(I) = 4) OR (M(I) = 6) OR (M(I) = 9) OR (M(I) = 11) THEN  LET X = 30
3785    IF (M(I) = 2) AND (Y(I) = 84) THEN X = 29
3790    IF DD(I) > D(I) THEN DT(I) = DT(I) + X
3800    IF DD(I) > D(I) THEN MT(I) = MT(I) - 1
3805    LET T1(I,5) = MT(I)
3806    LET T1(I,4) = YT(I)
3807    IF (YT(I) < 1) AND (MT(I) < 6) THEN T1$(I,6) = "  SHORT"
3809    IF (MT(I) = > 6) OR (YT(I) > 0) THEN T1$(I,6) = "  LONG"
3810    NEXT I
3820    RETURN
4000    REM  MONTHS INTO DATES
4010    IF  MID$ (T1$(I,3),1,3) = "Feb" THEN M(I) = 2
4011    IF  MID$ (T1$(I,3),1,3) = "Jan" THEN M(I) = 1
4012    IF  MID$ (T1$(I,3),1,3) = "Mar" THEN M(I) = 3
4013    IF  MID$ (T1$(I,3),1,3) = "Apr" THEN M(I) = 4
4014    IF  MID$ (T1$(I,3),1,3) = "May" THEN M(I) = 5
4015    IF  MID$ (T1$(I,3),1,3) = "Jun" THEN M(I) = 6
4016    IF  MID$ (T1$(I,3),1,3) = "Jul" THEN M(I) = 7
4017    IF  MID$ (T1$(I,3),1,3) = "Aug" THEN M(I) = 8
4019    IF  MID$ (T1$(I,3),1,3) = "Sep" THEN M(I) = 9
4020    IF  MID$ (T1$(I,3),1,3) = "Oct" THEN M(I) = 10
4021    IF  MID$ (T1$(I,3),1,3) = "Nov" THEN M(I) = 11
4022    IF  MID$ (T1$(I,3),1,3) = "Dec" THEN M(I) = 12
4023    REM  ** MM = MONTH BOUGHT**
4024    IF  MID$ (T1$(I,2),1,3) = "Jan" THEN MM(I) = 1
4025    IF  MID$ (T1$(I,2),1,3) = "Feb" THEN MM(I) = 2
4026    IF  MID$ (T1$(I,2),1,3) = "Mar" THEN MM(I) = 3
4027    IF  MID$ (T1$(I,2),1,3) = "Apr" THEN MM(I) = 4
4028    IF  MID$ (T1$(I,2),1,3) = "May" THEN MM(I) = 5
4029    IF  MID$ (T1$(I,2),1,3) = "Jun" THEN MM(I) = 6
4030    IF  MID$ (T1$(I,2),1,3) = "Jul" THEN MM(I) = 7
4031    IF  MID$ (T1$(I,2),1,3) = "Aug" THEN MM(I) = 8
4032    IF  MID$ (T1$(I,2),1,3) = "Sep" THEN MM(I) = 9
4033    IF  MID$ (T1$(I,2),1,3) = "Oct" THEN MM(I) = 10
4034    IF  MID$ (T1$(I,2),1,3) = "Nov" THEN MM(I) = 11
4035    IF  MID$ (T1$(I,2),1,3) = "Dec" THEN MM(I) = 12
4036    IF  ASC ( LEFT$ (T1$(I,3),1)) < 65 THEN M(I) =  VAL ( MID$ (T1$(I,3),1,2))
4037    IF  ASC ( LEFT$ (T1$(I,3),1)) < 65 THEN D(I) =  VAL ( MID$ (T1$(I,3),4,2))
4038    IF  ASC ( LEFT$ (T1$(I,2),1)) < 65 THEN MM(I) =  VAL ( MID$ (T1$(I,2),1,2))
4039    IF  ASC ( LEFT$ (T1$(I,2),1)) < 65 THEN DD(I) =  VAL ( MID$ (T1$(I,2),4,2))
5000    RETURN
5001    REM  *** PRINTOUT
5002    PRINT "STOCK"; TAB( 19)"BUY"; TAB( 12)"SELL"; TAB( 10)"YEARS"; TAB(
        3)"MONTHS"; TAB( 3)"TERM"
5003    PRINT
        "=========================================================="
5004    FOR I = 1 TO T
5005    PRINT T1$(I,1) TAB( 24 -  LEN (T1$(I,1)))T1$(I,2); TAB( 15 -  LEN
        (T1$(I,2)))T1$(I,3); TAB( 15 -  LEN (T1$(I,3)))T1(I,4); TAB( 7)T1(I,5); TAB(
        6)T1$(I,6)
5006    NEXT I
5007    RETURN
```

End of Listing 1.

---

of programs found in the book *Mastering VisiCalc* by Douglas Hergert (Sybex,1983). The programs contained there required data with titles down the left-hand side and sums of the other columns down the right. This program will accept a DIF file of any sort. However, one of the advantages of such a program, as Hergert points out, is that in many spreadsheets, there are a number of intermediate calculations which fill up a spreadsheet, but which are of no value in themselves and which fill up columns and separate the results from the data. If the calculations are done in BASIC, these intermediate calculations need not appear at all.

I am an inexperienced programmer; most of what I know I learned from doing these programs. If anyone has any improvements, I would appreciate being called at 749-4726.

—PETER GINGISS

AppleWorks Articles?

Due to the great popularity of *AppleWorks*, we would like to again ask for more articles, tips, techniques, and new product announcements relating to it. Since most *AppleWorks* users already have an investment in both time and money in using the program, there is most likely a great number of tips that could be shared about getting an even greater return on that investment. Furthermore, new product releases and reviews would be very much welcomed. Since this is one of the best forums for the dissemination of information, we would like to have at least one article monthly devoted to *AppleWorks*. But as Smokey the Bear says, "Only you..."

—CLF

```
600  REM   WRITE NEW FILE
603  REM
604  REM   *FROM ADB ORIGINAL*
605  REM
610  REM   HEADER SECTION
615  INPUT "WHAT IS THE NAME OF THE FILE? ";F$
616  PRINT : PRINT
617  PRINT "WRITING A FILE CALLED ";F$
618  PRINT : PRINT : PRINT
620  GOSUB 4000
625  V1 = 0:S$ = ""
630  T$ = "TABLE":V2 = 1: GOSUB 3000
640  T$ = "VECTORS":V2 = V: GOSUB 3000
650  T$ = "TUPLES":V2 = T: GOSUB 3000
660  T$ = "DATA":V2 = 0: GOSUB 3000
670  REM   **STRING TUPLE**
690  V1 = 1:V2 = 0
700  FOR I = 1 TO T
710  GOSUB 900: REM  ** BOT FLAG
715  V1 = 1:V2 = 0
720  FOR J = 1 TO V
730  S$ = T1$(I,J)
732  IF S$ = "" THEN S$ = "V"
735  IF S$ = "V" THEN  GOSUB 751
736  IF S$ < > "V" THEN V1 = 1:V2 = 0: GOSUB 3200
740  NEXT J
745  NEXT I
748  ONERR  GOTO 800
751  V1 = 0:V2 = T1(I,J): GOSUB 3100
760  RETURN
800  V1 =  - 1:V2 = 0:S$ = "EOD"
810  GOSUB 3100
815  GOSUB 5000
820  END
900  REM   ***WRITE BOT FLAG
910  V1 =  - 1:V2 = 0:S$ = "BOT": GOSUB 3100
920  RETURN
3000  REM  *** WRITE HEADER ITEM
3010  PRINT T$
3020  GOSUB 3200
3030  RETURN
3100  REM  ***WRITE DATA ITEM WITHOUT QUOTES
3110  PRINT V1;",",V2
3120  PRINT S$
3130  RETURN
3200  REM  ** PRINT DATA ITEM WITH QUOTES
3210  PRINT V1;",";V2
3220  PRINT  CHR$ (34);S$; CHR$ (34)
3230  RETURN
4000  REM  **OPEN FILE FOR WRITING
4010  D$ =  CHR$ (4)
4020  PRINT D$;"OPEN ";F$
4030  PRINT D$;"WRITE ";F$
4040  RETURN
5000  REM  **CLOSE FILE
5010  PRINT D$;"CLOSE"
5020  RETURN
```

Listing 2. *DIFWRITER.DB*

1. Locate a gold watch and chain
2. Wave it back and forth in front of your face.
3. Repeat the phrase, "I will write an article for the *Apple Barrel,*" until your eyes glaze over and you have a sudden urge to put your thoughts down on paper.
4. Place the paper in an envelope and mail it to HAAUG.
5. Wait for the mail to arrive with the *Apple Barrel* that you helped put together.

Apple II

Apple Access	Patrick Hennessey	342-0643
AppleWorks	Jack Cowart	467-4215
	John Slack	491-1747
AppleWriter II	Robin Cox	778-1635
AppleWriter III	Shorty Hirsekron	697-9797
BusinessWorks	Ron Burgess	270-8243
CAD-Apple	Dick Fairman	723-9619
CP/M	Kip Reiner	550-0568
dBase II	H. Helm	529-7405
Data Factory	Dick Fairman	723-9619
Dollars and Sense	Leon Cesazini	467-3650
DowJones Market		
Analyzer	Ed Hurley	497-8877
Family Roots	Clifton Taylor	462-1698
Home Accountant	Rick Oshlo	578-8530
IAC Calc	Chris Kravits	320-0381
Letter Perfect	Bill Muhlhausen	668-3963
List Handler	Rick Kravits	320-0381
MCI Mail	Patrick Hennessey	342-0643
Multiplan	Mackie Brown	688-8655
	Jack Bailie	440-5963
	Glen Ray	359-2103
MultiScribe	Henry Bernstein	728-0081
Palantir	Re Kramer	358-6687
Pie Writer	Ray Fielding	358-4217
Quickfile	Jack Cowart	467-4215
ScreenWriter II	Dick Fairman	723-9619
Space Eggs	T. Sloan Engle	774-9055
Time Is Money	Dick Fairman	723-9619
VersaForm III	Shorty Hirsekron	697-9797
VisiCalc(Advanced)	Gerald L. Penn	498-0079
Wordstar	Ed Haymes	522-1863
Word Handler	Rick Kravits	320-0381

Hardware

Abaton Optical		
Character Reader	H. Helm	529-7405
Applied Engineering		
Upgrade Cards	Bruce Sprague	360-1000
DataFrame Hard Disks	Mike Kramer	358-6687
Mac XL	Mike Conway	495-2292
Sider Hard Disk Drives	Robin Cox	778-1635
Mac XL	Mike Conway	495-2292

Would you like to be a specialist? Leave your name and phone number at the Answer Desk during the meeting or mail it to the *Apple Barrel*.

Macintosh

Aztec C	Mike Martin	486-9169
BeamMac	Ronnie Haws	444-6720
Cricket Graph	Rudge Allen	622-3979
	David Ullman-Dougherty	995-5795
DB Master	Mike Conway	495-2292
Dollars & Sense	Gerald L. Penn	498-0079
Filevision	Mike Martin	486-9169
FrameMac	Ronnie Haws	444-6720
FullPaint	David Ullman-Dougherty	995-5795
Helix	Steve Bass	847-4407
	H. Helm	529-7405
Mac-SIG	Steve Bass	847-4407
MacDraw	Billy Jacobs	480-4410
	Mike Kramer	358-6687
MacDraft	Jim Huck	496-9508
	Billy Jacobs	480-4410
MacLink	Mike Kramer	358-6687
MacPaint	Bill Hailey	682-5147
Mac Transfer	Mike Kramer	358-6687
MacWrite	Bill Hailey	682-5147
Mac 3-D	Larry Stage	358-1105
Megamax C	Chris Flick	363-3153
MGMS CAD	Steve Bass	847-4407
Microsoft BASIC	Bill Hailey	682-5147
Microsoft Chart	Mike Martin	486-9169
Microsoft Excel	Les Brown	795-2741
Microsoft File	Rudge Allen	622-3979
	Mike Martin	486-9169
	Gerald L. Penn	498-0079
Microsoft Multiplan	Mike Conway	495-2292
	Peter Lemettais	523-1390
Microsoft Word	Virginia Conway	495-2292
OverVue 2.0	H. Helm	529-7405
	Neal Scott	890-0532
PageMaker	Re Kramer	358-6687
Ready, Set, Go!	Cleland Early	941-7247
Reflex	Mike Kramer	358-6687
Red Ryder	David Ullman-Dougherty	995-5795
ResEdit	David Ullman-Dougherty	995-5795
Silicon Press	Re Kramer	358-6687
SuperPaint	Mike Kramer	358-6687
TML Pascal	Grady Beaird	529-8420
Tekalike	Mike Martin	486-9169
Thunderscan	David Ullman-Dougherty	995-5795
Turbo Maccountant	Jack Cowart	467-4215
XLisp	Mike Martin	486-9169
ZBasic	Tom Dillon	376-6502

Please remember that the specialists are volunteers and be considerate when you call them. Don't call late at night unless you clear it with them beforehand
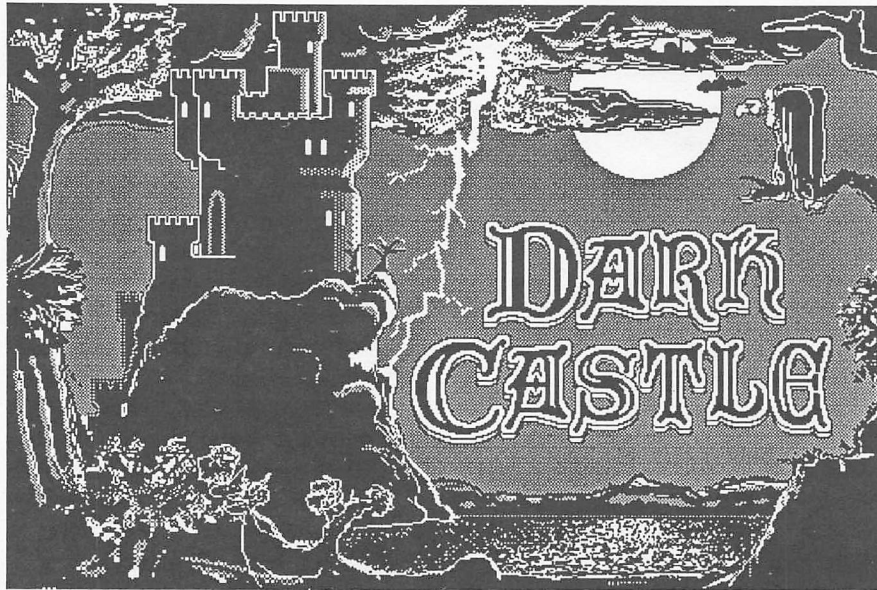
Imagine yourself magically transported to an eerie, forbidding castle. The drawbridge drops with a creak and a thud. You enter into a huge hall lit by flickering torches with four doors before you. Behind each door lies a series of treacherous obstacles and dungeons which you must conquer to fulfill your quest — to find and defeat the evil Black Knight! As you move from chamber to chamber you hurle rocks at poisonous rats and bats, zombielike guards who shoot deadly arrows, the Torturer snapping his whip with a crack, strange little mutants, swooping birds, the spell casting wizard and the firebreathing dragon. At the same time you have to avoid falling through trap doors, drowning or being smashed by rolling, bouncing boulders. This is the concept of a new game for the Macintosh called *Dark Castle*.

Does all this sound familiar? There have been a number of similar themes on other computers (and even on the Mac) but none of them approach *Dark Castle* in its look and feel. *Dark Castle* is a product of Silicon Beach Software. Their demo of this program at the Dallas Mac Expo immediately caught and kept my attention with its amazing graphics and state of the art animation. It has incorporated the use of *RealSound* (the same digitized technique that made *AirBorne* famous, but

even better!) to create the realistic and often amusing sound effects. They hired a graphics artist to design the first-class
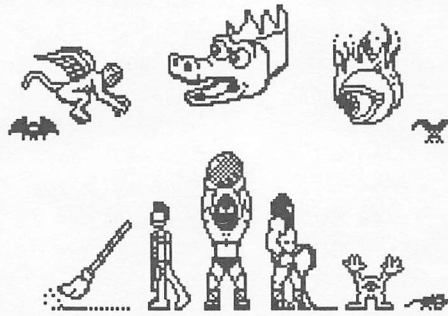
The Bad Guys

screens.

Both the mouse and the keyboard are used simultaneously to play which I

considered the first challenge of the game. The second challenge was the problem of having all the instructions on the program disk (ie. no hard copy) which, when playing the game the first few times, can be a real pain if you don't have a photographic memory. Once conquering those obstacles I found *Dark Castle* to be fast-paced, fun, and challenging (I should say, it is still fast-paced, fun, and challenging after a month of play)!

The game was officially released in October and is available at software stores and mail-order houses. *Dark Castle* requires at least 512K to run and it will also work with the newer Mac Plus. It comes on two diskettes but it claims it can run off of just one 400K drive (with some disk swapping I suppose). It is NOT copy protected (hurray) so you can put it all onto one 800k disk or even onto your hard disk (it works just fine on our Apple HD-20 SC). It retails for $50.00 and is well worth the price! The arcade action appeals to children of all ages and even the most stodgiest of adults will enjoy the graphics and sounds. Whenever I want to impress someone with what my Mac can do I show them *Dark Castle*.
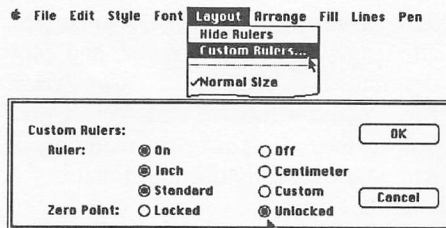
— MAUREEN
LIGHT
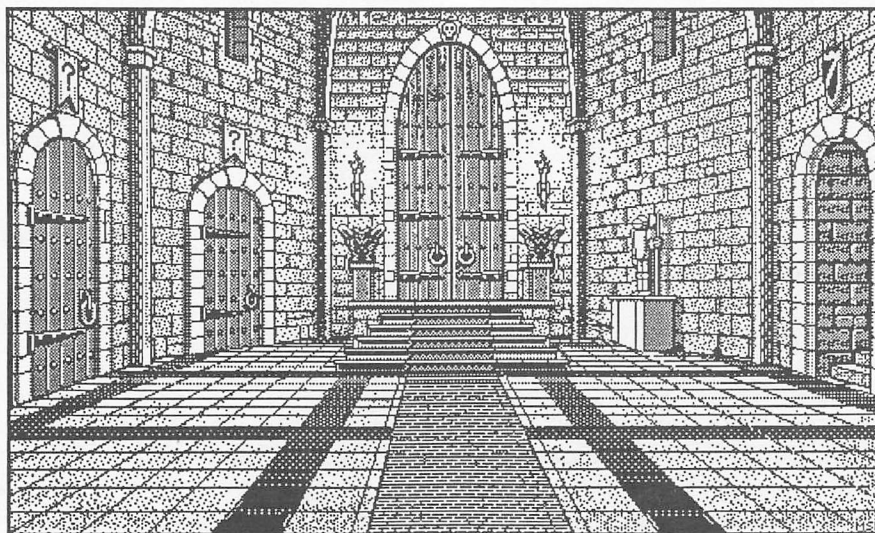
The Great Hall awaits...

Forms Design with *MacDraw*

MacDraw is an excellent package, but one of the areas where it can be most useful has been overlooked, and that is in the area of forms design when teamed up with a LaserWriter. There are a couple of tricks you need to know though, to really make *MacDraw* earn its keep in the design room.

Selecting and using **Custom Rulers**

Do you like the movable zero point in *PageMaker*? Comes in handy doesn't it? You also have one in *MacDraw*, and it's just as handy. Just pull down the **Layout** menu and let off the button when you're on **Custom Rulers**. This will give you a dialog box as shown, and what you want to do is click on **Unlocked** next to **Zero Point**. Then click on the **OK** button.

Now repositioning the Zero Point is pretty easy. Make sure you have the
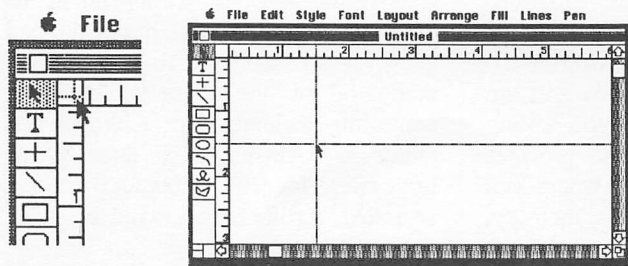
Changing the Zero Point

arrow cursor selected, and then just point to the box in the upper left hand corner of the draw window. Click the mouse button and while holding it down, drag your cursor out into the window. You should see two gray lines intersecting at the tip of your cursor as you drag it around the screen, the

intersection of those lines is the new zero point. When you let go of the mouse button, the lines disappear and the rulers renumber themselves to show the new zero point. To move the zero point somewhere else, just repeat the process as many times as needed.

Another real handy feature in *MacDraw* is a way to get type down to 4.5 points, which is necessary since most lettering on forms is between 6 and 8 points. By the way, this trick only works with either a LaserWriter or some other PostScript compatible device.

Pull down the **File** menu and go down to *Page Setup...* and let off. When the dialog box pops up, tell it that you want a 50% reduction, and then click in the **OK** button.

This will reduce anything you place on the page 50%, and *MacDraw* compensates for this by now telling you that your paper size is 15"x 20", since it will reduce to 7.5"x10". It also means that PostScript will do picture perfect reductions of all of your Laser Fonts, 12 pt. will become 6 pt. and so on. The only catch is that all of your measurements have to be doubled in you head, because if you want a three inch line on paper, it needs to be 6 inches on screen. This can get very confusing, but it doesn't have to be, because *MacDraw* also has a way around this problem, too.

Go back to the **Layout** menu and pull down to *Custom Rulers...*, and let off the mouse button. When you get the **Custom Ruler** dialog box, click on the **Custom** button, and the dialog box will expand to the one shown below.

You want to set you **Major Division Spacing** at 2, which tells the ruler to treat 2 inches of space as 1.

Number of Minor Divisions can be set as desired, usually 16 is good since most forms work doesn't require more than a 1/16" degree of accuracy.
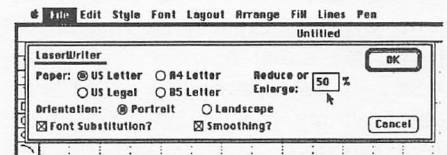
50% reduction on the LaserWriter

Numbering Increments should be set to 1. With your Custom Rulers set up this way and a 50% Page Reduction in effect, one measured inch on screen (either by ruler or by choosing *Show Size*) will be one measured inch on the printed page, the only thing you will have to remember is to cut your typestyle size in half, as that still doesn't change, 12 pt. shows as 12 pt. even though it will print out as 6 pt. Of course after you have all of these changes made, you'll want to save this blank page under some sort of name ("50% Blank" perhaps) so you don't have to do all of this setup again.
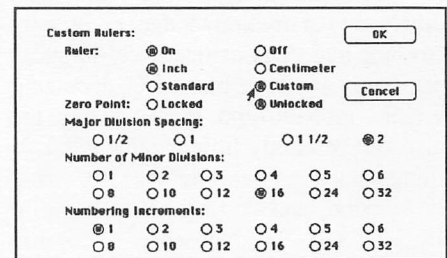
Custom Ruler dialog box

With these two tips in mind, *MacDraw* becomes a very powerful forms design tool, since its built in alignment features and grouping capabilities are wished for tools for most designers that have been doing it by hand. Try it out and see if you can tell the difference.

— *DARYL TSCHOEPE*

Turbo Pascal

UnitMover   Turbo

by Borland International

Reviewed by Dan Podwall

Borland International recently released their *Turbo Pascal* compiler for the Macintosh. While I haven't yet had time to put the system through all its paces, my initial reaction is that this system is *hot*. Previously, I had been doing all my Mac programming with *TML Pascal 2.0*. While *TML* still has some advantages, *Turbo* seems to be the clear-cut winner. The system I use is a Mac Plus with only one 800K floppy (I'm still saving for a hard disk!).

What you get for $99 (or less)

Turbo Pascal comes on two 400K floppies and a 462 page manual. Disk 1 contains the *Turbo Pascal* program and the System Folder. Disk 2 contains *Font/DA Mover 3.2*, *RMaker 2.0*, the *Unit Mover* utility (more about this later), *MacsBug 5.2*, *MacinTalk*, *AppleTalk*, and many sample programs. The sample programs are better than average. While some are just repeats of the Software Supplement examples, others are (to my knowledge) created by Borland and demonstrate such things as printing, parameter block I/O, accessing system globals, and more. They are also full of handy little routines that do things such as real number to string conversion, and sorting. According to Borland's "No-Nonsense License Statement," there are no licensing fees or restrictions whatsoever, and you may freely use the code from the example programs, as long as you do not substantially duplicate all or part of their sample program in your program. The disks are not copy-protected.

The manual has a detailed table of contents, index, and appendices. The manual covers topics such as getting started, writing "standard" Pascal programs, an introduction to Mac programming, writing desk accessories, "Inside Turbo Pascal", and practically anything else you might need to know. They document all the parameter-passing to assembly language routines, internal representation of data types, etc. There are also numerous code fragment examples, in addition to the commented examples on disk. The manual does not however, document the 500+ Macintosh ROM routines, so Inside Macintosh and any other relevant books are still required.

The Development System

Disk 1 contains a System Folder and the *Turbo* application (155K). That's all you need to write Mac programs. There are no additional include, object, or other files needed. My single 800K floppy has *Turbo Pascal*, a 220K System Folder (including *MacsBug*), *ResEdit 1.0*, 75K or so of source code, and still has over 150K free.

Turbo Pascal—as do *MPW*, *TML*, *Lisa*, and *LightSpeed* Pascals—uses units. Units are precompiled subroutines that have public sections, called the interface, and private sections, called the *implementation*. Thus if you declare that your program "uses" a precompiled unit, your program may access any constants, types, variables, or subroutines declared in that unit's interface. Any declarations made in the implementation portion of the unit are private and inaccessible to the calling program. *Turbo Pascal* also provides for an initialization section, where you may open files, allocate memory, initialize variables, etc. Units may be compiled into separate files, or automatically merged into a library file.

Unit Mover

The Macintosh interface is implemented as a series of units, in the same fashion as in *MPW Pascal*. These units, as well as those for "plain-vanilla" Pascal are stored as UNIT resources in the *Turbo Pascal* application, which explains the lack of all the auxiliary files. These units are preloaded and accessed with the "uses" clause. External units (library files) are accessed with a compiler directive and a "uses" clause or with the *Unit Mover*. To allow for future changes, there is the *Unit Mover* utility. *Unit Mover* works with a *Font/DA Mover*-style interface, allowing you to add, remove, and copy the units contained in the *Turbo Pascal* file. You can remove units you seldom use so they don't take up memory, or add your own fully-debugged units.

The Editor

Turbo Pascal is an integrated editor, compiler, and linker. While you can, of course, read in any text file, you can't replace the editor with your particular favorite, such as *QUED*. Luckily, the editor is pretty good. It is a RAM-based, multi-window, ASCII-only editor. You can choose any font and font sizes from 9 to 24 point are available. Window stacking, tiling, and zooming are implemented. Search and Replace functions are flexible and fast. Dialog cut and paste are not supported, but if you select text and choose Find or Change the selected text appears as the default. The editor also has undo, and auto-indent. You can alter and save many of of the defaults, both for compiling and editing. Although the editor is RAM-based, it does handle large files. It easily and quickly loaded, searched, scrolled, and saved a 90K+ file.

So what could be wrong? Really not much, but I do have a few complaints/suggestions. The main problem is with tabs. Although you can adjust tab width, they seem to be treated as spaces. For instance, if you have the tab width set to four spaces and are indenting two tabs in, then to remove two tabs you end up backspacing out eight spaces. Even

MockWrite is slicker than this. The shift left and shift right commands, which change the indentation of blocks of text, work the same way, adding and removing spaces instead of tabs. Also, the Mac Plus keyboard cursor keys are ignored. I don't use them often, but at times they do speed things up. Window-splitting is not supported, nor can you open two windows onto the same file. While overall the editor is good, adding these features would improve an already great product.

The Compiler

The compiler exists as part of the editor. With your source window active, you simply choose Run from the Compile menu, or just type Command-R. Your program is compiled in RAM and instantly launched. If there is a syntax error, an alert describes the problem and the error is highlighted in your source. There are actually two modes of compilation. The compile to RAM mode is FAST. Borland claims over 12,000 lines per minute and I wouldn't argue. The first time you compile anything after launching *Turbo Pascal* there is some disk activity while some resources are loaded. Subsequent compilations produce no disk activity. If you've been through the edit-compile-link-run cycle with traditional systems you'll feel like you got a brand-new computer. According to my informal timing, *Turbo Pascal* compiled and linked a 92K, 2200 line program in about 6 seconds. That's better than 20,000 lines per minute! Of course, a program that size would usually be divided into units, which would make the compile time even shorter. The Run command compiles and links your program, opens your resource file if necessary, and launches your program. The compile to disk option compiles and links your program with any necessary units and resources, and creates a standard double-clickable Mac program.

While the development environment provided by *Turbo Pascal* is not as complete as *LightSpeed Pascal's*, it is better than *TML*. Aside from the near instant compile-link-run cycle, quitting your program returns you immediately back to *Turbo Pascal* with all your files open and the windows just as you left

them. Also, the environment helps you find errors. You can, through compiler directives, turn on and off features such as range checking, debugger symbols (subroutine names included in code) and I/O checking. If you don't have a debugger loaded and a run-time error occurs, the standard System Error Dialog comes up with the Restart and Resume buttons enabled. Clicking Resume instantly returns you to *Turbo Pascal* and the compiler attempts to locate, in your source code, where the error occurs. If the error was in an include file, the file is opened and the error highlighted. If the error was within a unit, you may open the source file and choose "Find Error". The compiler does a partial recompilation and then, if possible, highlights the error.

According to my informal timing, Turbo Pascal compiled and linked a 92K, 2200 line program in about 6 seconds. That's better than 20,000 lines per minute!

If you have *MacsBug* (or *TMON*, I imagine), loaded, then error finding is not possible. Instead, the debugger takes control. One problem that I've discovered is that the debugger seems to get confused about debugger symbols when you are running your program from inside *Turbo Pascal*. Normally, debugger symbols are used to indicate the name of the subroutine currently being executed as you debug your code. Instead, the debugger insists that your code is a ROM patch. This problem only occurs from inside *Turbo Pascal*. When you compile to disk and transfer to your application the debugger works fine. The compiler produces reasonably fast code. It uses the moveq, bra.s, etc., 68000 instructions when possible. Using identical source code, *TML Pascal* executes the Sieve of

Erastothenes in 5.73 seconds while *Turbo Pascal* takes 5.93. However, *TML Pascal* produced a 2.5K application while *Turbo* produced a 6K file. On one of my own programs *Turbo Pascal* produced a 25K file where *TML* produced a 17K file. Apparently, *TML's* linker, while much slower, is also much smarter.

When porting code between Pascal compilers, be careful. They all seem to follow the same conventions for interfacing to the Macintosh. However, *Turbo Pascal* differs from *TML*, *MPW*, and *Lisa Pascal* in some of their intrinsic functions. If you rely heavily on things like Cycle, Leave, and short-circuited boolean expression evaluation then you'll have some rewriting to do. Also, *Turbo Pascal's* Exit routine does not take any parameters, so that needs to be changed. On the other hand, *Turbo Pascal* has many useful functions that, if you use them, will not port to other systems. Mostly these are low-level operations for things like bitwise logic, bit-shifting, and memory shifting and manipulation. There is also a "+" operator for fast string concatenation. Many people are familiar with the *MiniEdit* application from Stephen Chernicoff's *Macintosh Revealed* series. In order to port the *TML* version of *MiniEdit* to *Turbo Pascal*, I had to make four changes to the program. *TML* and others allow function result to be treated as variables, as in:

```
Rgn_to_Update := WindowPeek(Front-
    Window^.updateRgn);
```

In *Turbo Pascal*, you need to use a temporary variable, as in:

```
Window_to_Update := FrontWindow;
Rgn_to_Update := WindowPeek(Front-
    Window^.updateRgn);
```

Turbo Pascal also does not allow functions to return records, arrays, etc. You must change these to procedures. They have changed functions such as LLastClick, which usually returns a Cell (record consisting of two integers), to return a longint. Thus I had to change:

```
Cell_last_clicked :=
    LLastClick(myList);
```

to:

```
longint_that_wants_to_be_a_Cell :=
    LLastClick(myList);
Cell_last_clicked :=
    Cell(longint_that_wants_to_be_a_
    Cell);
```

It's Also for Beginners

Turbo Pascal also supplies units that make implementing textbook Pascal programs very easy. It puts up a window which emulates an 80 x 25 character display to be used for readln and writeln calls. The implementation is much better than *TML Pascal's* version, which does not allow mistakes in input to be corrected using the backspace key. There is also support for the standard Pascal file routines, simplified printing, and turtle graphics. All these make *Turbo Pascal* a very nice beginners tool as well.

Turbo Pascal for the Macintosh, while not perfect, is a welcome addition to the available Macintosh development tools. It has certainly removed a lot of the drudgery and added lots more fun to my Mac programming. Given the history of Borland's *Turbo Pascal* for MS-DOS, which is now in its 3.0 version and has dozens of add-ons and third-party products, this is a product to watch.
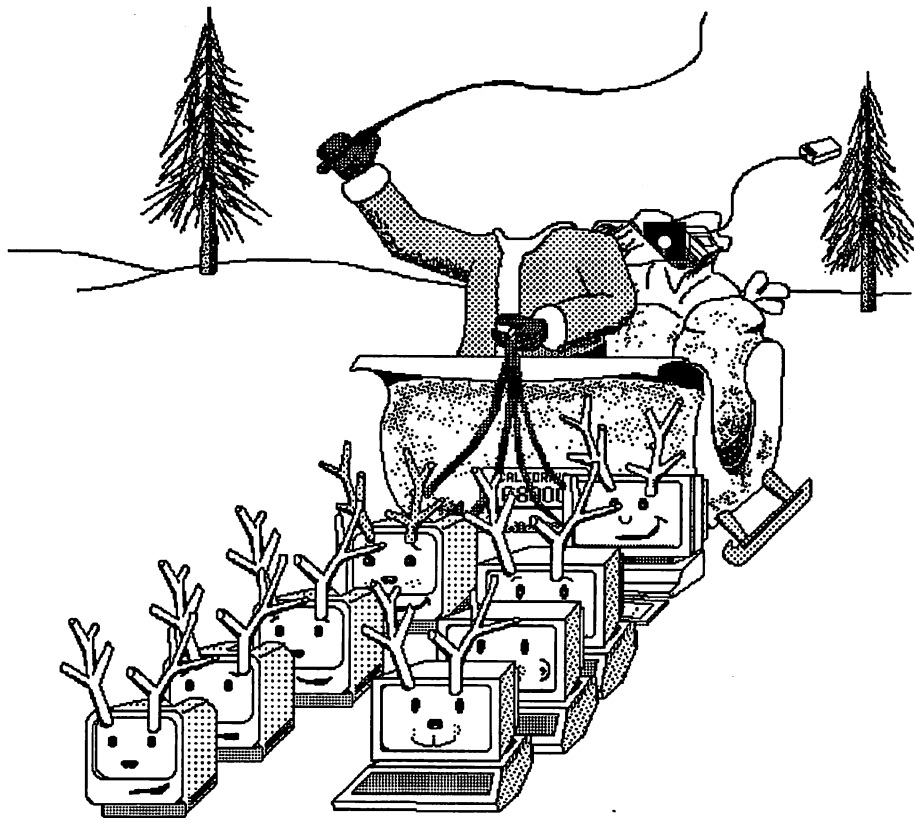
The Cover that was to Have Been

Christmas has come and gone and the above cover still exists. Unfortunately, schedules, equipment failures, and other catastrophes forced our November/December issue to remain on 3 1/2 diskettes. Although not topical, we decided to publish the cover anyway. Tom and I both felt that the idea and execution took enough time to at least warrant its presentation. For those who have already started their Christmas shopping, we hope this will get you back in the spirit.

Life Among the Mazes and Mages:
Macintosh
Wizardry

Computer crack. Microprocessor methadone. High-level-language heroin. Macintosh marijuana. *Wizardry*, the classic fantasy adventure game, is now available in a Macintosh version; like a drug, the game is seductive and addictive. *Wizardry: Proving Grounds of the Mad Overlord* is a crackerjack game.

Most Apple II owners are familiar with *Wizardry*, which is one of the best-selling Apple II games of all time. The Macintosh version is more than just a port; the creators, Andrew Greenberg and Robert Woodhead, claim to have spent more than a year-and-a-half completely redesigning the *Wizardry* user-interface for the Mac. The effort shows. Though Apple wants people to think of the Macintosh as a business computer, *Wizardry* helps prove that the Mac is also a superb game machine.
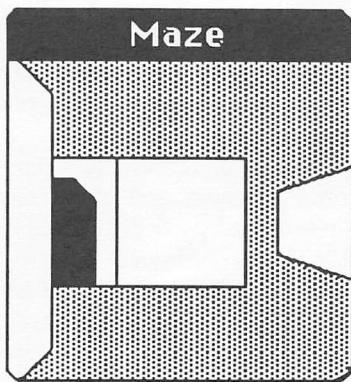
The Macintosh *Wizardry* maze. You can maneuver through it using the mouse or the keyboard.

Getting Strong

Wizardry is actually a series of three games, called scenarios. *Proving Grounds of the Mad Overlord* is the first in the series; presumably Macintosh versions of the other senarios, *Knight of Diamonds* and *Legacy of Llylgamyn*,

are in the works. In *Proving Grounds'* world of magic and mayhem, you create characters and then send them in search of adventure in the maze underneath the castle. All adventuring parties have the ultimate goal of confronting the Evil Wizard Werdna in his lair, defeating him in combat, and retrieving the magic amulet which Wernda has stolen from Trebor, the Mad Overlord. Should a group of characters succeed in their quest and survive the return trip to the castle, those characters will be inducted into Trebor's Elite Guard and be ready for the next scenario.

Trebor uses the Maze underneath his castle as a proving ground for potential Guardsmen. Newly created characters are no match for Werdna, so it's necessary for them to prove themselves in combat against the monsters who live in the maze.

As you might expect, the monsters grow more and more fiendish as the

adventurers venture deeper underground. Should a party encounter a group of monsters and defeat them, the party stands to collect loot—gold and other useful items. In addition, characters who survive these skirmishes gain experience points.

As characters grow more experienced, they pass certain predefined milestones and gain hit points and abilties. The more hit points a character has, the less vulnerable he is. As he gains abilities, a character becomes more potent in combat. Gaining enough experience to successfully confront Werdna is a long and involved, sometimes even tedious process.

Creating Characters

To create a character, you double-click on the training grounds icon. A dialog box appears, offering you two choices. You may retire, purge, an
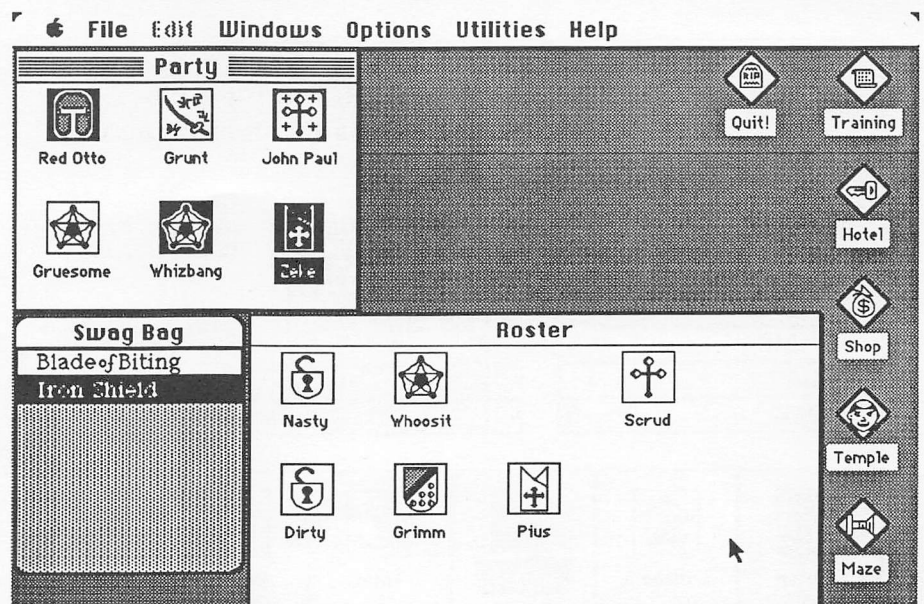
The *Wizardry* desktop. Icons represent characters or areas of Trebor's dominion. The roster window displays all available characters. The party window shows the characters ready to depart on an expedition. Clicking on one of the five application icons to the right will send a highlighted character or the whole party into the shop, temple, inn, maze, or training grounds.

existing character, or you may create an entirely new character. If you choose to invent a new character, a second dialog box appears, prompting you to choose a race and alignment for your new character. You may choose from five races: human, elf, dwarf, gnome, and hobbit. A character may have one of three alignments—good, neutral, or evil.

The next step in the character-creation process is choosing a class for your character. Initially, your character may be one of six professions, *classes*—fighter, mage (magic-user), priest, thief, bishop, samurai. Two other classes, ninja and lord, are available later in the game after the character has gained some experience. The exact class choices available to you depend upon the character's race and alignment. Certain classes must be of specified alignments. For example, ninjas must be evil.

Each type of race has a different set of initial characteristics, what Woodhead and Greenberg call *statistics:* strength, intelligence, piety, vitality, agility, and luck. The program randomly grants you bonus points which you can assign to any of the catagories. When the point total in a key catagory exceeds a threshold, a profession will be available

for you to choose. For example, a priest must have a minimum piety score of eleven; a fighter must have at least eleven strength points.

The mechanics of the character-creating process sound a lot more complicated than they really are. On the other hand, character-creating strategy is more complicated than you might expect. Fortunately, the game manual gives some hints about what races to assign to what job and what party make-up is most effective.

Off We Go...

Once you've created a few characters, you're almost ready to go adventuring. The first step is to organize a party. You are allowed up to six characters in a party; however, it generally doesn't make much sense to have fewer than six in one unless you are organizing a party to find and rescue charcters who are lost in the maze. Then you must leave room in the rescue party for the lost characters. To organize a party, you move characters from the roster window to the party window.

The order in which you place the characters in the party window is important. Only the characters in the top rank can conduct hand-to-hand

combat. Characters in the bottom or rear rank may cast spells against monsters, but they cannot actually hit a monster with a sword. Generally, you place the most vulnerable characters, like mages and thieves, in the back and more robust characters like fighters, bishops, and samurai in the front row.

After you've created a party, you need to equip them with armor and weapons. To do this, you double-click on the shop icon and enter Boltac's Trading Post. Once there, your characters can purchase the items they'll need. Each class of characters has limitations on the kind of armor and weapons it may use. Of course, economics is a consideration. Newly created characters will have little money, so they may have to settle for bargain-basement equipment. Boltac also buys unneeded equipment characters may pick up in the maze. Such items will initially be unidentified, and Boltac will also identify them for a price.

Having organized and equipped a party, it's time to go on an adventure. You double-click on the maze icon and find your party on the first level of the maze. As in most adventure games, the game manual gives precious little information about the maze itself, nor do the instructions provide many hints about how to proceed once you begin exploring. Also like most adventure games, mapping is an inordinately large part of successfully completing the game.

Though there are a few puzzles in *Proving Grounds*, they are relatively simple (in comparsion to those in, say, *ZORK*) and once you figure out one puzzle, the solution to the rest becomes pretty clear. For the most part, *Proving Grounds* is a game of exploring, mapping, and combat. Of these three, combat is by far the most important.

Combat

In *Proving Grounds*, combat occurs when your party of adventurers moves into an area occupied by a group or groups of monsters. Within the larger conflict, combat occurs in *rounds*. In a round, each character or monster has one opportunity to attack the opponents, using hand-to-hand combat (if in the first row), prayer (if a priest, bishop or lord), or magic (if able to use magic

Use the Bonus points to qualify for a class. When all the points are used, select a class and click Create.

Strength	11
I.Q.	11
Piety	11
Vitality	9
Agility	11
Luck	9

Bonus   1

Create

Cancel

Fighter   Mage   Priest   Thief   Bishop   Samurai

The character-creation window. You allocate bonus points by moving the scroll bars. To create a character, click on one of the active class icons at the bottom of the window and then click the "create" button.

powers). The order of the individual attacks is determined randomly, so the results of combat are frequently left to chance. When powerful creatures confront each other, the combatant who gets in the first blow is more likely to prevail. In addition, sword hits and magic spells are also subject to some randomness; we only know that a given spell or a sword hit by a particular character will cause damage in a predetermined range.

Occasionally, the program will decide that the adventurers have surprised the monsters; other times, the monsters will ambush the adventurers. In either case, one side will get a free shot at the other (hand-to-hand combat only). After this optional half-round, combat really begins. As the unseen general of the adventurers, you first must examine the relative strengths of the adventurers and the monsters and decide whether to attempt to run away or to attack. If you choose to run, you risk giving the monsters a free shot, for periodically they will cut off your party's escape route.

Most of the time, then, the wisest plan is to attack. You decide what action each of your characters will take that round: strike with a weapon, cast a spell, attempt to dispell a monster, or avoid combat. Monsters tend to congregate in parties, and within parties, in groups of like monsters. If a front-row character fights hand-to-hand, you must designate a group to attack; however, the adventurer can attack only one monster at a time. Most combat magic spells affect an entire monster group, but within each party and even within each group, monsters have differing abilities to resist magic. The other two adventurer choices, dispelling or avoiding, are usually less desirable. Dispelling—sending undead creatures back to the netherworlds—rarely works, and avoiding combat simply subjects the character to an attack without a retaliation.

At the end of the round, the combat process repeats, unless one side has exterminated the other. If the adventurers have won, the survivors collect experience points and, if the monsters aren't carrying a treasure chest, gold. If the monsters are guarding a chest, it may be booby-trapped. Chests

The character information window. Double-click on a character icon to see this display. This window shows a character's current status, allows you to manipulate items like weapons and armor, and lets you cast peacetime magic spells.

contain gold and may also hold other potentially valuable items like swords, scrolls, rings, and potions. With or without a chest, the spoils of victory tend to vary directly with the risks. A difficult fight against a tough group of monsters will carry more experience points than a battle against an overmatched foe. A booby-trapped chest will contain more loot than monsters' pockets.

Back Home

At some point in its trip into the maze, a party's hit points and magic spells will be nearly exhausted and perhaps one or two party members will be dead. It's time to head for the house. Back at the castle, wounded adventurers will need to check into the Adventurer's Inn for some R&R and some healing. At a cost, your characters may remain until they have regain all their lost hit points. After a stay in the Inn, you'll be told whether each character accumulated enough experience points to make the next experience level. If the character hasn't made a level, you'll be given a progress report.

Dead adventurers will need to visit the Temple of Cant. Two magic spells can be used to raise the dead. The first, Di, restores life, but the character will only keep one hit-point. The second spell, Kadorto, is more potent; it restores life and all hit points. However, the priests at the Temple only
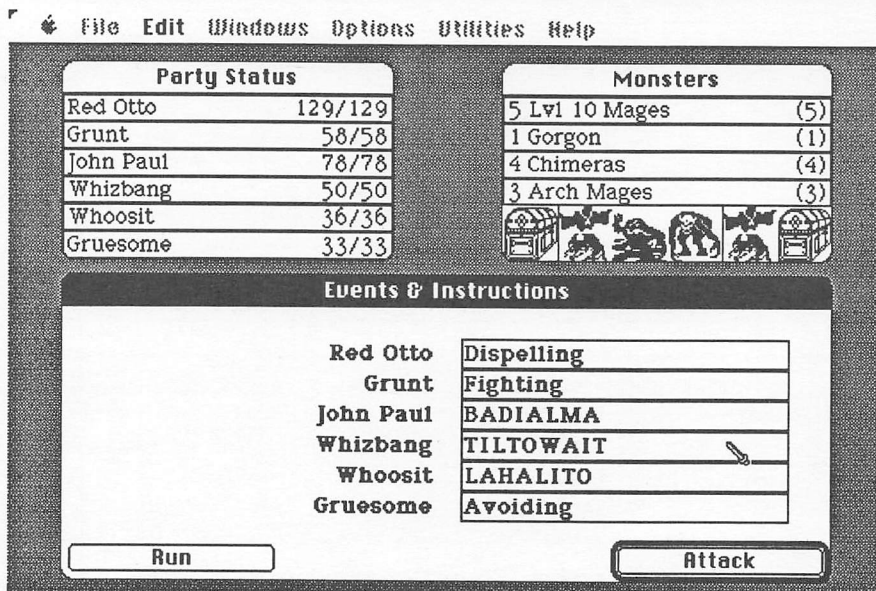
apply Kadorto when an adventurer has been reduced to Ashes by a failed attempt at a Di. Eventually, priests, lords, bishops gain the ability to both heal and resurrect characters, but the Temple remains less risky and more effective. Unfortunately, like everything else in the castle, a visit to the Temple costs gold.

Likes and Dislikes

As you can tell from even this brief description, *Wizardry* is a pretty complex game; yet, the complexity is not really onerous. The world of *Wizardry* is internally consistent and the game structure is logical. Because the characters gain skills gradually, you learn the complexity gradually. The game package includes a 68-page instruction book, which a new player should read before attempting to play. Even experienced players will need to keep the manual close for reference. Fortunately, the *Player's Guide* is a delight, nicely written and illustrated by charming line drawings.

The game's graphics are generally excellent and a big imporvement over the Apple II version. The Mac's icon-oriented interface seems particularly well-suited to a game like *Wizardry*. There is, however, at least one problem with the game's graphics design—the way the program redraws the desktop when a window is moved or closed is really clunky. During seemingly

File Edit Windows Options Utilities Help

Party Status	
Red Otto	129/129
Grunt	58/58
John Paul	78/78
Whizbang	50/50
Whoosit	36/36
Gruesome	33/33

Monsters	
5 Lvl 10 Mages	(5)
1 Gorgon	(1)
4 Chimeras	(4)
3 Arch Mages	(3)

Events & Instructions

Red Otto	Dispelling
Grunt	Fighting
John Paul	BADIALMA
Whizbang	TILTOWAIT
Whoosit	LAHALITO
Gruesome	Avoiding

[Run]   [Attack]

The combat display. Combat is a matter of deciding what action your characters will take and clicking on the "attack" button.

endless periods of time, large portions of the desktop are blank. There must be a better way to handle desktop updating.

Otherwise, Greenberg and Woodhead's design for Mac *Wizardry* shows an almost fanatical attention to detail. There are shortcuts for just about every action you might care to take, a great time saver for an experienced player. Despite the extreme care with which *Wizardry* is programmed, there is one other almost unbelievable instance where the programmers missed a chance to do the game player a favor.

When the party of adventurers steps on some squares in the maze, a message which is supposedly written on the wall or floor (i.e. "Forbidden zone! Turn back!") pops up in an alert box. When this happens, you wait for the disk drive to grind, hear a couple of beeps, wait for the alert box to appear, click in the "continue" button to dismiss the thing, wait for the screen to be redrawn, and activate the maze window before you can proceed. About the third or fourth time you encounter the same square and have to sit through this process, you start to get seriously annoyed. There ought to be a way to turn off these dumb messages.

The last gripe I have has to do with mapping, and this is a criticism of all adventure games, not just *Wizardry*. I hate mapping. No matter how careful you are, your maps are never exactly right, and they take forever to produce; yet, drawing an accurate map is essential to successfully completing *Wizardry: Proving Grounds of the Mad Overlord.* Now that computer memories are getting bigger, it ought to be possible to include some sort of automapping capability in the program. When your party moves through a part of the maze, that area of the corresponding level map could be revealed. You ought to be able to look at the map in an on-screen window and print a copy.

Fascinating, yes; fun, maybe.

I like adventure games, and *Wizardry* is one of the best. Despite their tendency to have somewhat junvenile premises, adventure games are terrific mental exercise. The process of learning the rules of an alternative world and then creating a strategy for coping with those rules is fascinating. However, I find that I prefer science fiction games, because science fiction universes usually behave less randomly than fantasy universes.

I do dislike the randomness. In *Proving Grounds,* you can conduct a strategically sound campaign, but still have everyone in your party destroyed because chance works against you. For example, you spend 20 or 30 minutes creeping up on Werdna's lair. Then, a key member of your party is zapped and drained of precious experience points by a Zombie Lord, making the whole expedition valueless. We're talking frustration here.

At such times, I find myself mentally cursing Werdna (Andrew, backwards) and Trebor (Robert, backwards) and thinking about the blurb on the back of the game box which says something like "hours of fun await you." This is fun? Nevertheless, I usually reboot, restore my characters, and start again. Even more curiously, I've found that though I've conquered Werdna and recovered Trebor's amulet more than once, my interest in *Wizardry* hasn't faded.

This strange, unreasonable compulsion makes me think that *Wizardry* wasn't really programmed and compiled. Instead, *Wizardry: Proving Grounds of the Mad Overlord* must have been grown in a field in northern Mexico and refined in a garage laboratory in upstate New York. Yes, *Wizardry* is a drug.
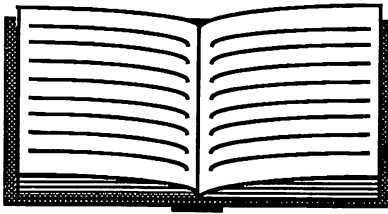
—CLELAND EARLY

Wizardry: Proving Grounds of the Mad Overlord.   Sir-Tech Software.   Requires 128K Macintosh. $59.95.   Apple II version also available.

Special thanks to Sir-Tech Software for the review copy of Wizardry.

How to Write Macintosh Software

by Scott Knaster

Reviewed by Cleland Early

Just as the Macintosh Plus is a pretty-much-state-of-the-art personal computer, Scott Knaster's *How to Write Macintosh Software* is a pretty-much-state-of-the-art computer book. *Macintosh Software* is a friendly overview of many key topics of interest to beginning and intermediate Macintosh programmers.

Knaster, manager of technical support for Apple and an instructor at the MacCollege developer training school, spoke to HAAUG several months ago. At the time, I was impressed by his enthusiasm, good humor, and technical knowledge. Each of these characteristics is evident in the clear, understandable prose of his book. *How to Write Macintosh Software* consists of ten chapters and five appendices. The main body of the book contains three sections—discussions of (1) Macintosh memory management, (2) program debugging, and (3) programming hints.

The debugging section is simply terrific. Knaster first details debugging methods which could be used by any programmer, then he turns to Macintosh-specific techniques. Knaster asserts that the most effective debugging is done at the object code level. He claims that most high-level language programmers can learn enough about assembly language and complied (object) code to be able to figure out what went wrong with their programs. Further, he says that even with a limited knowledge of assembly language, a high-level programmer may actually be able to improve on the object code generated by his compiler.

Optimizing, Knaster says, involves more than just tinkering with the compiled code to remove inefficiencies, however. Because effective memory management is particularly crucial to the Macintosh and other computers which use the 68000 microprocessor,

the programmer must carefully examine how the program uses memory. To improve memory use and optimize code, the programmer needs to use a debugger.

Knaster describes the use of the *TMON* object-code debugger (the Apple *Macsbugs* debuggers are described in an appendix). Before I read Macintosh Software, I had no idea how a debugger worked. Now I think I not only understand how *TMON* works, but believe I could actually use it to improve my programs. For those who have no assembly-language experience, Knaster includes an excellent assembly-language primer in an appendix.

The third section of the book deals with using Toolbox calls in programs. Though Knaster does talk a little bit about plain-vanilla programming using the Toolbox, he's more interested in talking about the facilities, the hooks, the Toolbox programmers provided so that application programmers could customize the toolbox routines. Knaster provides several code listings which could be incorporated into a program and make the resulting application that much better. There are a lot of good ideas here.

Even so, *How to Write Macintosh Software* is not a perfect book. I still have not seen an explanation of Mac memory management that entirely pleases me, though I've read six or seven attempts. Unfortunately, the first section of Knaster's book falls victim to the computer jargon demon: handles, pointers, purgeable resources, implicit and explicit and single- and double-dereferencing, stacks, heaps, master pointers...da, da-da, da-da. Still, Knaster's conversational tone and otherwise sparkling language help make his memory management explanation more palatable than most.

Somewhere I've read that

the average computer owner should expect to spend at least two or three times the initial cost of his computer on software. I think that a similar rule may be applied to programming books; you should expect to spend four or five times the cost of your language system (compiler or interpreter) on supplementary books. These suckers are expensive. Should *How to Write Macintosh Software* be in your collection? For an intelligent Mac owner without a computer background who is interested in doing some relatively serious programming, I can think of no better introduction. Others may want to evaluate their needs more carefully.

If you do buy Knaster's book, you'll probably find yourself constantly referring to it. However, *How to Write Macintosh Software* is not really a reference book. The neatest thing about it is that it can be read. Knaster has a wonderful knack for engaging language. One of the earliest and biggest programmer gripes about the Macintosh was that Inside Macintosh was impossible to read. Mac programming books are much better now; *How to Write Macintosh Software* is evidence of that.

Meeting Dates 1987

●

February 21, 1987
March 21, 1987
April 18, 1987
May 16,1987
June 20,1987
July 18,1987
August 15,1987