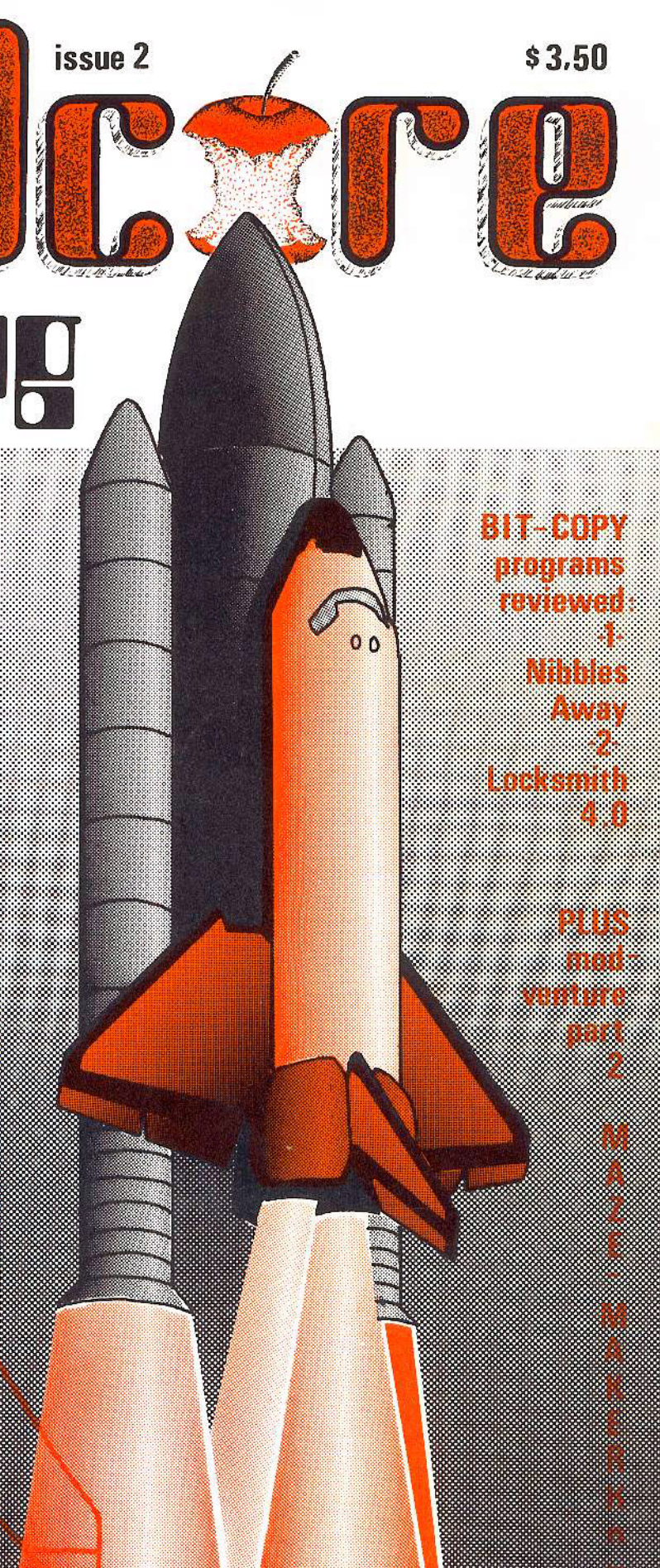


special expanded DISKLOCKS: how to copy-protect disks

volume 1 number 2 issue 2 \$3.50

hardcore

computing



PROGRAMS:

DISKVIEW utility nibbler.
updated DiskEdit 2.2
hi-res: easle.
lo-res: amber's t's
tx-res: invaders.

COLUMNS:

consumer's
legal forum.
HGR: hi-res
corner.

BIT-COPY
programs
reviewed:
+ Nibbles
Away
2
Locksmith
4.0

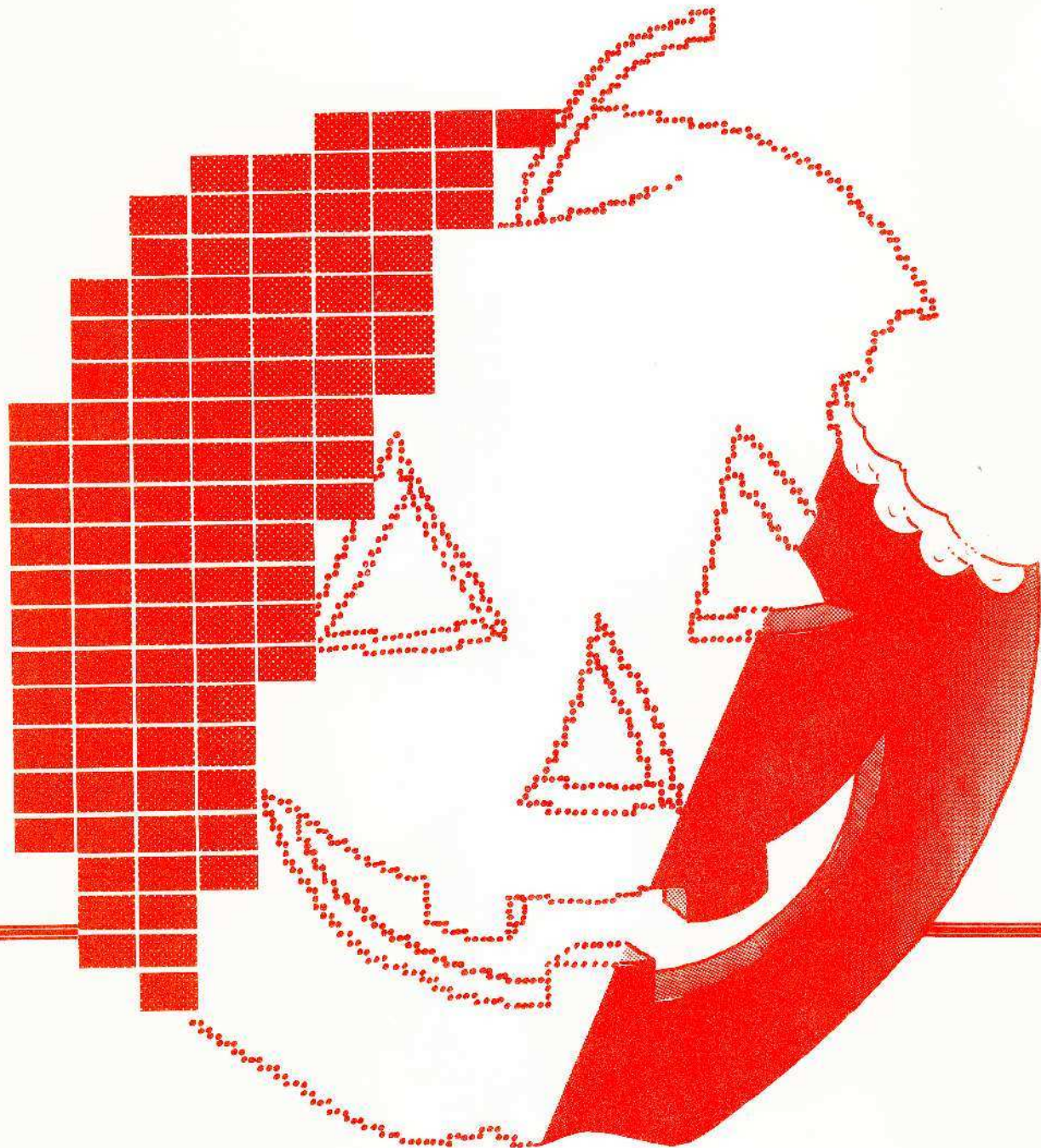
PLUS
mod
venture
part
2

M
A
Z
E
-
M
A
S
T
E
R
P
I
E
C
E
S

about the cover...

HIGHER AND HIGHER ... by ryuji

According to the artist, Higher and Higher refers not only to the rising space shuttle but to the hopes of a graphics resolution even higher than Hi-Res. Here Ryuji shows Columbia rising on her second voyage, just as HardCore publishes its second issue. From Low-Resolution to High-resolution to...reality...Higher and higher...



HARDCORE Computing

P.O. Box 44549

Tacoma, WA 98444

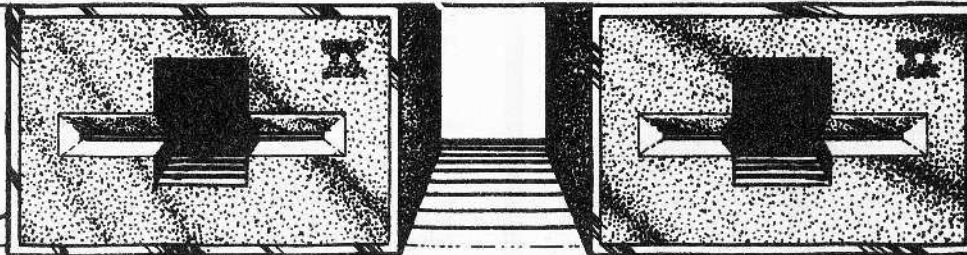
CATALOG ISSUE 2 HARDCORE

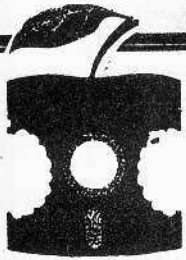
... articles

READER INPUT	4	Letters to HardCore Computing
A Few Words From The Publisher ..	6	HardCore Computing v.s. Copy Protection
EDITORIAL	8	What exactly is a program
Guest Opinion	9	Piracy on the High Keys by Art Cohl
Commentary	10	Software Piracy...the Other Side by Joe Zuis
REBUTTAL	12	A letter from Val Golding
LEGAL FORUM	13	Consumers, Computers, and the Law by Barry D. Bayer
Review	14	Nibbles Away, another 'bit' copier by Karen Fitzpatrick
Using Both Sides of Diskettes ...	16	Review of The Scanner and Disk Prep by Karen Fitzpatrick
Adventure Tips	18	Otopos,Ultima, and the Eamon series by Mike Flynn
Questions and Answers	19	Some problems with Demuffin and Surgery
Modular Adventure-arcade Game ...	20	Introducing the Maze Concepts by B. Bryte
Expanded DISKLOCKS	24	How to copy-protect your disks by Bobby
SOFTKEY part 2	29	How to back up copy-protected disks by Bobby
SoftKey to Integer Programs	30	Integer program pointers by Bobby
New SoftKey MUFFINS	31	How to make Muffin13 and Muffin16 by Bobby
APPLE SOFTIES	32	Text-Invaders 2.0 by Bev R. Haight
CHR\$(4)	42	Utility Nibbler - DiskView 1.0 by Charles R. Haight
Hardware Solutions	45	Curing those auto-start ROM blues by Charles R. Haight
HGR: Huey's Hi-res Corner	46	Artist's Easel by Jack Hewitt
Book Review	51	B.J. Korites: Graphic Software for Microcomputers
My First "Real" Program	52	Amber's T's by Bev R. Haight
Special Listing	60	updated DiskEdit 2.1 by Charles R. Haight
HardCore Program Library	64	the Honor Royalty System

... programs

Utility	DiskEdit 2.1 (A-soft)	a disk editing tool used to "free" locked programs
Utility	DiskView 1.0 (A-soft)	a disk viewing tool used to examine copy-protected disks
Art	Artist's Easel (A-soft)	a screen drawing tool to create graphics in hi-res
Game	Amber's T's (A-soft)	Elementary winner-determining lo-res Tic Tac Toe
Game	Text Invaders 2.0 (A-soft)	Invader-type game played on the Text page
Game	Relief Mapper 1.0 (A-soft)	Module to create 3-D Mountainous Islands





hardcore computing 2.0

Volume 1
Number 2

Publisher: Charles R. Haight
Comptroller: Karen Fitzpatrick

Editor: Bev R. Haight
Graphics: Ryuji

Contributing Editors:
Mike Flynn
Barry D. Bayer
Jack Hewitt
Bobby
Chuck Haight
B Bryte
Bev Haight

Contributing Writers
Joe Zuis
Art Cohl

Published by Softkey Publishing, 14404 East D Street,
Tacoma, WA 98445

Printed by the Peninsula Gateway, P.O. Box 407, Gig Har-
bor, WA 98335

Typesetting by Clark Publishing, 3844 So. Pine, Tacoma,
WA 98409

Editorial address:
HardCore Computing
P.O. Box 44549
Tacoma, WA 98444
U.S.A.

Advertising address:
SoftKey Publishing
14404 East D Street
Tacoma, WA 98445

Hardcore Computing is presently published quarterly (with
monthly updates) by SoftKey Publishing. The entire contents
are copyrighted (c) 1981 with all rights reserved.

Apple usually refers to the Apple II or II+ computer and is a
trademark of Apple Computers, Inc.

..... letters to the editor

All letters we receive addressed or directed to the attention
of the EDITOR are assumed to be "submitted for publication."
Those letters published are chosen as representative of a sample,
or unique in content and may be edited.

..... note to writers

Manuscripts should be printed on white bond and must be
double spaced. Your submission of a manuscript (eg. article, pro-
gram or listing) means that you attest that the manuscript is
original, that you hold all the rights to it, and that you intend to sell
us all rights to articles, and first publication plus first reprint rights
to program listings (which means that disk and cassette copies of
the program are available from our Program Library unless other-
wise stated).

..... self-marketing offer

If you decide to self-market your program, the HARD-
CORE Program Library (as a service to our readers) will be allowed
to sell your program until 1 month after the next issue of HARD-
CORE is published. And as a service to self-marketers, we also will
run a mini-ad (1" high by 2" wide) for you and your program in the
next 3 issues of Hardcore . . . free of charge!

..... keep a copy

We cannot be responsible for lost material nor can we return
manuscripts, disks or tapes unless accompanied by a self-address
sufficiently stamped envelope. (You should always keep a copy of
anything you submit.)

..... product review policy

All samples, complimentary copies and other merchandise
sent in for review become part of the Hardcore Review Inventory-
Library and will not be returned.

If you have a product you think is good enough to be review-
ed by Hardcore then we encourage you to send a copy to either
Hardware Reviews (includes firm and coarseware) or Software
Reviews (includes books and magazines). Please include full
documentation/instructions that your customers would receive
after purchasing your product.

Hardcore sometimes reviews products not received for
review. We also encourage advertisers to send samples of pro-
ducts advertised in Hardcore. Although we do not endorse the pro-
ducts advertised in Hardcore, we do sometimes alert the readers to
particularly bad or good buys. Receipt of a sample does not en-
sure a positive review, nor does the non-receipt lead to a negative
review. We will, however, note whether a sample was supplied for
review.

..... opinions and claims

Disclaimer

All opinions stated in reviews, editorials, and articles are the
products of their authors. Hardcore encourages its writers to be
honest, open, and accurate. All claims made by advertisers are
their own. If, however, you feel you have been wronged by one of
our advertisers, or by any computer-oriented product vendor
please make your concerns known to us. Drop a letter to CRAB-
APPLES (the voice of the consumers).



in this issue ...



Here it is, finally: **HARDCORE 2 with almost twice the pages of the Premier Issue!**
In HC# 2 we welcome several new writers and columnists to the pages of **HARDCORE:**

Jack Hewitt begins his column on Hi-Res graphics with a program called the Easel 2.0.

Barry D. Bayer introduces his forum for software consumers, a column in which purchasers and vendors meet to resolve their conflicts.

Joe Zuis expresses his concerns about, and describes some possible solutions to, the so-called "piracy" problem.

And Art Cohl raps the "original pirates" for "crying all the way to the bank" about profits lost to pirates.

Besides the columns described above, we have a few others that we're introducing: "Apple Softie", and "My First 'Real' Program":

The "Softie" is a program listed in Update 1.1. It's called Text Invaders and for those who doubt that an action graphics game can be played on the text page, have you got a surprise in store for you.

"My First 'Real' Program", in this case, is the editor's own. In this simple low-res tic-tac-toe game, Bev Haight treats the readers to its "search" routine for determining the winner and its interesting colorful graphics. The author shares his opinions on how it can be improved as well as how he'd write such a program now-a-days.

He urges you to submit your own ideas and programs for publication under these two new multi-writer columns.

Of course, Bobby's DiskLocks has been updated to include 3.2 systems (with new tables and charts) to help you understand copy-protection and to show you how to do it yourself.

Mike Flynn is still tromping through adventure after adventure. He introduces the Aemon series.

Continuing his semi-column on writing the ultimate modular customizable adventure, B. Bryte introduces the Maze-Maker modules.

And for those who now have Chuck Haight's DiskEdit 1.0, we include the updated and revised DiskEdit 2.1. Because so much was changed to make it run faster (even with the additional commands), we have printed the entire listing in its own article.

Meanwhile, Chuck's column on DOS, CHR\$(4), presents a new utility program: DiskView, a nibbler!

In our new Rebuttal section, Val Golding responds to points raised in the interview with Dave Alpert.

We have reviews of another Bit-copy program: Nibbles Away

And, finally, we have lots of letters we want to share with you...

soaring to new heights
of freedom...



HardCore Computing

for Apple-users worldwide

the magazine that shows you
HOW TO:

Back up any diskette...
Do & Undo copy-protection...
Encrypt confidential data files...
Customize commercial programs...

PLUS
Game, Utility, Business
and Educational program listings
and...

Hard-core columns on:

D.O.S., Program Tricks,
Writer's Markets, Appledigest,
Software Reviews, Adventure Tips,
plus much more...

Subscriptions...
U.S.A...\$20.00
Canada...\$28.50
S. America...\$37.50
Mex., Central America...\$32.50
All others...\$42.00

HARDCORE Computing
Dept. 222
14404 East "D" Street
Tacoma, WA 98445
U.S.A.

Apple is a trademark of
Apple Computers, Inc.

Dealers inquiries invited

Reader Input

Dear Sirs,

I have notified my post office that I find the enclosed material pornographic and extremely offensive. Please refrain from ever polluting the American postal system with your distasteful magazine (?) again.

Thank you,
ON-LINE Systems

Dearest friends,

It's about time! A forum for questions (and answers) that other journals won't even talk about. For example, why has Apple Computer not updated their FORTRAN Compiler rather than providing the inconvenient FORT-FIX program? Because the compiler is copy 'protected' and the portion of the compiler that needs fixing is on the protected (ie. offset) track five. A true repair to the compiler would require giving everyone a new pair of disks ("too expensive!") or revealing the protection scheme in a program to fix the bad segment ("no way, Jack!"). An expensive software tool is rendered next to useless because of greed.

I have no patience with software that can not be backed-up, repaired, customized, or used to learn. I am tired of magazines that are thinly disguised software catalogs. I am angry that information is suppressed for profit's sake. What if scientists tried to conceal the methods used in their research? Where would we be if Sir Alexander Fleming (of Penicillin fame) had thought only of getting rich quickly?

My local computer club gave me your address and permitted me to review their archival copy of your premier issue. They are very much opposed to unauthorized duplication of copyrighted material, but they do not presume that the First Amendment to the Constitution was repealed with the advent of the home computer.

I am behind you. Enclosed is \$50. Please enter my subscription. I would desperately like to have my own copy of your premier issue. Since I have nothing to advertise, the balance should be considered "user support".

Sincerely,
John B. Matthews, M.D.
Dayton, OH

The staff of HardCore Computing would like to thank all those who wrote us, even if your response to our magazine and its policies were negative. Most of the letters we received were positive and suggested improvements, asked questions, and gave opinions. We appreciate all of those. HardCore is in the important formative stages of growth and your correspondence do aid us in making editorial decisions. So keep those letters coming!

===bev.

Dear Bev,

I received the first issue of your magazine and was delighted! May your efforts be rewarded, and may your magazine prosper.

However, I predict that your magazine won't last a year! Part of the reason being of course, that the computer fraternity will not take kindly to you. Quite frankly, however, the seeds of your destruction are of your own making.

1. ADDRESS: I had to search through your magazine to find your address. It should have been inside the front cover.

2. ATTITUDE: Your maverick, "damn everybody else" attitude offends me; and I happen to agree with your viewpoint. You can say what you want to say without repeating over and over that the rest of the computer magazines are indulging in censorship. They're just protecting their interests! Disagree with them, please do... but don't make them out to be worse than they are; it cheapens you.

3. CURRENCY: You wrote a fine article on Akalabeth, assuming that only the 3.2 version was available. I have the 3.3 version and so your major article was of no value to me. I do not have the Programmers Aid chip and so was not able to use Demuffin Plus.

4. PUBLICITY: You are going to need publicity and advertising to survive. I doubt that you will attract advertising as long as you adopt your "underground" style of editorializing (?). However, and if you ignore the rest of the letter, pay heed to this: if you can not get publicity any other way, try this. Prepare a DOS utility that really does it all. Good documentation and outstanding advertising are essential. Price it lower than competitive utilities on the market. Then with every order, ship out a sample copy of your magazine together with a subscription form.

I really do wish you every success in your venture, and hope that my pessimistic prediction be proved totally wrong. Looking forward to receiving future issues.

Yours truly,
Terry Cashin
Burnaby, B.C.

Gentlemen:

Although it seems at times like Mr. Bev Haight protests overmuch about the goliaths of the software industry manipulating the other computer magazines, I do think he has a point.

It makes me wonder why I have never read how to get into a "locked" disk program that cannot normally be listed. This information is certainly known by better programmers, yet I have never seen it in other publications. After using your "open heart surgery" method we were able to get into the Temple of Apshai program and the Computer Ambush program.

My 14-year-old son is the one who benefits from this. He is now able to view the program listing, make changes and, as he puts it, "cheat" to make it easier to win the game. All of these things help him LEARN how to better use the computer. He can learn programming techniques used by the professionals. Hurrah for Hardcore for giving us this opportunity! Keep the information coming. I will certainly recommend you to my Apple user friends. I look forward to your next issue which you indicate will tell how to get into Integer and Binary programs that are "locked" and unlistable.

Yours truly,
Gary L. Schmitt
Wheaton, IL

Gentlemen,

I have no idea whether or not I will like your magazine, but anyone censored for intellectual reasons can't be all bad.

W. Buchman
Los Angeles, CA

Dear Charles and Bev Haight,

I liked the following articles:

1. DiskLock Update: How to copy-protect your disks.
2. Bit Copy Programs.
3. SoftKeys to Akalabeth and AppleSoft disks.
4. CHR\$(4) Utility RWTS: DiskEdit.
5. DiskLocks: How to copy-protect your disks.

Most of the other articles did not appeal to me. People are subscribing to your magazine to learn how to crack copy-protected disks like your advertising suggests. You

errors

I failed to catch an error in Issue #1, corrected it in an Alert and again in the Update. I then discovered that there was an error in my correction, and I corrected it in later printings of the Update. This time I'll get it right. Here are the corrections to the article on DEMUFFIN PLUS, steps b and c.

- b. 1900 < B800.BFFF ctrl Y* return
- c. 1900 < B800.BA10 ctrl Y return

KK

& apologies

Welcome to the editor's apology...

Both my brother, Chuck, and I are new to the magazine publishing business and there are many "tricks of the trade" that we must learn in order to run this magazine efficiently.

Beginning on a shoestring, we managed to gather a small staff of computerists who had skills that this magazine needed and we put out the first issue. All were volunteers, of course, working fulltime at other jobs and spending their free time making this magazine as professional as they could. We promised to be a regular quarterly with monthly updates on important columns. In this we have failed. The magazine took more time than we had and we underestimated the negative, anti-"hardcore" attitude of the major software houses. This realization has discouraged some and instilled more fighting spirit in others. Added to these problems was our inexperience in the publishing end of computing. And the result was a very late printing of issue #2. And just as the two Updates following issue #1 was combined into one, we find that we will be forced to do the same for the two Updates following issue #2.

Like many businesses in the expanding microcomputer field, we operate from a residence with the hope that we may soon be able to afford an office, and a real salary for the volunteer staff. So far, the only people who have been paid are the printers, typesetters and graphic designers who make the magazine the professional appearing periodical that it is. The actual staff however need much more experience in publishing... experience they are fast accumulating as our readership grows and our subscribers increase. We still need more manuscripts that fit the publishers guidelines and we hope to be able to pay competitive amounts soon.

Thank-you for your patience, understanding, and especially your support as we battle for a place among the national computer magazines. We really do try harder!

continued on page 35

HardCore Computing

There are many varied opinions on this topic

Last issue, the editor made clear his opinion of copy-protection. I did not then express mine. And some of our readers have mistakenly assumed that my opinion coincides with his. It does not.

The editor and I agree on many points, enough to collaborate and produce this magazine. However, we do disagree on other points. Copy-protection is one of these. The editor is a hard-core "First Amendment Advocate." To him, programs are just another form of literature, specifically: instructions derived from algorithms and written in a way that a device can interpret and carry out those instructions . . . much like the holes in a player-piano's music roll is a program for the piano to play particular musical pieces.

I am not an editor, certainly not a journalist. I am, instead, a computerist, a programmer, and a businessman. Copy-protection, to me, is simply a means of assuring greater monetary compensation for a publisher's venture capital and development expenses as well some royalty for the authors. In a market where the viable revenue-producing life of a particular product, specifically a mass-marketed computer program, is measured in weeks, it is important to assure widespread publicity and distribution of such a product before that product is made universally available via personal exchanges. **Game programs have the shortest profitable lifespan. For these ephemerally profitable products, I think that copy-protection is a necessity. It prolongs the profitable lifespan of the product and there is no great need to alter the product.**

Of course my editor disagrees. "Any form of so-called copy-protection," he argues, "that hinders that program's listing, modification and subsequent re-copying, is an affront to the software consumership and should not be tolerated in America!" He is quite verbal and outspoken on this issue, and just as dogmatic as the software companies that defend their "anti-piracy" schemes.


For those who agree or disagree with the editor, be sure to direct your arguments toward him. My views are not exactly his.

If copy-protection is going to remain as the industry's standard-operating-procedure, then consumers should focus their attention on the back-up and update policies of the software vendors, their warranties and their guarantees. After all, if we cannot make copies or modify these programs, we should demand that the industry have reasonable and adequate policies to back-up crashed disks and to provide updated versions when improvements are necessary. We should also demand product excellence since we are not given the option of de-bugging and improving them ourselves.

In other words, the present "buyer beware" syndrome is not good for business and is certainly not good for the consumer.

Although I agree that in some instances copy-protection is necessary, I do not agree with the present situation where computerists are denied useful information because the software industry has undue influence and control of the sources of information, specifically, the magazines, the computer stores and some major "user's" clubs. Both the editor and I agree on that one disgusting fact. Somehow, the users must be made aware of the present situation, a condition of ignorance created by members of the software elite who now own or operate software houses and are in control of computer clubs, stores, and magazines. This condition is perpetuated by users who condone or tolerate censorship and information suppression in the guise of "protecting the public" or "combating piracy."

I am convinced that in many clubs, this software elite is using censorship and information suppression in order to maintain what my editor calls the "conspiracy of silence" that now reigns supreme. Unless the members are made aware that they are being manipulated, and unless they stand up and speak out against these "agents of ignorance," the strangle hold on knowledge will tighten until a chasm forms . . . with those "in the know" on one side, and those "hoodwinked and manipulated puppets" on the other. And you can guess who will be using and abusing the other.



a
few
words
from
the
publisher

v.s. Copy-Protection

even among the members of our editorial staff

That is the reason why we formed this magazine. We have created a bridge over that growing rift between the sellers and the buyers, the manipulators and the manipulated, because the only difference between the two is: information, specifically knowledge of copy-protection.

This magazine is not out to "get" any software company. HardCore is a medium for the expression and distribution of ideas, opinions, directions . . . some of which are, unfortunately, barred from expression in the other major computer magazines.

HardCore is not a "pirate's" magazine any more than a pharmacist's journal is a "drug-abuser's" magazine or a periodical on guns is an "armed-robber's" magazine.

That we are called such is a very real measure of the fear that the information in our magazine will free the user from the dungeon that the sellers have created. It is the fear that censorship and information suppression is effective only so long as there are no "security leaks of secret information."

HardCore is not in the business of removing copy-protection from specific programs because we do not have the time. The SoftKey to Akalabeth was used simply as an example of how certain techniques are done. Such SoftKeys will be published only when accompanied by a legitimate "Fix" or program improvement. I feel that SoftKeying every single program is useless since all that needs to be done to re-protect the product is to utilize another protection method. A better way to help readers is to show them how such "protection" is accomplished. By removing the air of magic (wonder and ignorance) from copy-protection, readers can analyze and down-load their own programs irregardless of how the vendors may alter their schemes.

HardCore is against commercial piracy in any of its forms. However, my editor feels that personal exchanges, especially of modified commercial programs, is the right of the buyer. I feel that casual exchanges do hurt the market but no more than book and record exchanges (including personal recordings from copyrighted releases) hurt the book and music business.

I do not advocate nor do I approve of commercial piracy. However, it is impossible to halt exchanges between computerists simply because the duplication process is far too easy and commonplace to prevent. I feel businesses should cease their expensive efforts to battle such exchanges and instead take advantage of them. There is little else that can be done while still maintaining a cordial consumer-vendor relationship (a relationship that is rapidly deteriorating since the advent of copy-protection). Many feel that such a "lock" is unethical and that it is their duty to free it! Some, like my editor, feel that "any business scheme that requires, for its survival, massive media censorship and suppression of information (such as copy-protection) is an abomination to the free market system and is a threat to the American independent free spirit" (Please excuse his frequent flag-waving but, as I explained early on, he is a free-speech, free-expression radical).

I sympathize with both the consumer and the vendors, publishers, and distributors. That is why I am offering an alternative: "piratable" software. For more information on this, see the article on "HardCore Program Library Honor Royalty Diskettes." With these diskettes, computerists are urged to exchange them . . . with a catch: They are also urged to send the buck-a-program-copied directly to the author as royalty. The reason the "price" of a copy is only a buck is that there is no advertising cost and no copy-protection expense added in since the person exchanging the program is actually the advertising medium. Whether or not this method will succeed is debatable, but one thing is for sure . . . with this method in prevalence, there will be a marked drop in "piracy" simply because there will only be those who have and those who haven't honored the author with the token royalty payment of a buck-a-program-copy.

I welcome responses to my own opinions.

Charles R. Haight,
Publisher

FINANCIAL MANAGEMENT SYSTEM II

A HOME AND BUSINESS
ACCOUNTING SYSTEM
FAST - EASY TO USE

Enter an entire month's CHECKING, CHARGE CARD, and CASH accounts in just a few minutes using your own personalized macro lists. INSTANT ERROR CORRECTION. Audit all files by Code and Month with year-to-date totals. PERFECT FOR TAX ACCOUNTING.

- SELF PROMPTING, ERROR AVOIDING ENTRY SYSTEM. 1 to 3 KEY-STROKE entries with AUTOMATIC DATE, CODING and NUMBERING.
- SCREEN and PRINTER routines for listing files, balance, reconcile, search, and audit reports. Configure system to match almost any printer.
- BUDGET MANAGER to plan, review and balance your budget, generate COMPLETE reports for any 1 - 12 month period.
- SEARCH all files with specific and expanded searches.
- ACCOUNT AUDITOR totals all files by Code and any 1 - 12 month period with year-to-date totals.
- Enter your own Item and Code Macros, up to 100 each.
- 48K with ROM APPLESOFT and DISK required. SINGLE or DUAL DISK compatible. (printer optional)
- PRICE \$39.95

*****NOW AVAILABLE*****
CHECK WRITER II for FINANCIAL MANAGEMENT SYSTEM II. Print your checks directly from FMS II files.

- PRICE \$19.95 - (\$15.00 if ordered with FMS II)

GROCERY LIST

A USEFUL HOUSEHOLD PROGRAM
DESIGNED TO ORGANIZE SUPERMARKET
SHOPPING

Shoppers will INSTANTLY be able to use this easy, self-prompting program. Scan a file of up to 500 USER DEFINED ITEMS. Choose those needed with a single key-stroke. Then print a shopping list ORGANIZED BY TABLE NUMBER, SECTION, or four letter code such as "DARY", "BAKE", or "DELI".

- 32 - 48K with disk and printer required, (APPLESOFT)
- PRICE \$19.95

COMPUTERIZED MANAGEMENT SYSTEMS

by D R JARVIS COMPUTING INC.
1039 Cadiz Dr. - Simi, CA 93065
PHONE (805) 526-0151

Check, VISA or MASTER CARD accepted.
DEALER INQUIRIES INVITED

What exactly is a program?

e
d
i
t
o
r
i
a
l

Micro-computing is a rapidly expanding field that is opening up marketing frontiers and some thorny legal wildernesses only imagined a decade ago. New definitions and laws must be created to handle the problems emerging as users and producers struggle to shape this new market of intellectual creations we call: computer software.

Embodied in what we call a program, this product of intellect has the potential to renovate our entire concept of information and its exchange.

Is a program merely a collection of equations, public domain information garnered through mathematical-scientific research and hence both unpatentable and uncopyrightable.

Is a program a device, like holes punched in a card (template) that allows a machine to accomplish certain tasks.

Is a program merely a set of instructions which, because they are only directions that tell you how to do things, are copyrightable only as directions and not as something more: an idea?

Or is a program the written expression of something more intangible, the reaching for a goal, the completion of a task, a process?

Can it, or should it, be patentable, as a concept that lets you reach that goal?

In other words, should programs be copyrighted or patented?

Or should programs have a separate, new category of registration that combines the two aspects in a technological hybrid that a program truly is?

I believe that this last alternative is one that will benefit both the computer user and the world as a whole.

An entirely new category should be evolved: one that reflects the true nature of computer programs, one that recognizes both the patentability and copyrightability of certain facets of a program as well as the protection of the visual display.

An entirely new category should be evolved:

one that reflects the true nature of computer programs,
one that recognizes both the patentability and copyrightability of certain facets of a program as well as the protection of the visual display.

one that recognizes that, regardless of the language it is written in, the essence of the program is the same whether it is in FORTRAN or BASIC or some other language,

one that will give the author certain rights on the commercial use of that program, but also

one that recognizes the lesser but just as important rights of the author who translates it into other languages.

It should be one that, like the patent laws, recognizes that certain ideas, or computer tasks, are commonplace or necessary and can be derived by a person knowledgeable in that field and therefore are not patentable, but that some tasks and their solutions are unique and should be registered as such.

(While copy-protection, that is, protection from direct duplication, is a major concern among vendors, I am not referring to this form of protection. Rather, I am discussing the stealing of entire ideas and their modes of expression. It is that type of copy-protection that must be addressed by the legal community.)

This is especially important in the lucrative field of games. Consider the espionage and idea-thefts occurring in the board games industry . . . and you can see it happening already in computer games). Atari has made it clear that it considers the visual presentation of its games as copyrighted. If you write or market a program that looks like an Atari game, then you may have violated their copyrights. Of course, its enforcement is made difficult because of the still debatable legal classification of programs.

In making these new definitions and regulations, we should not forget the user, the consumer. Programs in this new category must be recognized as devices as well as literature, devices that can and should be alterable and repairable by the consumer; just as you can alter any physical product you purchase. We should not be forced by laws to depend entirely upon the vendor's updates for improved programs. We should be allowed to improve them ourselves!

Owners should be able to sell that altered product just as owners can sell any other device they own.

And because the programs can be so easily and inexpensively duplicated, it is imperative that this new category or classification be instituted in such a way that the consumer's right to make archival copies is not trampled in the producers' haste to protect programs from copyright infringements.

It's a brave new world out there and we should not let the vendors, marketers and writers of programs create laws that will not be in the interests of the consumers.

Unless we view programs in this new and separate category, the consumers will be the losers.



Let's all become Disney-Universal pirates

Our individual rights are being eroded again. Did you hear that the 9th U.S. Circuit Court of Appeals in San Francisco has ruled that the users (and the makers and distributors) of the video-cassette recorders (VCRs) are breaking copyright laws if the machine is used to tape over-the-air broadcasts of programs that are copyrighted (most commercial broadcasts are, by the way).

That means that if you want to watch a program on TV but cannot be there at the time it is shown, then it is illegal for you to tape it for later viewing! And if you do, then you are a PIRATE. Outrageous!

This is just part 2 of the battle between Sony and the team of Disney Studios and Universal Pictures whose goal is to make restrictions on what you can do with your own TV set in your own home.

But according to an AP release, industry officials and analysts predict that Americans will go on doing so anyway, whether it is illegal or not. Those are my kind of Pirates, my kind of Americans.

Of course that decision will be appealed all the way up to the Supreme Court. Meanwhile, there's a move to amend the

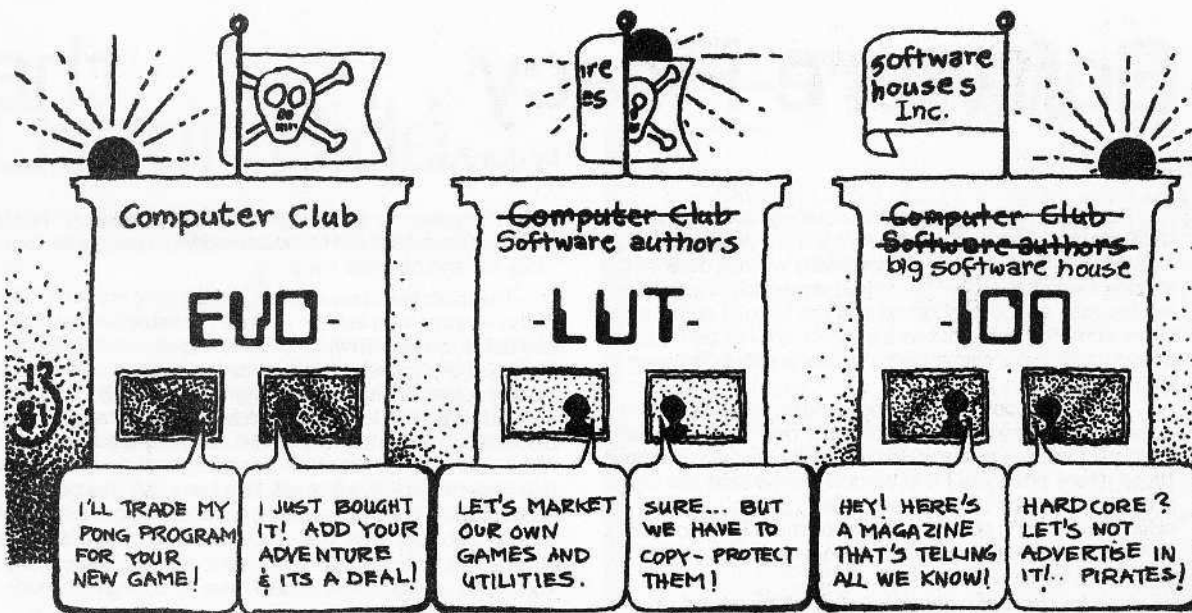
copyright laws in order to legalize the sale and use of the VCR for personal, non-commercial copying. I agree with this legislation, of course.

Could such court decisions be eroding your individual rights right now?

Could industry tell you what you can or cannot do in your own home? It's happening right now. Another example that comes to mind is a move by the recording industry in England to put a tax on blank recording tape and give the tax collected to the recording industry to make up for revenue lost by people recording music off the radio! Talk about government hand-outs!

Could this happen in America? Could Sony be forced to pay a tax or penalty fee for each Betamax sold, a fee to be donated to the "poor, penniless magnates of the woefully endangered" movie-TV industry? They could. If you stand by and let them. And if you didn't know it was happening, then so much the better for Disney and Universal.

So be warned. Everyone who uses a video recorder to record off commercial TV is now a Pirate, too. Welcome, pirates, to the ranks of your fellow independent Americans.



Piracy On The High Keys

by Art Cohl

The word "Pirate" is being thrown around quite a bit these days. Some software companies would like you to believe that if you trade software you are a Rogue, a Criminal, a Despicable Thing.

What they neglect to realize is that we, the computer owners and users, have rights, too. We have the right to buy software and what we do in the privacy of our homes is none of their business.

Actually, the software companies are in business because of us. We keep them in business and we can run them out of business. So where do these people come off calling us Pirates? The accepted definition of "Pirate" in the record industry, video tape industry, and all other industries is: a person or company who makes many copies of somebody else's work and then sells it, taking the profit for themselves. By this definition, I have never met a pirate.

The main consideration here is money. The software companies are pulling in huge amounts of money each month. Otherwise they would get out of the business. They are not stupid.

Then why the "hue and cry"? Because instead of making small fortunes, they want to make large fortunes, so they cry all the way to the bank. Poor babies!!!

One question always pops into my mind. Where are the original "Pirates"?...the ones who broke and traded programs a year or two ago? Answer: They are now running the software companies!! They are the programmers and businessmen of those companies!! In other words, it was alright to trade programs when they did it but, now that they are in the business and making money from software, suddenly it is wrong to trade.

Another thought is...Why did people start copying software? Perhaps if software was reasonably priced (greed again), and all programs on the market were worth the money and were error-free, there would be less of the so-called piracy. So who is to blame for the high prices? Who is to blame for the poor quality software on the market? Most of us have been taken at one time or another. Has anyone ever seen or heard of any attempt by a software company or computer magazine to police the market or criticize a competitor selling garbage????

Speaking of computer magazines, the "impartial voice of the industry"...what are they doing? Are they presenting both sides of the conflict? Are they being honest and deserving of respect? Not just NO, but HELL NO! They support the software companies. They present only one side of the argument and continue to advertise garbage programs. So much for integrity. If some users got together and agreed to buy just one copy of a magazine and made copies for all the others, then I could understand the magazines complaining. Software is none of their business unless they sell software themselves, in which case there may be a moral conflict of interests.

So why do the magazines keep calling us names? The answer is the almighty buck. They must cowtow to the software companies so as to get their advertising. They follow like spineless sheep licking the boots of whoever pays the most. Sickening, isn't it?

I would like to state that not all software companies and magazines are the targets of this article. Any magazine that would publish this is obviously not included. As far as software companies — those in the magazines are excluded by virtue of the fact that they advertise in **HARDCORE**. You know what I mean.

What it all boils down to is which side of the fence you are on...buying or selling software. Someone who is considered a pirate by some may be considered a fighter, buccaneer, or "admiral-able" by others. If the software companies and magazines continue their mud-slinging perhaps we might think of a few names to call them. Words like Garbagemen, Junkmen, Cheap Chiselers, Hypocrites, Conmen, and Thieves might apply to some. Yellow Journalism, Toilet Paper, or Comic books might apply to others.

And last but not least, if some still continue to use nasty names and insults, the computer users might use their ultimate weapon. He could stop buying their products. Wouldn't it be nice to have an Independence week or month or year when nobody buys their software? Then what would those software companies say???



Software Piracy the

by Joe Zuis

guest

commentary

If one believed the various articles about software piracy, then one must feel that the vendors are in a desperate situation. Most editorials point out the many wrongs done to the software companies by the consumers which, if continued, would justify additional protection in the form of stiffer laws. While admitting that software vendors are not perfect, the articles place the greatest blame for the present problems on their customers.

Yet the situation does not appear to be as claimed, for the number of programmers entering the market is increasing [just check the size of the advertising in the magazines writing those piracy editorials.] It is hard to believe that the newer programmers don't know about the "rip-offs" or that the older ones enjoy it and come back for more [masochistic?]. The whole situation sounds more like a case of crying all the way to the bank.

Besides, the software vendor has piled some of the problems onto themselves since it is their policies which have forced people to seek alternatives. The use of copy protection without offering a reasonable backup is one of the biggest complaints of the present. People cannot count on any company staying in business forever or making changes in the software to cover special applications. In addition, the "juggling act" that one must go through when switching disks is not very funny.

The best answer to the piracy issue would be to find other similar industries that have faced the same problems. Such industries include the book and record businesses. With sales of over 200,000 Apple systems, the software market is comparable in size to that of either books or records. Also, few can disagree with the fact that the amount of time spent on writing a book is no less than that spent on writing software - in case you doubt this, consider the work behind researching, rewriting, and proofreading.

But the biggest similarity between the three industries is the one we are most interested in, namely the piracy problem which they all have faced. Record sales were threatened by the low cost of blank magnetic tape, while the book industry suffered losses due to the practice of trading books instead of purchasing them. In both markets when the first copying mediums became available, there were the same howls of rage as are echoing around at present.

Yet, while the three industries have similarities, it is their differences that are the most noteworthy, particularly the mature attitude of the book and record businesses toward their customers. The only pirates that are prosecuted are the large scale operations with little attention paid toward the individuals who copy rather than buy. Due to the availability of cheap alternatives, the industries were forced into developing lower-priced products in order to obtain a high sales volume, with the higher prices going to the "limited market" items that are highly specialized.

The greatest difference is the accessibility of information, for in a book it can neither be hidden nor restricted. One can create a new or improved version of any idea with the general limitation that the idea's source is identified. This easy use of information is sweetened by the fact that far more information is obtained from a book than from a protected disk per dollar spent.

The attempt to hide information through the use of copy protection is probably the biggest exercise in futility at the present. There are more people breaking protection schemes than there are creating them, and there always will be more as long as people have more time than money. Besides, there isn't a scheme that can't be decoded by watching the data/address lines of the microprocessor with a logic analyzer - not expensive if the computer is converted into one. Of course if the protection schemes become too costly or incompetent,

then the consumer can always use his best weapon - no purchases. The low sales of computer systems using a hardware "lock in" should prove my point.

The most ludicrous attitude in the piracy problem is that of the vendors who ask for all kinds of restrictions and laws, only to issue a disclaimer or a "as is" clause on their product. They give the impression of wanting everything without taking any responsibility for anything. One is not likely to buy a product with such limitations unless its price is appropriately discounted. Also imagine a record company telling you that a record can only be played on one specific player - those with two players would have to buy two records! This type of attitude is what helps to cause the nonchalant posture of many pirates because it is the actual policy of some software houses. So while I agree a change is needed, I would like to know why only one side should have to change its mode of operation.

SOLUTIONS FOR THE CONSUMERS;

1

Along with the criticism, I have advice to offer as well. The most important thing that people can do is to write letters to the vendors, stating any complaints as well as any intentions of not buying a product until the complaints are answered. Copying a program does not convey the proper message or indicate the true number of dissatisfied potential customers. If enough people expressed an interest, a com-

TEN - copy protection - COMMANDMENTS

Thou Shalt:

1. NOT EXPECT CUSTOMIZABLE PROGRAMS for that means that thou should write your own.
2. NOT RETURN PROGRAMS IF NOT SATISFIED with product for thou may have made a copy of it.
3. NOT JOIN ANY COMPUTER CLUB not controlled by a software house or you shall burn in Micro-hell.
4. NOT MAKE BACK-UPS of copy-protected programs for then thou shalt be a vile and wicked sinner.
5. NOT ALTER PROGRAMS to fit thy needs for to do so thou must break this holy lock and that is a sin most evil.
6. NOT LEND PROGRAMS TO OTHERS for all others are Pirates.
7. NOT TAMPER WITH THIS HOLY LOCK or the program will destroy itself.
8. NOT COMPLAIN ABOUT THE LACK OF WARRANTIES, guarantees, or other promises given by other industries.
9. NOT EXPECT LOW-COST UPDATES and corrections for you are our sheep and we are fleeing you.
10. NOT BUY, READ, OR TALK TO ANYONE ABOUT HARDWARE COMPUTING.

by the editor...

Other Side

pany would be more inclined to respond in order to obtain the sales.

2

Another meaningful step would be to establish magazine forums where people can list the desired features of any type of program. By openly discussing the good and bad points of each feature, people can learn from the more experienced programmers, and at the same time establish a list of standard features to be used as a guideline by the programmers. A readers' poll at the end of a time limit would indicate the most desired features as well as the number of prospective buyers.

3

The forum could also be used to standardize reviews by indicating missing or confusing information with the intention of eliminating some of the bias or opinions. The features listed in the forum could be used as a standard with more time spent on reviewing the limitations or extra aspects of the program. Hopefully this could make choosing software a little easier.

4

Software companies should be encouraged to use magazine articles to answer customer questions instead of using the present system of individual responses. If the documentation listed the magazines in which to watch for their articles, more people could be reached with less necessity for answering the same questions repetitiously. As a side benefit, the vendor would be paid for answering his customer's inquiries. But if the article becomes an advertisement, it would only turn people away or cause them to go back to expecting individual answers.

5

The various clubs could coordinate their software purchases so that each one buys a different package of a given category. With swapping between clubs, people can inexpensively try the various products, and decide for themselves what is best without depending on someone else's opinion. Also, a "group buy" could be arranged for the best possible price break.

6

If the situation does not improve, then the customers should organize, just as the programmers are doing in order to lobby for fair changes in the laws.

7

As an alternative, the group could establish a list of features and hire a programmer to create the desired routines. The obvious advantage would be that the group would own the source code, eliminating the hassles caused by copy-protected programs. Modifications would be easier and cheaper, while programming techniques could be observed and improved. The cost to a large organization would be less than on the open market, and the programmer would know exactly how much he is earning.

These suggestions may not be the best ones, but they represent an attempt at showing that there are alternatives. The important thing is to not accept any solution that is unfair to either the majority or the minority of the people. There are reasonable laws and sensible approaches, but it is the continuation of the ridiculous ones that will cause an eventual rift.



THANKS

The editorial staff of HardCore Computing as well as the staff of SoftKey Publishing would like to thank all our readers who have supported us in our attempt to become a leading professional computer magazine. HardCore began as an idea expressed in two small ads that we carefully worded so that they would get by the ad censors. It was published in Call A.P.P.L.E. and in Softtalk magazines and it brought us about 200 subscribers and several ads that were printed in the premier issue. We also sent out junk mail flyers to addresses and names we got from other magazine pages.

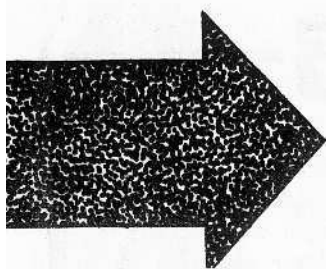
It was then that we actually put together the magazine itself. Little did we realize how costly and time consuming that process would be. We got help from professional printers and graphic artists. We commissioned an artist to do the cover. And we got the first issue out, late. Only 36 pages, it was our best effort.

Our second issue, we vowed, would be 64 pages. The same artist, now a regular staff member, did the cover again and redesigned the interior. We got together new writers, and the second issue was moving smoothly, but even more late. Then we had trouble with our computer and printer, and had to rely on our local friends' equipment. All this was done during our free time, of course. Then, suddenly, several members had to leave and we were short of help. But we did what we could. Now HardCore #2 is out and we're working on #3. Financially, HardCore was still in the red, but our subscribers now have edged over the 1000 mark. We have been inundated by telephone calls... and we are out of the "office" most of the time, anyway, so we haven't been able to answer many of them. Sorry.

Those of you who are editors of club newsletters and magazines know what we mean. We are all still basically computerists who have tired of the way other computer magazines have been handling information that Apple-ites desire and have tried to do something about it...

So please be patient with us, and in the beginning judge us not by our late arrivals but by the information we have presented in our pages... because that has always been our priority.

Issue #3 will appear in February, instead of January. And, instead of two updates between issues, there will be only one. We hope to have at least 64 pages, more programs, and many more articles (fewer on opinions, arguments, etc, though), programming aids and helpful hints. Happy Holidays!



rebuttal

"Rebuttal" is a column in which readers can express opinions contrary to those presented in prior articles. This column gives readers a chance to hear more than one side of any given argument. Rebuttals are usually letters directed to HardCore whose content reveals new and useful information on any controversy presented in an earlier issue.

We kick off this column with a letter from Val Golding. The editor solicited his opinions and views, and they are presented here unedited. Mr. Golding takes issue with the editor's interview with Dave Alpert of Omega Microware. Mr. Golding also expresses his views on copy-protection, pirates, the bit-copy controversy and HardCore's Editorial policy.

a letter from: Val Golding

Val J. Golding
6708 39th Ave. S.W.
Seattle, WA 98136
July 23, 1981

Dear Mr. Haight

I thank you for the complimentary copy of the June, 1981 HARDCORE, and the opportunity to respond to your articles and editorials.

I would like to be able to express my wishes for the success of HARDCORE, because I believe, in general, that the more information that appears in printed form, the more the Apple user will benefit, **but I cannot because I feel the policies you encourage are to the ultimate detriment of the user.** I would like to add, also, that the opinions expressed in this letter are my own, and may or may not coincide with those of Apple Pugetsound Program Library Exchange, the publishers of call-A.P.P.L.E.

I think that it is a sad commentary on the state of reportage — witness the Washington Post episode — that those in positions of responsibility are not always making the necessary checks to ascertain the accuracy of their reporting, particularly when that reporting has the potential of irretrievable damage to the credibility and livelihood of individuals and organizations.

I am referring, of course, to your interview with Dave Alpert, which contained many inaccuracies, and in addition alluded to facts which Mr. Alpert was not privy to. In a court of law, the statements made by Mr. Alpert would not be admitted without verification by other witnesses; they would be considered hearsay, third party conversations. In each instance, Mr. Silverman and Mr. Golding would be called as

witnesses to relate the facts as they understood them. This is what we strive for in good reporting, accuracy and verification from as many sources as possible of each alleged fact. You are asking your readers to draw conclusions based on hearing the evidence presented only by the prosecution.

While the damage has already been done, I would like to correct some of the impressions that were made in the interview with Mr. Alpert. In his second response, he accuses me (along with Bob Tripp, editor of Micro) of telling people that they would not be able to get advertising if they ran the Locksmith ad. This is completely untrue; I made no such statement, nor did I imply that that possibility existed, although I believe it did. Mr. Alpert then states I threatened to "destroy" the International Apple Core newsletter by pulling the material submitted by A.P.P.L.E. if the Locksmith ad was accepted for publication. This, too, is inaccurate. As an individual (and I do feel very strongly on this volatile subject) I did say that I would not allow any stories with my byline to be printed. In addition, I also indicated that A.P.P.L.E. might consider as a possible action, the withdrawal of material contributed by them. This action was never formally considered or approved.

Mr. Alpert states that I have "since been replaced" (as editor of the Apple Orchard) with the innuendo that I was being replaced because of my position on the ad and/or threatening to withdraw my stories. Here, Mr. Alpert is discussing a subject in which he is not in possession of all the pertinent facts. The subject of my resignation from the position of editor of the Apple Orchard was opened in correspondence with Ken Silverman, president of the I.A.C., long before the subject of Locksmith

came into the limelight. It was based on my growing dissatisfaction with the direction in which the magazine was proceeding, the lack of cooperation from the publishers, and the pressure of other business which prevented me from devoting the necessary time to the editing of the Apple Orchard. I, in fact received a beautiful wall plaque from the members and officers of the International Apple Core expressing their "gratitude and thanks...for serving as Editor", and I further continue as "Editorial Associate" and contributor to the Orchard.

Although this is not the full extent of the implications put forth by Mr. Alpert, some items are better responded to on a non-personal basis by A.P.P.L.E., upon which it would be inappropriate for me to comment as an individual.

It is regrettable that you have not seen the potential of positive actions. I am the first one to agree that the software industry itself is largely responsible for the current state of events. I for one would pay not one dollar for a protected diskette that had no way to back it up, nor would I buy a program that was so locked that I could not modify it for my own use.

We cannot now demand that the industry produce unprotected diskettes, but we can insist that they furnish a low cost, fast back-up support. I suspect that fully 50% of the software purchaser's dollar goes to the development of never-ending protection schemes. And who pays for this? The User. **The honest user is getting ripped off through the short-sightedness of the software industry and the minority of users who make a game out of giving their friends a copy of every piece of copyrighted software they can lay their hands on.**

Already the bit copiers are being defeated by more sophisticated protection schemes: programs are now being sold with hardware anti-copy devices. No doubt ways will be found to circumvent these, too, but let's call a truce. **As users, let's write good, unprotected software and sell it for a decent price. Let's get with the industry and beg them to provide backups. Let's not make it easier for anyone to steal software.**

Sincerely,
Val J. Golding

Val Golding is one of the leading figures in Apple Computing on the West Coast. He is presently the Editor of Call-A.P.P.L.E., a publication of the Apple Pugetsound Program Library Exchange based in Seattle, Washington. He is also one of A.P.P.L.E.'s directors as well as the former editor of the Apple Orchard, a publication of the International Apple Core.



LEGAL FORUM

by Barry D. Bayer

Consumers, Computers, & the Law

It's more than 9 months ABC (Anno Bit Copier) and the sky has not fallen in. Despite the dire predictions of the software production "doom-and-gloom" boys, the introduction of Back-It-Up and Locksmith did not result in the demise of the micro-software industry, nor even dampen the market in APPLEware [protected or unprotected variety]. Although anyone now can copy Visicalc if he wants to, the product continues to sell, like the excellent program it is.

Personal Software was apparently so little worried about Locksmith and the like, that the APPLE DOS 3.3 [16 sector] version is copiable using the old reliable techniques, rather than adding new gimmicks which would require a new version bit copier [which will soon be on the market, anyhow to counter Visi-Trend, Visi-Plot, and Visiwhoknowswhatelse].

Magazines which refused to carry bit copier advertising, either out of fear or supposed high ethical standards, don't have any problem with Central Point Software's COPY II PLUS, which sometimes is "not designed to copy 'protected' diskettes," and sometimes is, and sometimes can't make up its mind in the same magazine [See August, 1981 Creative Computing, pages 119 and 147].

And a new magazine, "hardcore computing" makes its debut shouting about the "copy protectors," explaining how to "unlock" a popular game program so that a fix and some enhancements can be made, and advertising Sensible [Back-It-Up] Software's "The New Protector" software protection program. [Will Dr. Emery come out with a "New Avenger" to counter the "New Protector?"]

In other words . . . it's business as usual in this crazy six-year-old industry.

But something new is happening. All of a sudden the big boys are moving in to the micro market. Take a look, and you see signs everywhere. BYTE Magazine is sold to McGraw Hill and is now rivalled only by the Sears Roebuck catalog for potential damage to my mailperson's back. Salesmen congregate around professional office selling "distributed logic" word procesors which turn out to be an only slightly souped up 64K Z-80, probably run-

ning under CP/M. [These things are huge, built like a tank, and probably work just as well as your . . . smaller but considerably cheaper - Z-80 CP/M system.] The "Incredibly Better Monster" company still hasn't arrived with its long heralded "micro," but the mighty Xerox Corporation beat IBM to the punch with a desktop word processor-computer that turns out to be a . . . 64K Z-80 machine running under CP/M. And at a not-unreasonable price.

In other words, the movers and shakers in the computer industry may be coming to the conclusion that small isn't bad, and in many instances may be better. It seems that the Z-80 CP/M configuration is becoming a really universal system, not just for the hobbyist, but for real business systems. [I'm just going to have to buy that Softcard for my APPLE, after all.] The 8-bit computer software industry is about to take a jump into mass-marketing which is going to make the last 6 years look like a tea party.

No, the sky didn't fall, and isn't likely to fall in the foreseeable future. To the contrary, opportunity for the small system software industry is clearly here. But are the software publishers equal to the challenge? I'm sure that businessmen will be willing to pay for quality software [in a mass-produced kind of way, of course] but will they stand still for the sort of sloppy products, and worse support, that is characteristic of much of the "hobbyist" industry?

These will be halcyon days for software vendors, with numbers of potential customers that will boggle the imagination. But businessmen won't put up with "protected" software which might go out at a crucial moment because it can't be backed up. Businessmen will not put up with software vendors who refuse to stand behind their products. Businessmen will not put up with the rather whimsical way some software vendors prefer to relate to their [hobbyist] customers. Example: [Maybe I'll inform my purchasers about known program bugs, and maybe I won't. Maybe I'll answer a letter . . . or a phone call . . . with a question, and maybe I won't.]

When a "Business Man" buys a product,

and it doesn't work, he doesn't just write a letter to the editor of a magazine, or complain at a Users Group meeting. He tells his lawyer to file suit. Take a look at headlines from a recent MISWeek . . . "Burroughs Admits to 160 Suits [Vendor Ends Silence, Says It Will Appeal 2 Verdicts]" . . . "119 Say They're Ready To Sue Data Gen." . . . "\$75,000 Payment Settles Mini Suit" . . . And that's just in the first four pages.

There's going to be lots of opportunities, but if software vendors are going to succeed in this new, "Micros-Are-Real-Computers-For-Real-Business-Uses-Now" world, they're going to have to get their acts together. And after getting used to doing things right for some of their customers, maybe the software people [who are left] will start doing things right for the rest of us.

Because a most obvious difference between business software and non-business software is that it doesn't really pay to enforce one's rights when amounts of money are small and when damages flowing from a breach of contract are calculable mainly in terms of lost enjoyment, the hobbyist must seek other avenues.

One thing that hobbyists can do, however, is to band together and exchange information on which product delivers on its claims, and which company just doesn't stand behind its product.

In this column we will attempt to sort our legal rights and wrongs, both in theory and in specific examples, that come to our attention from time to time and that have been sent in by our readers. If you feel particularly aggrieved by a company or product, write and let us know. Send copies of any promises made by the company [including advertising and claims in documentation] and an explanation of what went wrong, how you notified the dealer and manufacturer, and how they responded. The most interesting ones will be published; all of your examples will be read with interest.

We won't be able to change the computer world overnight. But maybe this column can do a small bit to make this economy safe for computer hobbyists.



REVIEW: of another bit copier

by Karen Fitzpatrick

In my first review (Premier issue) I should have made it clear that Sensible Software's BACK-IT UP is a single copy package... and I should have included their ratings as a single product and not two different ones.

I have not yet received the updated COPY II Plus, but I have received the new Locksmith and the new Back-It-Up... unfortunately, I haven't received the necessary data on their use of PARAMETERS which plays an important part in the effective use of both programs.

So....

Next Time

WHAT THE HECK IS A PARAMETER?

Your feedback will be greatly appreciated and very helpful. If you have a recent list of the Locksmith or the Back-It-Up parameters, please send it to me!

And, ofcourse, if you have a general bit-copy program and would like to have it reviewed, send a copy to me (the original only) along with documentation and a phone number that you can be reached at when I have questions.

Thanks,
Karen Fitzpatrick

NIBBLES AWAY

Here is a bit-copier that I missed in my first review. Since it was available at the same time as the other three (Locksmith 3.1, Back-It-Up and Copy II Plus), I will compare it to them rather than the newer versions. The name of this bit-copier is Nibbles Away. It is sold by Micro-Ware Distributing Inc. for \$59.

Nibbles Away boots on 13 or 16 sector Apple II's and supports single or dual drive copy with single controller. The user selectable options include:

1. Selectable start and end tracks
2. Automatic half-tracks
3. Synchronized copy
4. Raw data copy
5. Erase destination tracks
6. Reduced error checking

In addition to these options are the following features that set it apart from other bit copiers:

1. User-accessible nibbler
2. User defined parameter change
3. System filer (to save parameter changes)

The nibbler allows you to look at encoded bytes on the disk in order to determine what parameter changes to make. After changing the parameters, you may save them with the system filer to a data disk for future use.

Reliability

Reliability: With the parameter change function, this program was able to copy every disk I tested including a few that Locksmith 3.1 could not. The default value for the sync byte was \$FE which matched the changes on many protected disks that were copied with this program.)

Ease of use

Ease-of-use: The prompts are clear and easy to understand with default values used extensively. The ESC key will halt execution of the program and return you to the menu.

Amount of displayed information

Amount of Information given during disk copy: Nibbles Away gives the most information of any bit-copier I have tested. The screen display is dynamic and all information is clear and easy to understand. The program will inform you what it

is reading, analyzing, writing or verifying. It will tell you the type of errors encountered and keep a running total of each. The count is displayed next to the track number along with an inverse Y or N which tells if the track was copied or not.

Documentation

Documentation: The documentation is excellent. The operation of a bit-copier is explained and each function of the program is explored with examples of how to use them. You are shown how to use the nibbler and what to look for on a protected disk. Then you are shown how to use this information to change the parameters in the program. Finally, you are shown how to store these parameter changes on a data disk for use at a later date.

Speed

Speed of Copy: The program takes about 4 minutes to copy a disk, longer if it encounters many errors. It seems that none of the reliable bit copiers are very fast. Personally, I am more interested in a good copy than in blinding speed.

a note...

I was also going to review Locksmith 4.0 and Back-It-Up II. These are the much talked about NEW bit-copiers. They are definitely better than the old versions but they are also user-intensive. By user-intensive, I mean that they require more input from the user in the way of parameter changes. Unfortunately, these parameters are not well documented and in some cases are not even mentioned.

Rather than review these new copiers, I am working on obtaining the information needed to understand the parameter changes and what to look for on a protected disk in order to select the correct parameter to change. The article will be ready soon and will contain the following:

- A. Parameter numbers and values
- B. What each parameter does or defines)
- C. How to use each parameter

The article will also review Locksmith 4.0 and Back-It-Up II in a side by side comparison.

PLAIN TALK ABOUT "COPY PROTECTION"

A lot has been said and written about copy protection and software piracy since Omega made Locksmith available to Apple II users earlier this year. We have been accused of encouraging illegal copying of copyrighted software. Software publishers have threatened to boycott magazines which carry our advertising, and the pros and cons of Locksmith and copy protection devices have been debated in Apple forums throughout the country. But, we at Omega haven't really told you, the Apple user, our side of the story.

Locksmith was originally developed as an intellectual exercise by an Apple user over a year ago. And we suspect that sufficient information about the Apple DOS and the way information is stored on a disk has been long available to the general public, so that ANYONE who was REALLY interested, and who wished to spend a LOT of time, could have written a program that does many of the things that Locksmith does. Similarly, there is really no "secret" to writing data base programs, adventure programs, or even spread sheet programs. The literature is there if you want to look for it. But it takes a lot of hard work to develop any software package that works in all cases, that is crashproof, that interfaces easily with a non-experienced computer user, and that is well documented. A LOT of hard work.

But even before Locksmith was available to us, we, as Apple users, recognized a definite problem with the software we were buying and using. Much of it worked well. But it was very aggravating to not be able to make a backup copy of certain "copy protected" programs. Most software publishers didn't supply backups of their programs, and those that had any policy required signing oppressive agreements or paying questionably high yearly fees for presumed, but not guaranteed, updates. Among those who did not offer back-up was one who 'sold' us a new copy (when we returned our crashed disk). Although they advertised the importance of having their program running every day, they made us wait up to 6 weeks to get the replacement. Most vendors just ignored the problem. We, as consumers, were simply taken advantage of. In many cases we relied so much on a particular program, that it became very costly to have to wait weeks or more to replace a blown disk. Software publishers were just not responsive to the users problems caused by "copy protection".

When we first became aware of Locksmith, we investigated the state of the law, and discovered that no one knew whether the owner of a program could copy it for backup. And for quite a while we debated whether we should market Locksmith.

On December 12, 1980, a change was made to the Copyright Act which resolved these questions. It is now the law of the United States that the existence of a copyright notice on a computer program does NOT make it illegal for the legitimate owner of that program to copy it for archival purposes. Backups are now clearly legal. (Of course, when you sell your purchased program, you must destroy the backups you have made). Only after such use clearly became legal did we decide to sell Locksmith.

Now with the new copyright law, which for the first time gave software publishers clear rights that were enforceable in court, but which also gave "backup" rights to software purchasers, and with the demonstration that Locksmith could and would provide back-up for the user, we assumed that software publishers would drop their copy protection schemes and educate the public as to their rights and responsibilities. Even the use of hardware protection that gives copy-ability to the software would be acceptable. Unfortunately, their response has been to pressure magazine publishers into refusing our advertising, and to invent new copy protection schemes.

Well, the word about Locksmith was impossible to stop. We couldn't advertise, but we have sold a gratifyingly large number of programs. As to new copy protection schemes, the new Locksmith (version 4.0) will adjust to them, and copy virtually anything protected that way. But please. For us, for yourselves, and for the entire industry, use Locksmith only for its intended legal purposes.

The new version is more than just the best copy program available. There are also four additional utilities included. A disk speed program, a degausser, a nibble editor and a media surface analyzer are included. And we stand behind our products. Our customer service department is available (and anxious) to help with problems.

Locksmith 4.0 is available from us, or your local dealer. Visa and Mastercard users call Toll Free 1-800-835-2246. Kansas residents call 1-800-362-2421 or send \$99.95. (Registered owners of prior versions can obtain an update for only \$20. If you haven't received a letter from us, please call.)

Another Quality Product from Omega MicroWare, Inc.

(formerly Omega Software
Products, Inc.)
222 So. Riverside Plaza
Chicago, IL 60606
Phone 312-648-1944



more on....

USING BOTH SIDES OF YOUR DISKETTES

by
karen
fitzpatrick

DOUBLE your DISKETTES

The only reasons your Apple][cannot use the back side of your diskette are:

1. There is no notch.
2. The diskette manufacturer did not test the back side, or worse, put the flawed front to the back.

A nibbling tool will solve problem number 1.

DISK PREP will solve problem number 2.

DISK PREP formats and tests your disk. Sectors with flaws are left so that they cannot be used. Your disk is left ready to boot, complete with a flaw report program saved on it.

\$25.00

sympathetic software
9531 Telhan Drive
Huntington Beach, CA 92646

California residents add
\$1.50 sales tax
Dealer inquiries invited

a review of 2 programs that will CHECK FOR FLAWS

At some point in time, every computerist has considered the possibility of using the back side of their disks. The lack of a write protect notch is a hinderance that can be solved with a common paper hole punch. Some enterprising computerists have even defeated the write protect switch in the disk drive. The real problem is how to determine whether the flip side has any

flaws. There are two programs that will scan a disk for flaws and change the disk map so that these areas are not used by DOS:

1. **Disk Prep**, by Sympathetic Software (\$25).
2. **Disk Recovery ("the Scanner")**, by Sensible Software, Inc. (\$30).

The SCANNER

DISK RECOVERY "THE SCANNER" can be used on 13 or 16 sector systems with 48K memory, and can be configured for a language card.

Documentation

The documentation for The Scanner is clear and easy to understand. It explains the different functions that are possible with the program. There is a section of "Special Notes" that gives a brief explanation of some of the properties of The Scanner program.

Screen Lay-out

The Scanner has an excellent screen lay-out that includes a continuous readout of the sectors being scanned. There is a "report" of the bad sectors, files deleted, number of errors found, number of mismarked sectors, and number of free sectors. When The Scanner is scanning and comes upon a bad sector, the readout stops and the track and sector is printed in inverse on the screen. Then the scan continues. The Scanner uses a default entry system.

16 Sector

Before using The Scanner, you must first configure it for your DOS or language card. This takes about 5 seconds and is a one-time requirement unless you change the DOS. The program will only scan a previously INITIALIZED disk. It takes about 20 seconds to do a complete

scan (a little longer if it discovers any bad sectors) and 35 more seconds if you INITIALize it first. After scanning a disk the program will ask you if you want to rebuild the VTOC. If there is a bad sector in one of your files, you will be given the option of deleting the file.

13 Sector

Before using The Scanner on a 13 sector system, you must first configure it for 13 sector DOS. When you run The Scanner on a 13 sector disk it will automatically rebuild the VTOC. This is due to the way 13 sector DOS INITIALizes a disk. You are informed to ignore errors on track 0, sector A and track 2, sectors 9 thru C. These errors occur either on a newly INITIALized disk or if Update 16 has been run on the disk. When finished with the scan, the program will ask you if you want to rebuild VTOC. It takes The Scanner 1 minute to rebuild the VTOC and scan a disk (INITIALIZATION takes about 2 minutes).

Extras

The Scanner does not INITIALize a disk when it scans for flaws. This means that it can be used on disks that already have files. In addition to scanning a disk for flaws, the program will rebuild the VTOC using the directory and display a map of the used and unused sectors. When rebuilding the VTOC, the Scanner will display the file names (control characters in inverse) and the sectors used by that file.

DISK PREP

DISK PREP runs on both 13 and 16 sector systems with 32k or 48k memory. (The disk has a 16 sector program on the front and 13 sector on the back.) The program will INITIALize a disk, scan for flaws and open a "Bad Sector File".

Documentation

Documentation for Disk Prep is included on the disk. It is very clear and concise. The instructions are easy to follow and there are explanations of the possible drive and I/O errors that may occur. There is also a good argument for and against putting DOS on your disk, and directions for prepping the back of your disks. The documentation is the same for both the 13 and 16 sector sides.

Screen Lay-out

The format for Disk Prep is good. It prompts you to insert your disk and select a volume number (1 thru 255) and uses a default entry system. You will be asked if you wish to have DOS put on the disk. If you select this op-

tion, you will then be asked to enter the name of your greeting program. When DOS is put on your disk, the bad sector report will be SAVED to the disk using your greeting program name. If you choose not to put DOS on your disk, the program will automatically start a file called "Bad Sector Report".

16 Sector

The program takes about 3 minutes and 40 seconds to INITIALize and scan a 16 sector disk. Adding DOS takes an additional 20 seconds.

13 Sector

It takes 3 minutes to INITIALize and scan a 13 sector disk and 18 seconds to add DOS.

Extras

Disk Prep INITIALizes as it scans, saving the user one extra step.



SENSIBLE SOFTWARE, INC.

The Utility People

Making Life With the Apple II A Bit Easier

DISK RECOVERY "THE SCANNER"

**48K+, Disk II
Apple II/Apple II+
\$30.00**

Are you plagued with disk I/O errors?
Does the integrity and safety of your
data concern you?
This program is a must for all
Disk II owners!!

Just as "Apple Pascal" provides a "BAD BLOCK SCAN", DISK RECOVERY will do a complete scan of your diskettes' recording surfaces. Damaged areas will be "marked" as used in the disk directory so that no attempts will be made to "WRITE" to a bad sector. The VTOC is completely rebuilt to accurately reflect BAD, USED, and FREE SECTORS, further insuring the integrity of the diskette. A complete report is generated advising the user of all corrections BEFORE the object disk is altered. A resulting "DISK MAP" is presented for your review. Flexibility of the program allows evaluation of NEWLY INITIALIZED DISKETTES as well as DISKETTES THAT ALREADY CONTAIN PROGRAMS AND/OR DATA! The SPEED of analysis is unsurpassed - only 18 seconds for a 16 sector diskette! DISK RECOVERY is supplied in the revolutionary QUICKLOADER format which allows easy reconfiguring to make it fully compatible with 13 sector and 16 sector diskettes as well as your SUPER-TEXT data diskettes! TREAT YOURSELF TO PEACE OF MIND KNOWING THAT YOUR PROGRAMS AND DATA ARE SECURE WITH...DISK RECOVERY.

BACK IT UP II has been improved and enhanced to provide the end user with the most powerful duplication system for the Apple II computer. Numerous PARAMETER CHANGES are allowed to directly handle the multitude of possible security formats as they are encountered. Direct study of these formats is possible with our own NIBBLE READER. Both 'full tracks' and 'half tracks' may be analyzed individually. Preview the source disk with the QUICK SCAN option to readily locate pertinent data. When duplication is begun AUTOMATIC HALF-TRACKING is readily enabled. A powerful SYNCHRONIZATION MODE has been added to the BACK IT UP II system. The MEND TRACK function manipulates data from designated tracks that are otherwise irretrievable with other programs. The end user is kept WELL INFORMED of the progress and any problems encountered once duplication has commenced. As with all SENSIBLE SOFTWARE products BACK IT UP II is extremely easy to use - USER FRIENDLINESS is more evident than ever before. The documentation has been expanded to a TUTORIAL ON DISKETTE FORMATTING as well as an easy to read USER GUIDE.

The availability of this program in no way implies that SENSIBLE SOFTWARE, INC. supports software piracy and is intended for the sole use of the computerist who wishes to 'back-up' his/her diskette library. Although itself uncopyable by itself, SENSIBLE SOFTWARE, INC. does have a very liberal \$5.00 disk update/replacement policy.

BACK IT UP II

**48K+, Disk II
Apple II/Apple II+**

\$60.00

**WE LISTED TO THE APPLE USERS!
WE LISTENED TO THE REVIEWERS!**

NEW! IMPROVED!

NOW EVEN BETTER THAN BEFORE!



SENSIBLE SOFTWARE, INC.

6619 Perham Drive Dept H
West Bloomfield, Michigan 48033
313-399-8877

Visa and Mastercard Welcome

Please add \$1.25 postage and handling per disk

adventure tips

by mike flynn

How now? What's news with you?

I have received several requests for clues and maps, but have received only a few reader-contributed clues. Many thanks to those who sent me maps and notes.

At this time I wish to re-emphasize the basis upon which this column is built: USER SUPPORT. Without it, this column will cease to exist and I will be back on the bread line. So, to make a short story long, get those clues in the mail!! If you have only made a small degree of progress in a game, send in what you know so far. Remember: EVERY CLUE HELPS.

Now, for some adventure tips!

This quarter I will give some tips on OO-TOPOS and ULTIMA:

OO-TOPOS:

Once you have gotten out of your cell [the "North" command should help out here], you should come upon an abandoned guard cell. Get the food and proceed East. This way you will have a laser at your disposal. Bear in mind that this adventure is like no other, so most of your regular tactics will not work.

ULTIMA:

I have had the best luck in the old days by being a thief. In fact, I managed to steal a phaser in mideaval times! The only other real valid clue is: Get as much information out of the bartenders as you can. If buying a drink doesn't loosen his tongue, keep buying!!

This issue I will give a short review of a series of adventures known as "EAMON".

EAMON is a very unique series of adventure games in that you start out with what is known as the "EAMON MASTER" diskette. With this diskette, you can build your character, buy any needed weapons, armor, etc. After having accomplished this, you can go on your first adventure!

In order to go on an adventure, you must have a separate game disk with one of the many EAMON adventures on it. However, on the master game disk is the simplest of all the EAMON games. This is the "BEGINNER'S CAVE". It is recommended that you try this game out first in order to build your character's strengths so as to be better able to negotiate some of the harder games.

One of the nicest things about EAMON is the fact that the disks are not protected so you can make any modifications yourself. The author states in each of his game disks that he encourages non-commercial distribution. In other words, you can trade or give away EAMON disks to your heart's content.

I would like to point out that there are about ten EAMON adventures currently available. One of these disks is known as the EAMON designer disk. This disk allows the user to design his own adventure scenerio. The format of this disk, and the manner in which it allows you to design your game, is excellent.

For more information, write to:

**Donald Brown
407 Peery Parkway
Golden, Co. 80401**

So, until next time, when we ask the question "Who is ZORK?"...keep adventuring!

**Mike Flynn
c/o HARDCORE**

Although we sometimes answer questions asked in letters directed to staff members, we usually don't have the time. However, we will answer those important questions in this column.

Because many of our readers have asked the same questions (in different words), we have gathered them together here, with their answers, without crediting their authors.

some problems with demuffin & surgery...

Q Are there any programs Demuffin will work on?

A Demuffin will only work if the RWTS in the DOS used by the disk has not been extensively modified.

Q Several programs in V.1, No.1 require Applesoft and an Integer Card. Will any other language cards work as a substitute for the Integer Card?

A In order to use Demuffin Plus you must be able to get into the monitor when you press RESET. The old monitor ROM on the Integer Card allows this. The new Auto-start ROM does not. A RAM Card which has a switch to select the RAM or on-board ROMs will sometimes work. Load Integer first, write-protect the card, then BOOT the protected disk. Use the switch to select the RAM Card. When you press RESET you should be in the monitor. If not, then your card won't work.

Q How can I use the open-heart surgery method to get Binary and Text files off a disk?

A First see the article on using Muffin13. If this doesn't work then follow steps 1 thru 9 of open-heart surgery in order to reconnect the DOS. BLOAD the Binary file, then press RESET to disconnect DOS. Look at location \$AA72, \$AA73 for the start of the file and location \$AAG0, \$AA61 for the length. (Hex addresses are standard L0-Byte, HI-Byte format) SAVE the file to tape or BOOT a slave disk and BSAVE. BOOTing a slave disk will only overwrite page 3 (\$300-\$3FF) and page 8 (\$800-\$8FF). If the Binary file resides in these areas, use the monitor move command to move it to a safe location. Move the binary code back before you BSAVE it.

For Text files, you must write a short program that OPENS the file then GETs the information a byte at a time. I usually send the information to a printer as I GET it, but you could also store each byte in memory. BOOT a slave disk and recover the

information. (Remember not to use page 3 or 8.)

Q I have an Apple II+ 48K-single disk with Integer Card and 3.3 DOS, and have been unable to get Demuffin Plus to work.

A A number of people are having problems with Demuffin Plus. Before you start, insure the following:

- 1) That you have the Programmers Aid ROM. (Either resident in the Integer Firmware Card or loaded into your language card.
- 2) That you are in Integer language. (Type INT and press RETURN) The Integer ROMs must be enabled before you call the Programmers Aid ROM. Calling the Programmers Aid ROM while in Applesoft will only result in calling the Applesoft Interpreter with unpredictable results.

If you have the Programmers Aid ROM and are in Integer and the steps still do not work, then stop, go to your dealer and get a more recent copy of Muffin. We have found that some of the early copies of Muffin will not work with the Demuffin Plus procedure.

Q How can you defeat a reset vector change?

A With the Autostart ROM in control it is not possible. However, the Integer F-8 ROM does not use the reset vector and is the only way I know of defeating the vector change.

Q I have a large number of disks with the volume #254. How can I change these volume numbers to be in consecutive order?

A A lot of us have disks with a volume number of 254. This is the default value used by DOS whenever you INITIALize a disk and do not specify a volume number. The disk Volume number is part of every sector address mark. The only way to change this is to INITIALize a new disk with the Volume number you wish, and then transfer all files to that disk.



Customizable Modu

In part 1 (published in Update 1.1), I introduced my version of the Customizable Modular combination Adventure-Arcade Game, a disk-based game in modular format so that the user could change, add or remove modules and thus create a unique version. In this installment I will reveal the first maze module used to create the topography through which the adventurer must travel.

It is called the **Mountainous Island Maze-maker** which is graphically presented in a novel way: as a relief or 3-D map. Other maze modules will be presented later as this series continues. Parts 3 and 4 will be published in Updates 2.1 and 2.2, and part 5 will appear in issue No. 3. I welcome your responses to my efforts.

part 2 INTRODUCING THE "MAZE" CONCEPTS

All adventures in role-playing computer games take place in what I call "the MAZE". This all-inclusive term is also used here to describe the unknown quality of places you must travel through as the adventure progresses. The mysterious elements of the "maze" can be revealed to the player in a number of ways, the most common being regular TEXT.

But Apple-adventuring has come a long way since the all-TEXT versions. Many now display the "maze" in a number of ways. Some have flat but colorful maps of islands, forests, rooms, caverns. Some even create the illusion of 3-D depth by PLOTting the rooms in perspective. Some have hi-res pictures that are loaded from disk at opportune moments.

Maze Detail & Complexity

Mazes can be simple to elaborate. The more complex it is, the more details must be stored that can be called up to describe it for the intrepid voyager. Data for the maze can be stored in the program itself as DATA statements to be READ when needed; but this method severely limits the detail that a good adventure must have. Data for the maze can be stored in memory as variables (usually string arrays) or it can be coded and stored in memory as actual bytes. One can also store maze data on disks to be loaded when needed. I prefer a combination of all these methods.

What is maze detail? Simple adventures just tell you that to the West is a wall. More maze detail would include: a crack in the wall, or a blood stain near the top. Even more detail would let the program determine whether the adventurer could destroy the wall, or dig a hole through it, or take a stone from it, or hide some coins in it, and so on. The greater the detail, the more complex and realistic the role-playing becomes.

Maze Scale & Quantity

Maze detail and maze scale are inversely related because of data storage restrictions (whether on disk or in the computer). The larger the scale and complexity of the maze, the more storage it will require and the less detail can be stored about each element of the maze. Of course, in many instances great detail is not necessary (as in moving from one rendezvous to another across a map) but there are moments when enormous detail must be immediately available to maintain the adventure's fantasy world-view.

I have broken down the scale/detail spectrum into three general categories which will become useful later in this article:

1. ROOM BY ROOM

This category requires a great amount of detail for it is at this small scale that battles should be fought, and essential objects gained and lost. This is the greatest detail and the smallest scale.

2. SPECIFIC TOPOGRAPHY

An example is movement through a city or forest, where the detail is less important (but not irrelevant) and movement is stressed. The adventurer is in transit, looking for that Room-by-room action. This is medial in detail and scale.

3. GENERAL TOPOGRAPHY

Here, detail can be almost missing since the emphasis is on motion from one "event" to another, from town to town, from cave to castle, etc. Here, detail is minimal, and the scale is very large. Enormous maps, spanning entire disks, would fall into this category.

The Concept of "a Room"

Let me explain what I mean by a "room". Motion through a maze of any scale is usually by discrete units that I call "rooms". It is the detail of these rooms that I am referring to when I speak of maze detail.

Maze data is stored differently for these three general levels of detail/scale.

In General Topography, the scale can be enormous because usually only one byte of storage is used to describe each "room". 256 different "rooms" are thus available, and these rooms are usually graphically displayed by using a character-generator link, substituting a graphic set in place of the letter or character and PRINTing it on the Hi-res Page. Large maps can thus be displayed and then scrolled up or down, left or right. One can also code the byte differently and display a different kind of map. It is this route that I have taken in my Relief Mapping technique.

In Specific Topography, the scale is medium because more than one byte is used to describe each room. This allows more detail as a greater number of bytes are used. Generally, though, normal strings and data statements are used. Most adventure games fall into this category. Usually very little disk access is necessary except to display some hi-res graphics.

In Room-by-room Topography, only data for just a few rooms can be stored in memory at any one time and the game must access the disk often as the player moves from room to room. The greatest amount of detail can be revealed when using this method.

I think that a good adventure should range throughout this spectrum of scale and detail, and our modular adventures will do so by having a module for each of the three detail-scale levels.

Relief (3-D) Mapping

Drawing a map in simulated 3-D is difficult at best, and a frustrating failure at worst.

The Island Maze-maker is a compromise I worked out between the need for visual clarity and the scale of each "room". As you can see from the illustration, the islands appear to be three dimensional because of the way we have learned to translate occlusion and elevation cues.

The actual grid is 20 by 20 points, with all 400 points (rooms) stored as bytes on TEXT page 2. The program itself loads above Hi-res Page 1 and PLOTs on this page.

It is not a long program, and it is not a polished product. I present it here because it is a stepping-off point to a number of different maze possibilities. You need not limit yourself to an island. You can have a much larger number of rooms, but you'll have to show only segments of the map at any one time or the screen will blur the elevations points together, making the map unreadable. Besides the mountains, you can have lakes, rivers, volcanoes.

Each byte can have a value of 0 through 255. I have coded this byte in two ways:

1. The X and Y map coordinates are stored in the byte's actual position in memory (lines 300 and 310).

Since the map is 20 by 20, one can derive the X and Y positions by dividing by 20 (X%) and subtracting it from its position in memory (Y%). The actual screen position is calculated in lines 370, 380.

2. The elevation and site data is stored in the byte value itself (lines 350, 360).

Elevation values can be from 0 to 12. It is stored in the byte as the "tens" and "hundreds" place marker (place 10).

Site values are from 0 to 9 (for elevations above 9, only site values of 0 to 5 are valid) and are stored as the "ones" place marker.

The mountains are created by:

1. Selecting a position in memory (ZTIP%) that is on the map (line 1110) and

2. Giving it an elevation from 8 to 12 (line 1120).

3. Giving each of its sides its own slope from 1 to 3 (line 1000).

4. Going in concentric squares clockwise around this pinnacle (lines 1200-1410) and

lar Adventure-arcade Game

PART
2

by
b.
bryte

program: MAZE-MAKER

5. Examining the elevation on its right (ZMEM%: the elevation closer to the peak) and

6. Choosing a lower elevation depending on a random factor and the slope for that side of the mountain (lines 200-240).

7. If it is higher than the actual elevation at that point then it will change that elevation (line 1450).

8. Plot the point and draw lines to the elevations in all 4 directions except if that point is not on the map. (lines 2000-2090).

9. Do this for a few mountains (line 1100) and

10. Clear the screen and redraw the entire island (lines 2000-20090)

I've written this version so that it PLOTS the high peaks (higher than 9) in white to simulate snow-capped tops.

This technique has tremendous possibilities. By storing the actual screen X and Y positions elsewhere in memory, one can make it PLOT a saved (previously plotted island) isle rather quickly so that one can move from isle to isle.

By calculating and saving the island X, Y coordinates elsewhere, one can make the island turn around in 3-D style as you fly around it, or over it.

The possibilities are exciting and I hope that with this rough Maze-Module you can create unique graphic presentations for later incorporation into your own adventure games (whether it is part of this modular series or simply your own independent game).

next time: Site Characteristics

Next time I will have some extra lines that will let you move through the map and edit it as well. I will also explain the concept of "site". The next addition to this module will also PLOT towns, castles, cave entrances and reveal whether the voyager is standing on water, swamp, grass, bare rock or earth, or is beneath poplars or pine or maples, etc.

— read this —

Because this program must be loaded above Hi-Res page 1 (to make room for the maze data stored on Text page 2 and to permit plotting on Hi-Res page 1), you must POKE 104,64 and POKE 16384,0.

OR you could RUN this Program Loader:

```
1 D$=CHR$(4): POKE 104,64: POKE 16384,0
2 PRINT D$;"RUN MAPMAKER 1"
```

Be sure to SAVE the mazemaker as MAPMAKER 1.

Island and Mountains maker

50 GOTO 10000

=====

MEMORY CHECK #1

```
100 IF TMEM% < BMEM% OR TMEM% > EMEM%
    THEN POP : POP : GOTO 1500:
    REM <NOT ON MAP>
110 RETURN : REM < OK, ON MAP >
```

=====

T MEMORY CHECK

```
130 IF TMEM% < BMEM% OR TMEM% > EMEM%
    THEN POP : GOTO 2080
140 RETURN
```

=====

MAKE ELEVATION

```
200 ELEV% = INT ( PEEK (ZMEM%) / 10)
    - 10 - INT ( RND (1) * (SLOPE%
    (SIDE) + 1)) - 2
210 IF ELEV% < 0 THEN ELEV% = 0
240 RETURN
```

=====

N-E-S-W SIDES

```
250 TMEM% = PMEM% - 20 : RETURN : REM
    <NORTH>
260 TMEM% = PMEM% + 1 : RETURN : REM
    <EAST>
270 TMEM% = PMEM% + 20 : RETURN : REM
    <SOUTH>
280 TMEM% = PMEM% - 1 : RETURN : REM
    <WEST>
```

=====

T MEM CALCULATION OF X, Y, ELEV, SITE

```

300  YYZ = (TMEMZ - BMEMZ) / 20
310  XXZ = TMEMZ - BMEMZ - YYZ * 20
320  IF PEEK (TMEMZ) < 100 THEN RETURN
340  FACTZ = PEEK (TMEMZ) / 10
350  ELEVZ = FACTZ - 10
355  IF ELEVZ > 8 THEN SNOW = 1
360  SITEZ = PEEK (TMEMZ) - FACTZ * 10
370  XMAPZ = XXZ * 10 + YYZ * 3
380  YMAPZ = YYZ * 5 - ELEVZ + 13
390  RETURN

```

=====

NORTH SLOPE

```

500  TMEMZ = PMEMZ + 1
510  GOSUB 100: REM <MEM CHECK>
520  ZMEMZ = TMEMZ + 20
525  IF WIDTH = LIMIT * 2 THEN ZMEMZ =
      ZMEMZ - 1
530  GOSUB 200: REM <ELEV>
550  PMEMZ = TMEMZ
590  RETURN

```

=====

EAST SLOPE

```

600  TMEMZ = PMEMZ + 20
610  GOSUB 100
620  ZMEMZ = TMEMZ - 1
625  IF WIDTH = LIMIT * 2 THEN ZMEMZ =
      ZMEMZ - 20
630  GOSUB 200
650  PMEMZ = TMEMZ
690  RETURN

```

=====

SOUTH SLOPE

```

700  TMEMZ = PMEMZ - 1
710  GOSUB 100
720  ZMEMZ = TMEMZ - 20
725  IF WIDTH = LIMIT * 2 THEN ZMEMZ =
      ZMEMZ + 1
730  GOSUB 200
750  PMEMZ = TMEMZ
790  RETURN

```

=====

WEST SLOPE

```

800  TMEMZ = PMEMZ - 20
810  GOSUB 100
820  ZMEMZ = TMEMZ + 1
825  IF WIDTH = LIMIT * 2 THEN ZMEMZ =
      ZMEMZ + 20

```

```

830  GOSUB 200
850  PMEMZ = TMEMZ
890  RETURN

```

=====

PICK A DIFFERENT SLOPE FOR EACH OF
THE 4 SIDES OF EACH MOUNTAINS

```

1000 FOR SIDES = 1 TO 4 : SLOPEZ
      (SIDES) = RND (1) * 4 + 1 :
      NEXT SIDES

```

=====

MAKE MOUNTAINS

STEP 1 SET PEAK HT.

.....

```

1100 RZ = RND (1) * 3 + 3 : FOR
      MOUNTAIN = 1 TO RZ
1110 ZTIPZ = RND (1) * 400 + BMEMZ - 1
1120 ELEVZ = RND (1) * 5 + 8
1130 IF PEEK (ZTIPZ) < 110 OR PEEK
      (ZTIPZ) > 180 THEN 1110
1140 ELEVZ = RND (1) * 5 + 8
1150 POKE ZTIPZ, (ELEVZ + 10) * 10

```

STEP 2 CALCULATE SIDES

.....

```

1200 FOR LIMIT = 1 TO 5
1210 PMEMZ = ZTIPZ - 20 * LIMIT - LIMIT
1300 FOR SIDES = 1 TO 4
1400 FOR WIDTH = 1 TO LIMIT * 2
1410 ON SIDES GOSUB 500, 600, 700, 800
      : REM <N,E,S,W SLOPES>
1430 ZELEVZ = (ELEVZ + 10) * 10
1440 IF PEEK (PMEMZ) < 110 THEN 1500
1450 IF PEEK (PMEMZ) < ZELEVZ THEN
      POKE PMEMZ, ZELEVZ
1490 GOSUB 2000
1500 NEXT WIDTH
1510 NEXT SIDE
1520 NEXT LIMIT
1530 NEXT MOUNTAIN
1540 GOSUB 20000
1590 END

```

=====

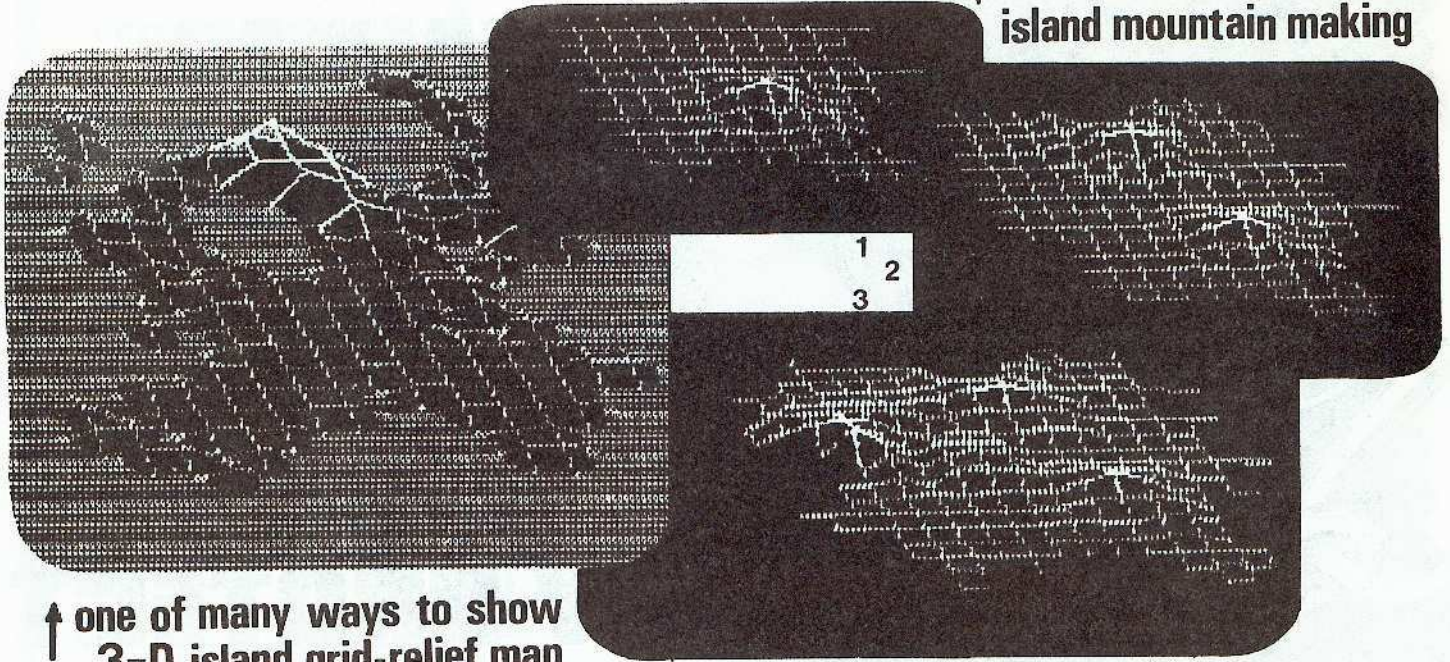
PLOT, GRAPH & DRAW LINES
TO NEW ELEVATION

```

2000 HCOLOR= 3
2005 TMEMZ = PMEMZ : GOSUB 300
2010 HPLLOT XMAPZ, YMAPZ : XZ = XMAPZ
      : YZ = YMAPZ

```


↓ 3 of the several steps in island mountain making



↑ one of many ways to show 3-D island grid-relief map

```

2020 FOR TEST = 1 TO 4 : ON TEST GOSUB
    250, 260, 270, 280 :
    REM <N,E,S,W SLOPES>
2030 GOSUB 130 : GOSUB 300
2040 IF ELEV% > 8 THEN SNOW = 1
2045 IF TMEV% < 100 THEN 2080
2050 HCOLOR= 2 : IF SNOW > 0 THEN
    HCOLOR= 3 : SNOW = 0
2055 IF ABS ( X% - XMAP% ) > 10 THEN
    2080
2060 H PLOT X%, Y% TO XMAP%, YMAP%
2080 NEXT TEST
2090 RETURN

```

=====

INIT VARIABLES

=====

```

10000 BMEV% = 2100 : EMEV% = 2499
10005 G$ = CHR$ (7) : D$ = CHR$ (4)
10010 GR : POKE - 16299, 0
10011 MAX = 4

```

=====

CLEAR TEXT PAGE 2

TO VALUE OF 110 (ELEV OF 1)

=====

```

10020 FOR A = BMEV% TO EMEV% : POKE A,
    110 : NEXT A

```

=====

CREATE IRREGULAR ISLAND PERIMETER

=====

```

10030 AA = BMEV% : BB = BMEV% + 19 :
    TEST = MAX : GAP = 20 : FOR IN =
    AA TO BB : GOSUB 11000 : NEXT IN
    : REM < TOP EDGE>

```

```

10040 AA = BB : BB = BMEV% + 399 :
    TEST = MAX : GAP = - 1 : FOR IN
    = AA TO BB STEP 20 : GOSUB 11000
    : NEXT IN : REM <RIGHT EDGE>
10050 AA = BB : BB = BMEV% + 380 :
    TEST = MAX : GAP = - 20 : FOR IN
    = AA TO BB STEP - 1 : GOSUB
    11000 : NEXT IN : REM <BOTTOM>
10060 AA = BB : BB = BMEV% : GAP = 1
    : TEST = MAX : FOR IN = AA TO BB
    STEP - 20 : GOSUB 11000 : NEXT
    IN : REM <LEFT EDGE>
10080 HGR
10090 GOTO 1000

```

=====

CLEARS MEMORY AROUND ISLAND

=====

```

11000 R% = RND (1) * MAX + 1 : IF ABS
    ( R% - TEST ) > 2 THEN 11000
11020 TEST = R% : POKE IN + (GAP *
    TEST ), 100 : FOR JJ = IN +
    ( GAP * TEST ) - GAP TO IN STEP
    - GAP : POKE JJ, 0 : NEXT JJ
11090 RETURN

```

=====

DRAW ENTIRE MAP

=====

```

20000 HGR : HOME
20010 FOR I = BMEV% TO EMEV%
20030 PMEV% = I : GOSUB 2000
20090 NEXT I : RETURN

```



EXPANDED DISKLOCKS

by bobby

How To COPY-PROTECT your disks

In the most popular column in the Premier Issue, Our mysterious Bobby will attempt to cover the entire topic of Copy-'protection' techniques in upcoming installments.

Published in slightly different form in Update 1.1, this expanded DiskLocks now includes 3.2 information as well. The tables have been updated. Information you garner from this article and the SOFTKEY II should be used along with DiskEdit and DiskView, Chuck Haight's pet programs. Future articles will assume that you already have these soft-tools at your disposal.

Copy protection can be loosely divided into 4 methods.

1. copy-protected disks
2. software protection
3. hardware protection
4. memory protection

I say "loosely" because each area overlaps the others. Method 1 usually involves changing the format of information on a disk. The disk is not copyable by normal means. In method 2 the program goes back to the disk and searches for a particular pattern. The disk is seemingly copyable but bombs after the program is run. And in method 3 a card or other hardware is used by the program. The disk is copyable but requires the hardware to run. Method 4 is really a subgroup and is employed by the other three protection schemes. It includes such things as: locking-out the reset and control "C".

The number of ways that a program can be protected has yet to be counted. Some companies even use other computers to devise disk formats that cannot be duplicated by the Apple.

In this article we will look at method one: Protecting your program by altering the disk format. In order to understand these changed formats you need to first look at a normal disk.

The flexible (or floppy) diskette can be thought of as a disc-shaped piece of recording tape, and essentially that's all it is. The reason for the flat disc shape as opposed to using a flat strip (as in a tape) is for speed of information retrieval. For instance, if you were to have a program stored in the middle of a tape, the computer would have to READ in all of the tape preceding the area where your program resides. This method of information retrieval is known as "sequential access". The disk, on the other hand, is set up in such a way that the computer can go directly to a piece of data or program by scanning the disk laterally. This method of information retrieval is known as "random access".

When a disk is INITIALIZED the Disk Operating System (DOS) divides it into 35 concentric tracks (see figure 1). DOS 3.3 then divides each track into 16 blocks called "sectors". DOS 3.2 creates 13 "sectors". Each sector contains an address mark and a data mark. These marks are a unique set of bytes that are used by DOS like street signs.

The address mark tells the DOS what track/sector it is currently READING. It contains the volume, track, sector and checksum information. The data mark surrounds the actual

data. It tells the DOS where the data begins and ends and contains a checksum that is used to verify the accuracy of the data.

If you have ever tried to LOAD a program, and the disk drive started making a slight chatter, chances are that the DOS could not READ one of these markers. It then recalibrates the READ/WRITE head by moving it back to track zero and stepping (counting each track that it passes over) back out to where it was supposed to be.

The tracks are numbered from \$00 (0) to \$22 (34) and the sectors from \$00 (0) to \$0F (15). Track \$00 thru track \$02 (a total of three tracks; zero, one and two) contain the DOS program.

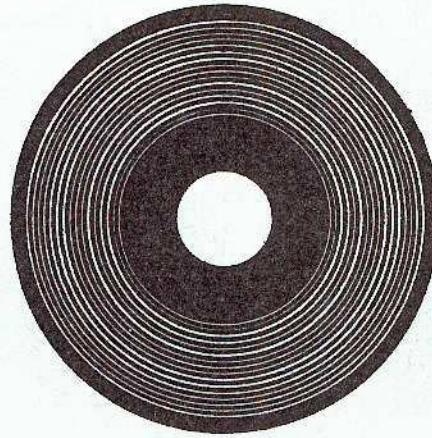
The DOS gives the Apple the ability to manipulate data on a diskette. In this program are all of the commands related to controlling the disk drive (i.e.: CATALOG, INIT, LOAD...) and a set of ERROR messages which, unless you are either a magician or don't use the Disk II, you have probably seen before.

The disk controller card that connects the Disk II to the Apple also has a small program on it. When you BOOT a disk, this program tells the Disk II to READ track \$00 (0), sector \$00

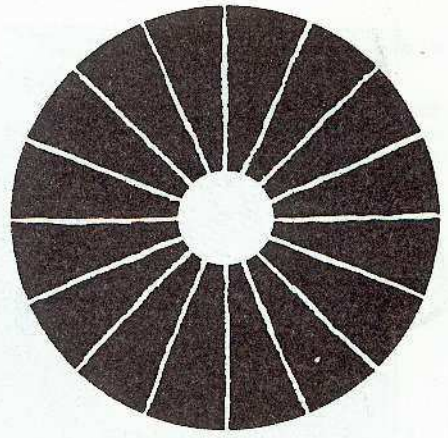
FIGURE 1

LEGAL BYTES

HEX	DEC	HEX	DEC
----- 13 Sector -----			
AA	170	AB	171
AD	173	AE	174
AF	175	B5	181
B6	182	B7	183
BA	186	BB	187
BD	189	BE	190
BF	191	D5	213
D6	214	D7	215
DA	218	DB	219
DD	221	DE	222
DF	223	EA	234
EB	235	ED	237
EE	238	EF	239
F5	245	F6	246
F7	247	FA	250
FB	251	FD	253
FE	254	FF	255
----- 16 Sector -----			
96	150	D5	213
97	151	D6	214
9A	154	D7	215
9B	155	D9	217
9D	157	DA	218
9E	158	DB	219
9F	159	DC	220
A6	166	DD	221
A7	167	DE	222
AA	170	DF	223
AB	171	E5	229
AC	172	E6	230
AD	173	E7	231
AE	174	E9	233
AF	175	EA	234
B2	178	EB	235
B3	179	EC	236
B4	180	ED	237
B5	181	EE	238
B6	182	EF	239
B7	183	F2	242
B9	185	F3	243
BA	186	F4	244
BB	187	F5	245
BC	188	F6	246
BD	189	F7	247
BE	190	F9	249
BF	191	FA	250
CB	203	FB	251
CD	205	FC	252
CE	206	FD	253
CF	207	FE	254
D3	211	FF	255



35 tracks

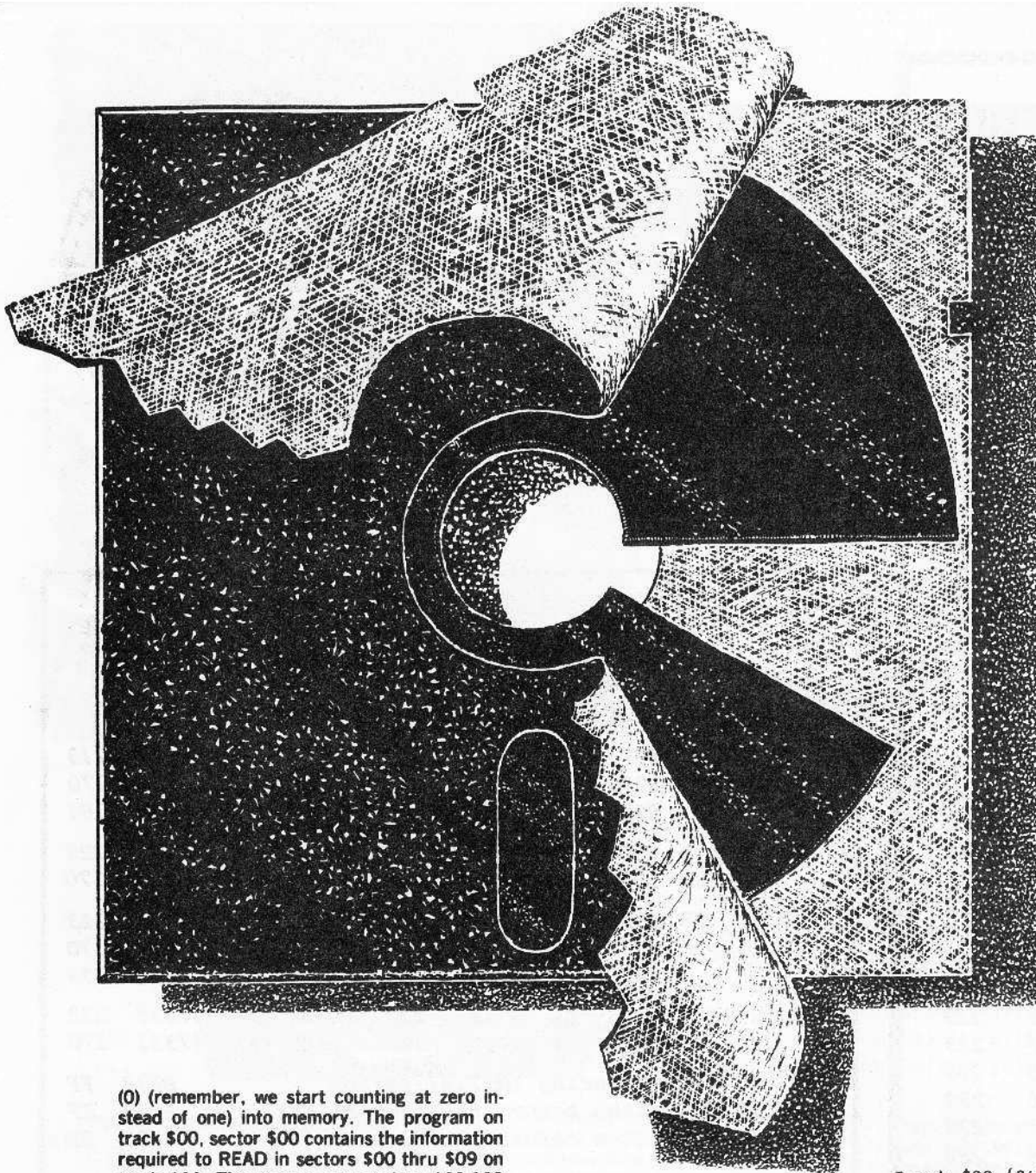


16 sectors

DOS ADDRESS AND DATA MARK LOCATIONS

DOS 3.2	READ LOCATIONS		WRITE LOCATIONS	
	-HEX-	-DECIMAL-	-HEX-	-DECIMAL-
START OF ADDRESS	B976	D5 47478 213	BEF5	D5 48885 213
	B980	AA 47488 170	BEFA	AA 48890 170
	B98B	B5 47499 181	BEFF	B5 48895 181
END OF ADDRESS	B9B2	DE 47538 222	BF29	DE 48937 222
	B9BC	AA 47548 170	BF2E	AA 48942 170
START OF DATA	B908	D5 47368 213	B893	D5 47351 213
	B912	AA 47378 170	B898	AA 47256 170
	B91D	AD 47389 173	B89D	AD 47261 173
END OF DATA	B956	DE 47446 222	B8DE	DE 47326 222
	B960	AA 47456 170	B8E3	AA 47331 170
SYNC BYTE used during INITIALization			BF38	FF
SYNC BYTE written before the Address Mark			BF73	FF
SYNC BYTE written before the Data Mark			B87E	FF

DOS 3.3	READ LOCATIONS		WRITE LOCATIONS	
	-HEX-	-DECIMAL-	-HEX-	-DECIMAL-
START OF ADDRESS	B955	D5 47445 213	BC7A	D5 48250 213
	B95F	AA 47455 170	BC7F	AA 48255 170
	B96A	96 47466 150	BC84	96 48260 150
END OF ADDRESS	B991	DE 47505 222	BCAE	DE 48302 222
	B99B	AA 47515 170	BCB3	AA 48307 170
START OF DATA	B8E7	D5 47335 213	B853	D5 47187 213
	B8F1	AA 47345 170	B858	AA 47192 170
	B8FC	AD 47356 173	B85D	AD 47197 173
END OF DATA	B935	DE 47413 222	B89E	DE 47262 222
	B93F	AA 47423 170	B8A3	AA 47267 170
SYNC BYTE written before the Address Mark			BC60	FF
SYNC BYTE written before the Data Mark			B83E	FF



A diskette with the cover peeled away to show the inner lining and actual plastic diskette

A sector section has been cut out and is shown below.

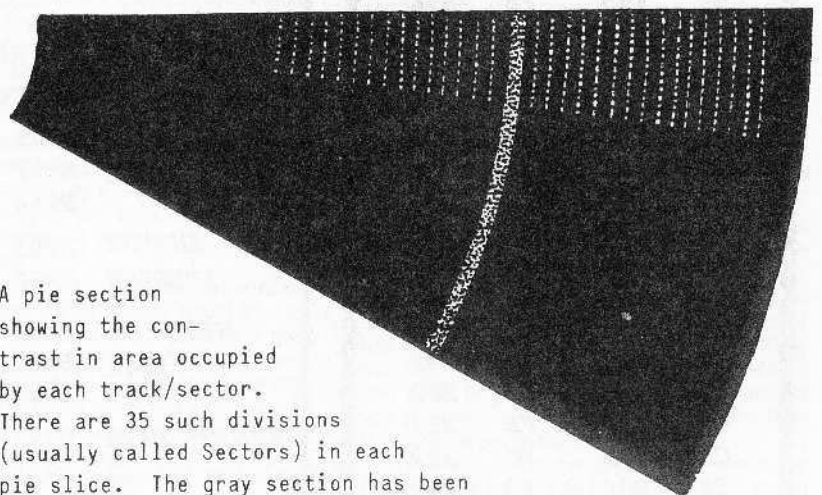
(0) (remember, we start counting at zero instead of one) into memory. The program on track \$00, sector \$00 contains the information required to READ in sectors \$00 thru \$09 on track \$00. The program on sectors \$00-\$09 READs in the remaining information on track \$00-\$02. Thus the Apple LOADs the DOS.

At this point DOS takes over and RUNs the program you have INITIALized the disk with.

In order to find your HELLO program DOS goes to the Volume Table of Contents (VTOC) and Directory located on track \$11 (17). The VTOC or "bit map" shows which sectors are in use and which are free. The Directory begins on sector \$0F (15) and continues down to sector \$01 (1). The VTOC and Directory are used by DOS whenever you SAVE or DELETE a file. The Directory contains a list of all the files on the disk. Each entry contains a pointer to the track/sector list, file-locked and file-type code, file-name and file-size. The track/sector list is a list of track/sector pairs that are used to store that program. This is why SAVEing a blank file always takes 2 sectors. One for the blank file and one for the track/sector list.

DOS formats a track by first writing a unique byte called a "sync byte". This byte (normally \$FF) allows the Disk II hardware to synchronize with the data on the disk. DOS then

Track \$22 (34) -- to -- Track \$00 (0)



A pie section showing the contrast in area occupied by each track/sector. There are 35 such divisions (usually called Sectors) in each pie slice. The gray section has been enlarged and is shown on the next page.

WRITES an address mark, some more sync bytes and the data mark. At this time the data field is full of \$00's.

The following is a normal address mark for 3.3 DOS:

D5AA96FFFEAABBAAEAFBEFDEAAEB

It can be broken down into:

Start of address D5 AA 96
 Volume number FF FE
 Track AA BB
 Sector AE AA
 Checksum FB EF
 End of address DE AA EB

The volume, track, sector and checksum are in a 4+4 coded format (see 4+4 conversion chart). This means that 4 bits in each byte are

actual data. The first byte is rotated left and logically ANDed with the second byte to recover the data.

The data mark consists of:

Start of data D5 AA AD
 Data field (342 bytes)
 Checksum (2 bytes)
 End of data DE AA EB

The data field is encoded in a 2+6 format. Six bits of each byte are valid data.

The basic structure of 3.2 DOS is similar to 3.3 DOS with these notable exceptions:

1. When initializing a disk, DOS 3.2 does not WRITE a blank data sector. Instead, it just WRITES enough \$FF's to fill the space a data sector would use. Trying to READ a track/sector that has never been written to will always generate I/O errors.

4 + 4 CONVERSION CHART

AA+AA=00	AE+FA=58	BB+EA=62	EA+BA=90	EF+EA=CA
AA+AB=01	AE+FB=59	BB+EB=63	EA+BB=91	EF+EB=CB
AA+AE=04	AE+FE=5C	BB+EE=66	EA+BE=94	EF+EE=CE
AA+AF=05	AE+FF=5D	BB+EF=67	EA+BF=95	EF+EF=CF
AA+BA=10	AF+AA=0A	BB+FA=72	EA+EA=C0	EF+FA=DA
AA+BB=11	AF+AB=0B	BB+FB=73	EA+EB=C1	EF+FB=DB
AA+BE=14	AF+AE=0E	BB+FE=76	EA+EE=C4	EF+FE=DE
AA+BF=15	AF+AF=0F	BB+FF=77	EA+EF=C5	EF+FF=DF
AA+EA=40	AF+BA=1A	BE+AA=28	EA+FA=D0	FA+EA=E0
AA+EB=41	AF+BB=1B	BE+AB=29	EA+FB=D1	FA+EB=E1
AA+EE=44	AF+BE=1E	BE+AE=2C	EA+FE=D4	FA+EE=E4
AA+EF=45	AF+BF=1F	BE+AF=2D	EA+FF=D5	FA+EF=E5
AA+FA=50	AF+EA=4A	BE+BA=38	EB+AA=82	FA+FA=F0
AA+FB=51	AF+EB=4B	BE+BB=39	EB+AB=83	FA+FB=F1
AA+FE=54	AF+EE=4E	BE+BE=3C	EB+AE=86	FA+FE=F4
AA+FF=55	AF+EF=4F	BE+BF=3D	EB+AF=87	FA+FF=F5
AB+AA=02	AF+FA=5A	BE+EA=68	EB+BA=92	FB+EA=E2
AB+AB=03	AF+FB=5B	BE+EB=69	EB+BB=93	FB+EB=E3
AB+AE=06	AF+FE=5E	BE+EE=6C	EB+BE=96	FB+EE=E6
AB+AF=07	AF+FF=5F	BE+EF=6D	EB+BF=97	FB+EF=E7
AB+BA=12	BA+AA=20	BE+FA=78	EB+EA=C2	FB+FA=F2
AB+BB=13	BA+AB=21	BE+FB=79	EB+EB=C3	FB+FB=F3
AB+BE=16	BA+AE=24	BE+FE=7C	EB+EE=C6	FB+FE=DE
AB+BF=17	BA+AF=25	BE+FF=7D	EB+EF=C7	FB+FF=DF
AB+EA=42	BA+BA=30	BF+AA=2A	EB+FA=D2	FE+EA=E8
AB+EB=43	BA+BB=31	BF+AB=2B	EB+FB=D3	FE+EB=E9
AB+EE=46	BA+BE=34	BF+AE=2E	EB+FE=D6	FE+EE=EC
AB+EF=47	BA+BF=35	BF+AF=2F	EB+FF=D7	FE+EF=ED
AB+FA=52	BA+EA=60	BF+BA=3A	EE+BA=98	FE+FA=F8
AB+FB=53	BA+EB=61	BF+BB=3B	EE+BB=99	FE+FB=F9
AB+FE=56	BA+EE=64	BF+BE=3E	EE+BE=9C	FE+FE=FC
AB+FF=57	BA+EF=65	BF+BF=3F	EE+BF=9D	FE+FF=FD
AE+AA=08	BA+FA=70	BF+EA=6A	EE+EA=C8	FF+EA=EA
AE+AB=09	BA+FB=71	BF+EB=6B	EE+EB=C9	FF+EB=EB
AE+AE=0C	BA+FE=74	BF+EE=6E	EE+EE=CC	FF+EE=EE
AE+AF=0D	BA+FF=75	BF+EF=6F	EE+EF=CD	FF+EF=EF
AE+BA=18	BB+AA=22	BF+FA=7A	EE+FA=D8	FF+FA=FA
AE+BB=19	BB+AB=23	BF+FB=7B	EE+FB=D9	FF+FB=FB
AE+BE=1C	BB+AE=26	BF+FE=7E	EE+FE=DC	FF+FE=FE
AE+BF=1D	BB+AF=27	BF+FF=7F	EE+FF=DD	FF+FF=FF
AE+EA=48	BB+BA=32	EA+AA=80	EF+BA=9A	
AE+EB=49	BB+BB=33	EA+AB=81	EF+BB=9B	
AE+EE=4C	BB+BE=36	EA+AE=84	EF+BE=9E	
AE+EF=4D	BB+BF=37	EA+AF=85	EF+BF=9F	

SECTOR FORMAT



Start of Address
 Volume Number
 Track
 Sector
 Checksum
 End of Address
 Break (random bits)
 Sync Bytes
 5 - 10 bytes
 Start of Data
 Data Field
 342 bytes
 Checksum
 End of Data

2. The data is encoded in a 3+5 format which requires 410 bytes to encode 256 data bytes. This is one reason why there are only 13 sectors.

For those of you who are interested in more details and think I glossed over some things (and I did), the best book I have seen that talks about DOS is BENEATH APPLE DOS by Don Worth and Pieter Lechner.... A worthwhile addition to any Apple computerists library.

There are two things to consider when you decide to protect your disk:

1. Protecting against normal DOS copiers and,
2. Protecting against bit (nibble) copiers.

The first is fairly simple. Almost any change in the normal disk format will cause DOS to generate an I/O ERROR. The end-of-address and end-of-data bytes (DE AA) are not used in the initial BOOT. Therefore, changing these bytes will allow a disk to BOOT but not allow normal DOS to copy it.

The easiest method of accomplishing this is to change the appropriate locations in DOS and INITIALize a disk with the changes. However, LOADing anything other than your "hello" program onto this disk could be tedious. You would have to change the DOS back to normal, LOAD your program from a normal disk then change those same DOS locations and SAVE the program to your protected disk.

Another method would be to LOAD the programs from the cassette input and SAVE them to the disk.

The best method would be to use "Muffin" to upLOAD your programs from a normal disk. This works because Muffin uses an internal DOS (RWTS) image to READ data and resident DOS to WRITE data.

A more elegant method is to take advantage of the way that DOS WRITES data to the disk. When DOS WRITES a sector, it first looks for the address mark. When it has found the correct mark for the sector it wants, DOS WRITES a whole new data mark. It first WRITES 5 sync bytes then the complete data mark with the new data. Every time you SAVE data to a disk this process is repeated.

NOTE: The address mark is only written once when the disk is INITIALized.

A program could be written that would READ the data from a normal disk and WRITE it to another in a changed format. This changed disk would not be copyable by any normal DOS. Table 1 is a list of address and data mark locations and their normal values for a 48K Apple.

CAUTION: Always change both the READ and WRITE locations for any byte you alter. Also, not all values are valid. See table 2 for a list of valid bytes to use when making changes.

The second part of disk protection involves bit or nibble copiers.

These programs do not depend on a rigid structure of address and data marks to find valid data. Instead they make as few assumptions about the disk format as possible. These copiers use pattern recognition to find the data. They are not as reliable as a normal DOS copy

program since they don't use checksums.

Many of the current bit copiers do make one assumption. They look for a \$FF sync byte. These copiers can be defeated by INITIALizing a disk with a different sync byte.

I am sure that the new crop of bit copiers will have corrected this oversight but, until then, this is a valid protection method. The location of the sync bytes are also given in Table 1.



Bobby has been quite busy lately and he expects to become busier in the future so his columns may be printed on a more irregular basis. For this reason, He has put as much into this DiskLocks as he has had time for. Like the rest of us, he is quite busy, but we'll be printing anything he sends us. Don't forget to read his SoftKey part 2!

If you have more information on the subject of Copy Protection schemes, please send us a letter including that valuable information.

----editor

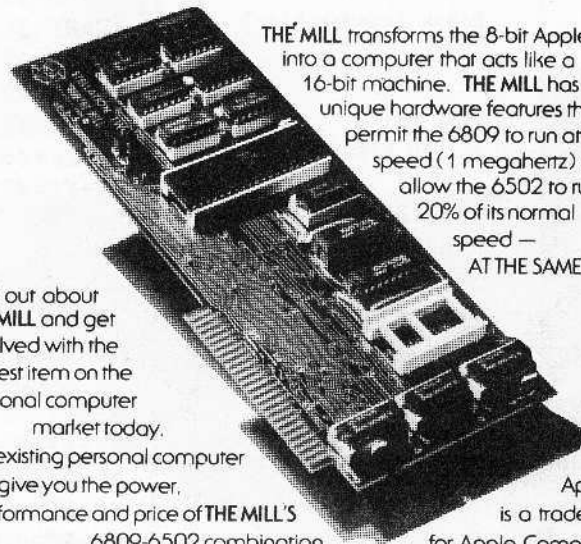
Your Apple too slow? Not anymore...

Now you too can write 6809 programs for your Apple II that are DOS 3.3 compatible. But you don't have to stop there, you can also program your Apple II's 6502 and the 6809 of THE MILL to run SIMULTANEOUSLY.

THE ASSEMBLER DEVELOPMENT KIT, including THE MILL, is a full feature assembler, designed to use the text editing system of your choice. The system will also boost your computer programming productivity, since the 6809 is today's easy to learn and program computer. Take advantage of the 8-bit 6502 and the 16-bit abilities of the 6809 running at the same time, create your own MULTIPROCESSING ENVIRONMENT on the Apple II.

Put THE MILL into your Apple II and get power, performance and price matched by no other personal computer. STELLATION TWO offers a full 1 year warranty and a 60 day money back guarantee, if you're not completely satisfied with the power of THE MILL.

Shop around, then compare the service, quality, price and power of THE MILL. Take this ad to your local Apple Dealer and see the endless possibilities of adding THE MILL to your Apple II.

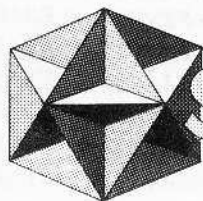


THE MILL transforms the 8-bit Apple II into a computer that acts like a 16-bit machine. THE MILL has unique hardware features that permit the 6809 to run at full speed (1 megahertz) and allow the 6502 to run at 20% of its normal speed — AT THE SAME TIME!

Find out about THE MILL and get involved with the hottest item on the personal computer market today.

No existing personal computer can give you the power, performance and price of THE MILL'S 6809-6502 combination.

Apple II is a trademark for Apple Computer, Inc.



STELLATION TWO

P.O. BOX 2342 - H2
SANTA BARBARA, CA. 93120
(805) 966-1140

for Apple II+ without an INTEGER firmware card

How To Back Up Copy- Protected Disks

a continuing series...
by BOBBY

For more information on making back-ups of non-standard diskettes, see:

HARDWARE SOLUTIONS on page 45.

DISKLOCKS on page 24.

REVIEW of NIBBLES AWAY, another 'bit' copy program, on page 14.

DISKEDIT 2.1 listing on page 60.

CHR\$(4) and DISKVIEW listing on page 42.

Many Apple II+ owners do not own an Integer firmware card. This method is especially appropriate for you.

Do you have a disk that boots on a 13 sec-

tor and a 16 sector Apple? The chances are good that the disk follows a 13 sector format. Muffin was created to move 13 sector files to a 16 sector disk. With a few changes, it can also be used to READ nonstandard disks.

USING SOFTKEY MUFFIN13

Requirements:

- Apple II or AppIIe with 48K MUFFIN13 or MUFFIN 16...
- Blank initialized disk (16 or 13 as appropriate)...
- DiskView (or other nibbler)...

Perform the following steps:

1. **RUN DiskView**
2. Remove the DiskView disk and insert the back-up copy of your program disk.

NEVER USE THE ORIGINAL DISK

3. **READ track 3** and look for the starting address and data marks. The normal values are:

```
D5 AA B5 (address)
D5 AA AD (data)
```

These bytes can be located by looking for the sync bytes. These are a group of similar bytes or pattern of bytes. (ie. a series of FF's or a pattern that repeats. D5 AB D5 AB D5 AB etc. See Disk Locks)

4. **Write down the values that you find.** Skip about 6 or 7 tracks and READ again.

5. **Repeat step 4 several times.** If the values you found were not the same on each track then this method will not work for you. There is one exception and that is if the same byte was changed each time and that byte is the last byte of the data mark (ie. D5 AA FE D5 AA DD D5 AA FD).

NOTE 1: See disklocks for more information on address and data marks. Practice on a normal DOS disk until you are familiar with the disk format and can easily find the address and data marks.

NOTE 2: Check a higher track (ie. track 3 thru 34) and use the 4 4 Conversion Chart in Disklocks to convert the sector bytes for each address mark to hex. Verify that the sector numbers range from 0 to 12. If the sectors are numbered differently (ie. by even numbers. 0,2,4,6,8,10 etc.) then this method will not work.

6. **BLOOD MUFFIN13**
7. **CALL -151**

----- NOTE -----

Replace the boxed symbols with the proper address or data mark.

Replace: with:

A1	1st byte of address mark
A2	2nd byte of address mark
A3	3rd byte of address mark
D1	1st byte of data mark
D2	2nd byte of data mark
D3	3rd byte of data mark

8. **Type in the following changes and use the values you found in steps 3 thru 5:**

```
1A76: A1 Return
1A80: A2 Return
1A8B: A3 Return
1A08: D1 Return
1A12: D2 Return
1A1D: D3 Return
```

NOTE: If your disk uses a variable third byte in the data mark, then make the following additional change:

```
for MUFFIN13 only...
1A1E:EA EA Return

for MUFFIN16 only...
19FD:EA EA Return
```

9. 803G return

10. Respond to the prompts with the appropriate slot and drive for the source and destination disks. Be sure to insert the disks in the correct drives. If you are using only one drive, label the disks clearly to avoid confusion. Respond to the file name prompt with the equals sign (=) and type **N** to "do you want prompting". If you are using one drive, the program will still prompt you to switch disks when necessary.

There's MORE on SOFTKEYs on the next page:

HOW TO USE MUFFIN 16, and
the SOFTKEY TO INTEGER

Soft Key

Here is some supplementary information for the open-heart surgery method presented in issue 1.

TO integer PROGRAMS

by BOBBY

Several people have asked about how to find Integer programs. The zero page pointers for Applesoft and the normal DOS pointers for Binary were given in Softkey part I. The zero page pointers for Integer are:

Start of program...
\$CA, \$CB (202,203)

End of program...
\$4C, \$4D (76,77)

These pointers tell your computer where the program is in memory. The hex address is in normal lo-byte,hi-byte format.

Use these pointers and apply them in the OPEN HEART SURGERY METHOD revealed in the Premier Issue, page 27. Good luck. If the DOS is not too altered, it will work. For those who don't have the integer card, see HARDWARE SOLUTIONS, page 45.



---using MUFFIN16---

The steps for using MUFFIN16 are the same as above except for the following:

a. In step 3, the bytes to look for are:

D5 AA 96 Address
D5 AA AD Data

b. In step 6, BLOAD MUFFIN16.

c. Replace step 8 with:

8. Type in the following changes and use the values you found in steps 3 thru 5:

1A55: **A1** Return
1A5F: **A2** Return
1A6A: **A3** Return
19E7: **D1** Return
19F1: **D2** Return
19FC: **D3** Return

(of course, replace boxed symbols with the appropriate byte.)






the Computerist's Directory

The national phone book of computing

\$7.50

"Let your fingers do the talking"



Alphabetical Index

Key Word Finder

Bulletin Boards

User Groups & Associations

Contents

White Pages

Yellow Pages

FOR JUST \$10.00 YOU GET

1) a Free copy of current issue
2) a One Year Subscription!
3) a Free 25 word White Page Listing!
4) \$5.00 Credit on a Yellow Page Listing!
5) \$10.00 Credit towards any size Display Ad!

Total Value = \$37.50

VISA
 MASTER CHARGE
 AMERICAN EXP

ACCOUNT NO. _____

Save \$18.50 and Join the Information Revolution, **Subscribe Now**

Send me Display ad rates Send me your retail store order form.

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

ENCLOSED \$

The Computerist's Directory is becoming the one reference to your products.

The Computerist's Directory is published twice annually in January and July with quarterly updates planned for April 82.

THE NEW SOFTKEY

MUFFINS

FOR 3.2 DISKS

how to make

MUFFIN 13

FOR 3.3 DISKS

how to make

MUFFIN 16

by BOBBY

Muffin is a program on your 3.3 DOS Master disk. Those of you with 3.2 DOS will need to get a copy from a friend or your local dealer. Muffin is used to transfer 13-sector files to 16-sector disks. To do this, it contains an image of 13-sector RWTS (Read or Write a Track and Sector). It uses the 13-sector RWTS to read a file and then writes to your 3.3 diskette using the resident DOS.

Muffin13 is created by changing the RWTS image so that it doesn't use the checksum or end marks.

1. BLOAD MUFFIN
2. Type in the following POKES:

POKE 6826,24 return
POKE 6827,96 return
POKE 6734,24 return
POKE 6735,96 return

3. BSAVE MUFFIN13, A\$803, L\$18FD

Muffin can be modified to READ files on 16-sector disks. You'll need the Programmer's Aid ROM (The Integer card comes with this ROM).

1. Type INT return (for integer card)
2. BLOAD MUFFIN
3. CALL -151
4. Type in the following:

A. D4D5G return

Initializes the Programers Aid Code-Relocation feature. See page 16 of your Programers Aid Manual (PAM).

B. 1900<B800.BFFF ctrl Y*
return

Tells the relocate routine what we're moving and where it goes. See page 16, PAM.

C. 1900<B800.BA10 ctrl Y
return

Moves the first Code segment down to \$1900. See page 17, PAM.

D. .BC57M return

Moves the Data segment. See page 17, PAM.

E. .BFFF ctrl Y return

Moves the last Code segment.

F. POKE 6701,24 return
POKE 6702,96 return
POKE 6793,24 return
POKE 6794,96 return

5. BSAVE MUFFIN16, A\$803, L\$18FD



ROMPLUS* NEEDS ROMS FOR APPLES

RESIDENT FIRMWARE UTILITY PROGRAMS WILL EXPAND THE CAPABILITIES AND INCREASE THE USEFULNESS OF YOUR APPLE II OR APPLE II PLUS! *

*APPLESOFT EDITROM

Global search, change, or remove any string, variable, literal, constant, or basic command word that appears in your Applesoft Programs. EDITROM uses no ram space that will interfere with your program. It does not reset any system pointers to protect itself and will operate with any size system - 16K, 32K, or 48K. After EDITROM has been initialized, the ampersand (&) command can then be used to call the EDITROM back for repeated use without readdressing the ROM + BOARD. Completely compatible with Konzen's Program Line Editor. If PLE is up, EDITROM will keep PLE up and allow joint operation. Will operate with any version of DOS and requires FP in Rom. 35.95

COMMANDROM

COMMANDROM is like having a resident 'FID' but with more operating features and conveniences. COMMANDROM will read a disks (13 or 16 Sector) File Directory Listing and display the following: A Command Menu, current drive number, number of sectors used and left, number of pages set up to hold all file names, the first page of file names, and an identification letter next to each file name on display. Pressing any one of the command keys will load or run any file (A-B file load will display start and length addresses), lock or unlock a file or all files, delete a file - with verify before deletion, change from one drive to the other, read a new disk, display a Track/Sector Map, change page numbers to view all file listings without recataloging, or exit to current language or monitor. No system pointers are reset and no RAM is reserved for COMMANDROM. Requires 48K, 3.2 or 3.3 DOS and INT or FP in ROM 35.95

BASICSROM

Will boot a 13 sector disk on a system configured for 16 sector operation. The BASICSROM can be addressed on coldstart (without Auto Start ROM or warmstart (with Auto Start ROM) at any time. 35.95

DISK COPY/SPACE ROM

Duplicates a disk, from either SINGLE or DUAL drive, single or dual controller, 13 or 16 sector and with or without VTOC. Options include - Gross copy, active sectors only copy, DOS overwrite, auto boot of copy disk, free space on disk in sectors and kilobytes, and Init and volume number are selectable. Requires a minimum of 32K. 35.95

*APPLESOFT RENUMBER/ MERGEROM

Made famous by Apple Computer Inc., this powerful utility will not disturb any part of a program in memory, when it is activated. Requires 48K, with or without Disk II. 35.95

'YOUR' PLE ROM

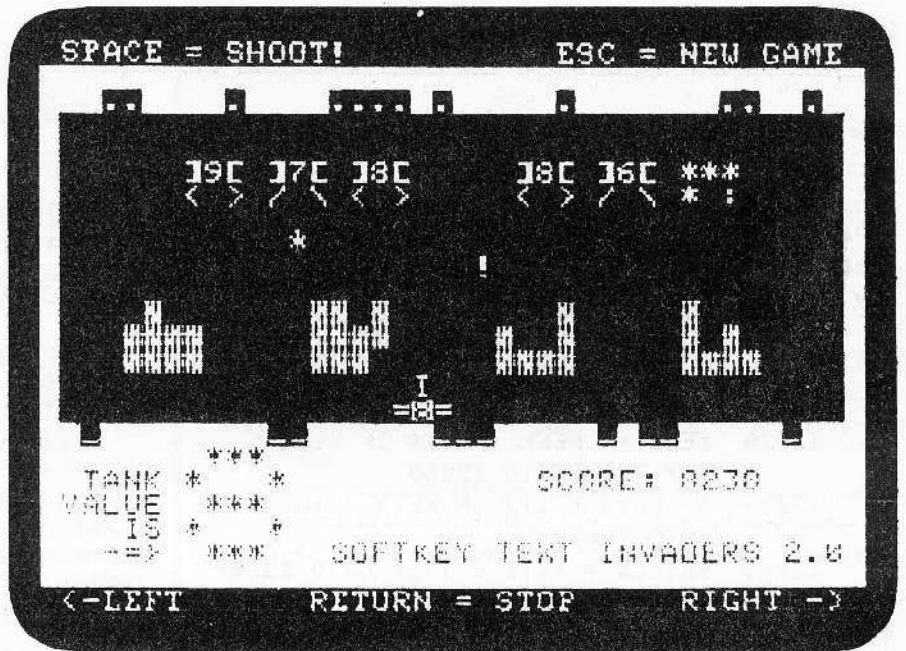
Now you can put your Program Line Editor in ROM. Write for details. \$45.95 REGISTERED TRADEMARKS

BOX 599
WEST MILFORD, N.J. 07480
SOFT CTRL SYSTEMS

201-728-8750

TEXT invaders 2.0

by
Dev.
R.
haight



```
10130 POWER% = 9: TTRAVEL% = 2
10140 HGUN% = 19: VGUN% = 17: DISPLAY%
      = 1: AGAIN% = 1
10150 BUNKER% = CHR% (220) + CHR%
      (239) + CHR% (220) + CHR% (239)
10160 IVRAY% = 0: IHRAY% = 0: THRAY%
      = 2
10180 ZAP%(0) = "*": ZAP%(1) = " ":
      ZAP%(2) = " "
```

Tank Hit Power Numerals

```
10500 DATA " *** ** ***** ** *** "
10510 DATA " ##  ##  ##  ##  ##  "
10520 DATA "***** ** *** ** *****"
10530 DATA "#####  ##  ####  ##### "
10540 DATA " ** ***** ***** ** ** "
10550 DATA "#####  ####  ##### "
10560 DATA " ** ** ***** ** ** ** "
10570 DATA "#####  ##  ##  ##  ##  "
10580 DATA " *** ** ** *** ** ** *** "
10590 DATA " ### ##  ##  ####  ##  ##  "
```

Fill Special STR%

```
10800 ERR% = "CHOOSE A NUMBER FROM 1
      TO 5"
10900 FOR A = 1 TO 40: SPACE% = SPACE%
      + " ": NEXT A
10910 FOR A = 0 TO 9: READ POWER%(A):
      NEXT A
```

Print Graphics On Text Page

```
11000 INVERSE : FOR A = 3 TO 24: VTAB
      A: HTAB 1: PRINT SPACE%: NEXT A
11010 VTAB 19: HTAB 24: PRINT "SCORE:"
11020 VTAB 19: HTAB 2: PRINT "TANK":
      PRINT "VALUE": HTAB 4: PRINT
      "IS": HTAB 3: PRINT "-->";
11030 VTAB 22: HTAB 14: PRINT
      "SOFTKEY TEXT INVADERS 2.0";
11050 NORMAL: VTAB 24: HTAB 2
11060 PRINT "<-LEFT": HTAB 15: PRINT
      "RETURN = STOP": HTAB 33:
      PRINT "RIGHT->";
11070 VTAB 1: HTAB 2: PRINT "SPACE =
      SHOOT!": HTAB 26: PRINT "ESC =
      NEW GAME";
```

Make and clear game window

```
11100 POKE 32,1: POKE 33,38: POKE
      34,3: POKE 35,16
11110 VTAB 3: HTAB 1: CALL - 958
11120 VTAB 3: HTAB 1: INVERSE : PRINT
      RIGHT% (SPACE%,38);
```

Print Bunkers

```
11200 FLASH : FOR A = 12 TO 14: VTAB
      A: HTAB 4: PRINT BUNKER%: HTAB
      13: PRINT BUNKER%: HTAB 22:
      PRINT BUNKER%: HTAB 31: PRINT
      BUNKER%: NEXT A: NORMAL
```


continued from page 5 ...

Various Noise Routines

```

1500 FOR A = 0 TO 16 - VINVADER%: IF
FRHYTHM% > 0 THEN 1520
1510 ZZ = PEEK (NOISE) - PEEK (NOISE):
NEXT : RETURN
1520 ZZ = PEEK (NOISE) + PEEK (NOISE):
NEXT : RETURN
1530 ZZ = PEEK (NOISE): RETURN

```

Invaders Move

```

2000 IF LEN (INVADER$(1)) < 5 THEN
20000
2010 IF ITRAVEL% = LEFT% THEN 2500

```

Invaders March Right

```

2400 HINVADER% = HINVADER% + 1: IF
HINVADER% > 37 - LEN ( INVADER$(
(1)) THEN ITRAVEL% = LEFT%: VTAB
VINVADER%: HTAB 1: CALL - 868:
IF SKILL% < 3 THEN VINVADER% =
VINVADER% + 1

```

Print Invaders

```

2410 VTAB VINVADER%: HTAB 1: CALL
- 868: VTAB VINVADER% + 1: HTAB
1: CALL - 868
2420 VTAB VINVADER%: HTAB HINVADER%:
PRINT INVADER$(1): VTAB VINVADER%
+ 1: HTAB HINVADER%: PRINT
INVADER$(FRHYTHM% + 1):
2430 FRHYTHM% = - FRHYTHM%: GOSUB
1500: GOTO 1000

```

Invaders March Left

```

2500 HINVADER% = HINVADER% - 1
2510 IF HINVADER% < 2 THEN VTAB
VINVADER%: HTAB 1: CALL - 868:

ITRAVEL% = RIGHT%: VINVADER% =
VINVADER% + 1
2520 GOTO 2410

```

continued on page 36

should concentrate on this subject and leave the other types of programs (eg. entertainment) to NIBBLE and SOFTSIDE and computer club magazines. Less complaining about censorship would also improve the magazine. Specific suggestions include:

1. Further discussion of Locksmith and other copy programs. What and how do you determine the Locksmith parameters? Give copy parameters for different programs.

2. give more specific examples of how to crack copy-protected disks including the copy programs themselves. I am glad you will give copying techniques for the Apple II+ with language card owners.

3. Review Locksmith 4.0 in detail. I see you will review Nibbles Away shortly.

4. Tutorials on DOS and disk formats. The DiskLock is a good start. You should also give examples of how this information is used.

5. I don't see the purpose of the last Alert and the Garbage Award. Concentrate on the positive since people are paying a lot for your magazine.

Enough preaching--

Good luck,

James Williamson

Greenbelt, MD

[This next letter requires the reprinting of a news item that appeared in Update 1.1:

Creative Computing Censors Hardcore Advertisement

Creative Computing is still censoring ads and suppressing information about copy protection. It has just refused to publish the **HARDCORE** advertisement shown on this page.

What did the editors of CC find unprintable in our ad?

"Only two lines," explained their Advertising Manager, Jerry Thompson. "They are: 'back up any diskette', and 'do & undo copy-protection'."

Jerry said that either we agree to let them remove those lines or they would refuse to run the ad. Well, we stood by our editorial policy of fighting censorship and they stood by their policy of information suppression. So you won't see this ad in CC.

One of their editors, George Blank, defended CC's policy by offering the lame comparison to Reader's Digest's policy against running liquor ads. He added that at least CC will run ads for competitors.

Big Deal! All of them share CC's censorship policy. All of them... except **HARDCORE COMPUTING**. And they will not run our ads.

George's final defense was that CC had a readership of well over 100,000... so obviously they must be doing things right. What can I say to that?

And when I mentioned that the readers have had no alternative, he answered promptly, "Yes, they do!" (meaning his competitors). I retorted that they practice the same form of censorship.

"That might mean that we're right and you're wrong," he concluded.

continued on page 37

continued from page 35...

So... The battle goes on.
 CC is probably just the first computer magazine to censor
HARDCORE's ads. We plan to try to put one in Call A.P.P.L.E., but
 unless Val Golding, the editor, has altered his policy, they will prob-
 ably refuse to run our ad, too.
 We'll keep you posted on further developments.

Addendum: Call A.P.P.L.E. didn't run our ad, but we
 learned from Grawin Publications that it wasn't Val
 who had our ad pulled from its pages at the last mo-
 ment. It was the doing of Dick Hubert, President of
 Call A.P.P.L.E., and its Executive Director.

Anyway, on to the letter (parenthesis bv bev.):

Dear Editor,

You quoted me correctly (in the update and alert),
 and for that I am grateful. I do not feel that you
 quoted me in context. For example, my "lame excuse"
 that Reader's Digest refused liquor ads came at a
 point in our conversation when we were discussing the
 fact that the courts have allowed publishers the right
 to set their own standards for accepting advertising.

I invite you to practice what you preach. If soft-
 ware prices are extortionate, how do you justify \$20
 a year for your few miserable pages of hard-to-read
 mimeographed text? For the same price, your readers
 could buy over 2500 pages of Apple material printed in
 Creative Computing- or even over 400 pages of Apple
 material if you want to throw away all our ads, gener-
 al articles, and material on other computers!

The truth is most current software companies are
 not profitable. We know; we keep getting bankruptcy
 notices instead of payment for advertisements, even
 from established firms. And it is casual copying, not
 professional pirates, that hurts profits most. In my
 own case, I no longer write programs for publication;
 it is much more profitable to write custom software.

Sincerely yours,
 George Blank,
 Editor, Creative Computing

Dear Sir,

I received the first issue of **HARDCORE** in the
 mail a few days ago and am quite pleased. I hope that
 you get as big as **BYTE** and do not degenerate into a
 game oriented publication.

There are a few comments that I want to make
 about some of the articles and editorial columns.
 First, let me say that I, too, have felt the need for
 a magazine such as **HARDCORE**. The main reason I sub-
 scribe to a computer magazines is for the useful pro-
 gramming tips that I can pick up. When the tips stop,

continued on page 39

-----Additional lines-----

```
1020 GOSUB 100: REM <CHECK KEYBOARD>
11210 GOSUB 400: REM <MOVE TANK>
12210 GOSUB 400: REM <MOVE TANK>
```

4 the invaders attack

You are now ready for step 4 where you will witness the wholesale
 slaughter of your tank and your bunkers. Key in step 4: The Invaders
 Attack!

The sequence here is more difficult to explain because it consists
 of numerous subroutines. The most important of these is the Attack
 Sequence itself which will be explained first.

A. The attack itself begins (900) with a random number be-
 tween 1 and the number of original invaders, N%, [between 4 and 8].
 This number times 4 plus 3 gives the "centers" of each invader in
 INVADERS{1}. The old invaders are erased and the random number,
 RAN%, is checked for validity.

B. That particular "center" is then:

```
1. Plucked out of INVADERS{1}:
910 TEMP% = MID$( INVADERS{1} ), RAN% + 1, 1)
2. Turned into a number:
920 TEMP% = VAL (TEMP%)
3. Dropped down a notch in value:
940 TEMP% = TEMP% - 1
4. Turned back into a string:
TEMP$ = STR$(TEMP%)
5. Stuck right back into INVADERS{1}:
970 INVADERS{1} =
LEFT$( INVADERS{1}, RAN% ) +
TEMP$ +
RIGHT$( INVADERS{1}, LEN (INVADERS{1} ) - RAN% - 1)
```

C. If the invader is dead, then it will get rid of its legs in almost
 the same way...[980]. Lack of hit power [and therefore no more
 bombs to drop!] is another reason for the invader's demise and disap-
 pearance. Later, in step 5, this same routine is used to decrement the
 invaders when you hit them with your missiles.

D. Condense the invaders by removing all spaces on the right
 side [660] and the left side [670], keeping the string compact and
 changing the HTAB value appropriately.

The next phase in step 4 is the sequence followed when the screen
 is examined in order to determine if the invader's bombs have hit
 anything.

The most important part here is, of course, the actual routine that
 examines the text screen memory. This is accomplished by using the
 SCRN command [see page 87 of your Applesoft manual. Normally, the
 SCRN(x,y) command will return the color code [0-15] of the x,y coor-
 dinates [between 0 and 39 for "x", 0 and 47 for "y"].

Since each text character is composed of two such color codes, a
 formula must be used that will return the text character instead of the
 colors. The formula given on page 87 of the Applesoft manual is:

```
CHR$( SCRN [x-1, 2 * [y-1] + 16 * SCRN [x-1, 2 * [y-1] + 1] ] )
...which will return the character at position [x,y].
```

In this program, a "window" has been used which causes this for-
 mula to give an incorrect value because it is looking at the wrong spot
 on the screen. The x value must be incremented by 1 [line 510] before
 passing the value to this subroutine [lines 10-20]. I've removed the
 CHR\$ function because I'm using PEEKed keyboard values. Here is
 what the program does:

- A. Check the Screen. Has the bomb hit anything?
 1. Empty space...continue on...line 520.
 2. A "!"...hit a missile!...line 530.
 3. Flashing Slashes...hit the bunkers...line 540.
 4. Oh no, too low! Hit the dirt!...line 550.
 5. Must have hit a tank!...lines 570-580.

continued on page 38

B. Whatever the bomb has hit, let it be destroyed...Here I've tried to include various modes of destruction in order to show the assortment available.

1. The tank goes boom [13200-13290]! This is the most complicated destruction scene except for the way the invaders will die [more on that in step 5].

2. The bunker goes zap [850-890]! This scene simply buzzes while flashing from inverse to normal a few times.

3. The bomb hits the ground! This does a short buzz and leaves a crater after the screen flashes back and forth from text page 1 to text page 2.

4. Your missile gets hit! The two vanish in a noisy incandescence of slashes, "I"s and dashes...

Finally, go ahead and type in lines 20000-20090 preparatory to step 5. Now RUN and Debug. When it is running correctly, no error statements should arise, and the bombs should be dropping one by one. The tank should self-destruct only when hit by a bomb, then the large number on the lower left of the screen should change. To test these numbers, try to get the tank destroyed as many times as you can.

<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>

=====
Get ASCII of SCRN
=====

```
10 XXKEY% = X% - 1: YYKEY% = 2 *
   (Y% - 1)
20 XKEY% = SCRN ( XXKEY%, YYKEY% ) +
   16 * SCRN ( XX%, YY% + 1 ): RETURN
```

=====
Has Invader Bomb Hit Anything
=====

```
500 IF IHRAY% = 0 THEN 900
510 IVRAY% = IVRAY% + 1: X% = IHRAY%
   + 1: Y% = IVRAY%: GOSUB 10
```

----- (Empty Space) -----

```
520 IF XKEY% = 160 THEN 700
```

----- (+) -----

```
530 IF XKEY% = 161 THEN 800
```

----- (bunker) -----

```
540 IF XKEY% = 92 OR XKEY% = 111
   THEN 850
```

```
550 IF IVRAY% > 17 THEN 750
570 INVERSE : GOSUB 13200: POWER% =
   POWER% - 1: IF POWER% = 0 THEN
   21000
```

```
580 VTAB VGUNX - 2: HTAB IHRAY%:
   IHRAY% = 0: PRINT " ";
```

```
590 GOSUB 400: RETURN
```

=====
Condense Invader STR\$
=====

-----<< Condense Right Side >>-----

```
660 IF MID$ (INVADER$(1), LEN
   (INVADER$(1)) - 2,1) = " " THEN
   FOR AA = 0 TO 2: INVADER$(AA) =
   LEFT$ (INVADER$(AA), LEN (INVADER$
   (AA)) - 4): NEXT AA: IF LEN
   (INVADER$(1)) > 4 THEN GOTO 660
```

-----<< Condense Left Side >>-----

```
670 IF MID$ (INVADER$(1),4,1) = " "
   THEN FOR AA = 0 TO 2: INVADER$(AA)
   = " " + RIGHT$ (INVADER$(AA), LEN
   (INVADER$(AA)) - 5): NEXT AA:
   HINVADER% = HINVADER% + 4: IF LEN
   (INVADER$(1)) > 4 THEN 670
690 RETURN
```

=====
Print Invader Bomb
=====

```
700 VTAB IVRAY%: HTAB IHRAY%: PRINT
   "+": VTAB IVRAY% - 1: HTAB
   IHRAY%: PRINT " ": RETURN
```

=====
Bomb Hits Bottom
=====

```
750 VTAB IVRAY% - 1: HTAB IHRAY%:
   PRINT " ": FOR T = 1 TO 5
770 POKE -16299,0: INVERSE : VTAB
   IVRAY%: HTAB IHRAY%: PRINT "^":
   GOSUB 1530: POKE -16300,0
780 NORMAL : HTAB IHRAY%: PRINT CHR$
   (223): GOSUB 1530
790 NEXT T: IHRAY% = 0: RETURN
```

=====
Bomb Hits Tank's Ray
=====

```
800 FOR BOOM = 1 TO 10: ZZ = PEEK
   (NOISE)
810 VTAB TVRAY% - 1: HTAB THRAYS% - 1:
   PRINT CHR$ (220): "I/": VTAB
   TVRAY%: HTAB THRAYS%: PRINT
   "--#--": VTAB TVRAY%: HTAB THRAYS%:
   PRINT "/I": chr$ (220): ZZ = PEEK
   (NOISE)
```


READER INPUT

continued from page 37

my subscription payments also stop. I am glad that I saw your ad in SOFTALK.

On using the "reverse" side of disks; I understand that some disk drives use the top side while others, like the Apple Disk II, use the underside. How can the disk manufacturer know which side to have as the "good" side if this is true?

I have about 200 disks -- all of them have been recorded on both sides. In three years I have not had any problems develop because both sides were recorded. Most of my disks are Verbatim, but I also have Georgia Magnetics, Memorex, 3M, Apple, and one or two other brands, and non-brands. The only disks that I have had any real problems with are the new Verbatim Data-life. Three or four of the last 20 did not want to INITIALize with the 16-sector format. I kept getting an I/O ERROR when the process was almost complete. This was on the "good" side, too. The way I get around this problem is to try again with 13-sector, and if that does not work, I try the DAKIN5 disk copy program or the 13-sector LAWRENCE HALL disk copy program. One of these methods has always worked -- so far. Once INITIALized, however, the disks have not given any more trouble --so far.

Instead of cutting notches in all the disk covers to allow the second side to be written to, I have disabled the microswitch sensor in one of my three drives. All that was necessary to do this was:

1. remove the drive's case,
2. loosen the two mounting screws holding the switch,
3. Swing the switch down so it will never be activated,
4. retighten the screws and
5. replace the drive's cover.

It all took about ten minutes. I am reluctant to go along with the nay-sayers from a disk manufacturer even though they may have some good points. If everyone used both sides of every disk they would sell only half as many disks.

I have both Locksmith and Copy II Plus. Locksmith has been giving me good results, albeit very slowly. Copy II Plus has given almost no results -- quickly. I am waiting for my copy of the updated version and hope that it will work better.

The Disk-0-Doc program from M.D. Software has all the features of a nibble copier except it is manually operated. They have an ad in Softalk. So far I have not had any luck using the program to copy anything from one disk to another. I am sure that the instructions are clear to the person who wrote them, but they seem vague to me. It would be a big help if there were some step-by-step examples.

continued on page 55

```
820 VTAB TVRAY% - 1: HTAB THRAY% - 1:
PRINT " ";: VTAB TVRAY: HTAB
THRAY: PRINT " ";: VTAB TVRAY:
HTAB THRAY: PRINT " ";: NEXT
BOOM
830 FOR A = 1 TO 10:ZZ = PEEK (NOISE):
VTAB TVRAY%: HTAB THRAY%: PRINT
"#": FOR B = 1 TO 5: NEXT B: HTAB
THRAY%: PRINT " ";: NEXT A
840 TVRAY% = 0: THRAY% = 0: IVRAY% =
0: IHRAY% = 0: RETURN
```

Bomb Hits Bunker

```
850 FOR T = 1 TO 6: VTAB IVRAY% - 1:
HTAB IHRAY%: INVERSE : PRINT " ";:
VTAB IVRAY%: HTAB IHRAY%: PRINT
"%": GOSUB 1530
860 VTAB IVRAY% - 1: HTAB IHRAY%:
NORMAL : PRINT " ";: VTAB IVRAY%:
HTAB IHRAY%: PRINT " ";: GOSUB
1530: NEXT
880 IHRAY% = 0
890 RETURN
```

Invader's Drop Their Bombs

```
-----<< Pick a random invader >>-----
900 RZ = RND (1) * NZ: RANZ = RZ *
4 + 3: IF LEN (INVADER$(1)) < 5
THEN VTAB VINVADER%: HTAB 1: CALL
- 868: VTAB VINVADER% + 1: CALL
- 868: RETURN
905 IF RANZ > LEN (INVADER$(1)) THEN
RETURN
910 IF RANZ < 0 THEN RANZ = - RANZ
915 TEMP$ = MID$ (INVADER$(1),RANZ +
1, 1)
920 TEMP% = VAL (TEMP$): IF TEMP% =
" " THEN 900
930 IF TEMP% > 9 THEN TEMP% = TEMP% /
10: GOTO 930
935 IF 55% THEN IF TEMP% < = POWER%
THEN TEMP% = 1

-----<< Decrement Invader >>-----
940 TEMP% = TEMP% - 1: TEMP$ = STR$
(TEMP%): IF TEMP% < 1 THEN TEMP$
= " "
945 IF 55% THEN 960
```

continued on page 40


```

----- ( Hit Nothing ) -----
55  XZ = THRAYS + 1:YZ = TVRAYS - 1:
    GOSUB 10: IF XKEYZ = 160 THEN
    RETURN

```

```

----- ( Hit Bunkers ) -----
60  IF XKEYZ < 127 THEN FOR A = 1 TO
    5: GOSUB 30: GOSUB 1530: GOSUB 35:
    GOSUB 1530: GOSUB 30: GOSUB 1530:
    NEXT A: TVRAYS = TVRAYS - 1: GOSUB
    30: TVRAYS = 0: RETURN

```

```

----- ( Hit A Bomb ) -----
65  IF XKZ = 171 THEN 800

```

```

----- ( Can Missile Hit Invader? ) -----
70  IF TVZ < > VIZ + 1 THEN RETURN
72  55Z = 1

```

```

----- ( Hit Invader on Left Side ) -----
75  IF XKEYZ = 221 THEN GOSUB 600:
    RANZ = XZ - HINVADERZ: GOTO 910

```

```

----- ( Hit Invader on Right Side ) -----
80  IF XKZ = 219 THEN GOSUB 620: RANZ
    = XZ - HINVADERZ - 2: GOTO 910

```

```

----- ( Hit Invader in Center ) -----
85  GOSUB 610: RANZ = XZ - HINVADERZ -
    1: GOTO 910

```

=====

Missile Hits Top of Screen

=====

```

90  VTAB TVRAYS: HTAB THRAYS: PRINT
    " " : VTAB TVRAYS - 1: HTAB THRAYS:
    PRINT "$" : FOR A = 1 TO 7: BUZZ =
    PEEK (NOISE): NEXT A: HTAB THRAYS:
    PRINT ".": TVRAYS = 0: RETURN

```

=====

Invaders Destroyed

=====

----- Hit on right side -----

```

600  FOR A = 0 TO 2: VTAB VINVADERZ
    + 1: HTAB XZ - 1: GOSUB 630: VTAB
    VINVADERZ: HTAB XZ - 1: GOSUB
    630: HTAB XZ: GOSUB 630: HTAB XZ
    + 1: GOSUB 630: VTAB VINVADERZ
    + 1: HTAB XZ + 1: GOSUB 630: HTAB
    XZ: GOSUB 630: NEXT
605  BPTSZ = 0: GOSUB 1600: GOTO 640

```

----- Hit on left side -----

```

610  FOR A = 0 TO 2: VTAB VINVADERZ:
    HTAB XZ - 1: GOSUB 630: HTAB XZ
    - 2: GOSUB 630: HTAB XZ: GOSUB
    630: VTAB VINVADERZ + 1: HTAB XZ
    - 2: GOSUB 630: HTAB XZ: GOSUB
    630: HTAB XZ - 1: GOSUB 630: NEXT
615  BPTSZ = 10: GOSUB 1600: GOTO 640

```

----- Hit snack in the middle -----

```

620  FOR A = 0 TO 2: HTAB VINVADERZ
    + 1: HTAB XZ - 1: GOSUB 630: VTAB
    VINVADERZ: HTAB XZ - 1: GOSUB 630:
    HTAB XZ - 2: GOSUB 630: HTAB XZ
    - 3: GOSUB 630: VTAB VINVADERZ
    + 1: HTAB XZ - 3: GOSUB 630: HTAB
    XZ - 2: GOSUB 630: NEXT
625  BPTSZ = 0: GOSUB 1600: GOTO 640

```

----- Dissolve invaders -----

```

630  PRINT ZAP$(A) : X = PEEK (NOISE)
    + PEEK (NOISE): RETURN
640  THRAYS = 0: TVRAYS = 0: GOTO 970

```

=====

Increment Score

=====

```

1600 PTZ = BPZ * SKILLZ + POWERZ + PTZ
1610 VTAB 19: HTAB 30: INVERSE: PRINT
    " " : HTAB 30: PRINT PTSZ :
    NORMAL : RETURN

```

----- Other Lines -----

```

110  IF DISPLAYZ AND FSHOOTZ < 0 THEN
    KEYZ = 160
150  IF KEYZ = 160 THEN AUX = 1: GOSUB
    40
1050 IF AUX THEN GOSUB 50

```

Oh, yes...improvements:

I would want a few more rows of invaders, dropping a few more bombs. And I sure would like to shoot a whole screenful of missiles!!!

I also used an applesoft compiler on it to make it really zip along... And it did... too quickly. If you use the compiler on Text Invaders, I firmly suggest you slow down the invader's descent... I did. I made them randomly choose to move, or stand still, but still wiggling those skinny legs of theirs... that way you don't quite know which way they're going to go...



CHRS (4)

UTILITY NIBBLER



by
**charles
 r.
 haight**

Certain tools are required to help you understand DOS and to allow you to manipulate your disk files. The most important is the disk editing utility which you acquired in Volume 1, Number 1. In this column you will acquire another of these tools and learn how to use it. This program is called: DiskView...

DiskView is a mini "nibbler". A nibbler is a program that reads the raw nibbilized data from a disk without regard to disk format.

This means that you can view data on a nonstandard format disk [copy-protected] as easily as from a normal DOS formatted disk.

With DiskView you can examine a nonstandard disk to see what was changed. Often these changes are minor and a similar change can be made to your DOS. This would allow you to use DiskEdit to read that disk.

The format of DiskView is similar to DiskEdit. A full screen of HEX bytes is displayed with the status prompts on the bottom of your screen. The buffer extends from \$2000 to \$4000 Hex which is large enough to insure reading in an entire track. The slot, drive and track are selectable. Half-tracks can be accessed by appending a ".5" to the track number. The commands are:

- D** change the drive
- L** read the last track (steps by half tracks)
- N** read the next track (steps by half tracks)
- P** print screen contents
- R** read the current track
- S** change the slot
- T** select a track or half track
- X** exit to basic
- increment buffer
- decrement buffer

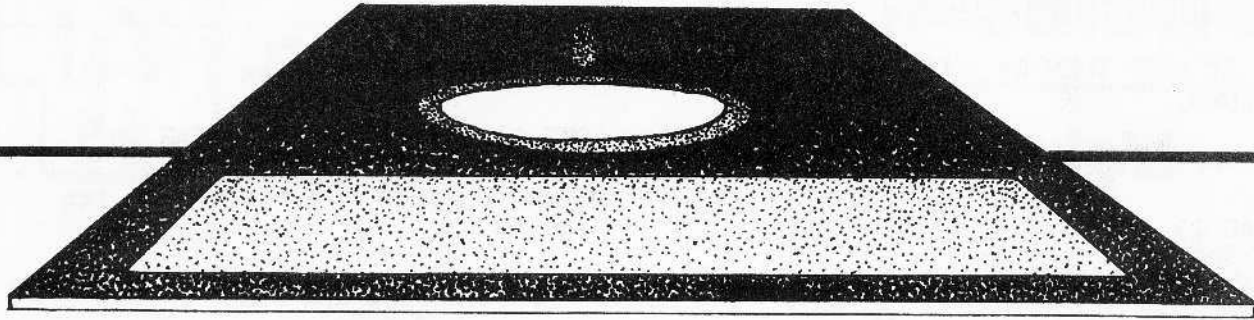
Type in the program and save it to disk. Be especially careful with the data statements. When those values are poked into memory they become a machine language subroutine that is the heart of the program. Run the program. When you see the flashing "COMMAND" prompt, press the "R" key. The screen will fill with hex bytes. You are looking at the actual data stored on the disk. Read Disklocks for a better understanding of this nibbilized data.

List of variables

BS CHRS (8) Backspace
GS CHRS (7) Bell

The remaining variables are locations in the machine subroutine.

S1% 774 Poke slot times 16
S2% 805 Poke slot times 16
DR% 789 Poke drive number
TR% 789 Poke track times two
MV% 830 Call to print to screen
CT% 856 Page position in buffer
IO% 768 Call to read track
NM% 890 Poke number to print
HX% 889 Call to print hex
DR 1 Default drive value



DISKVIEW 1.0

CAUTION: This program is DOS dependent. It calls directly into DOS in order to step the drive motor. You must have DOS 3.3 and 48K of memory. This program can be used to read 13 or 16 sector disks or any other Apple disk but it will only run under a 48K Apple 3.3 DOS.

```
***** Disconnect DOS - GOTO Menu *****
10 TEXT : HOME : IN# 0 : PR# 0 : LOMEM
   : 16384 : POKE 1144,90 : GOTO 90
```

```
***** Get key and return *****
```

```
20 KY% = PEEK ( - 16384) : IF KY% <
   128 THEN 20
30 POKE - 16368,0 : RETURN
```

```
***** Print 40 dashes *****
```

```
40 FOR X = 1 TO 40 : PRINT "-"; : NEXT
   : RETURN
```

```
***** Recalibrate drive *****
```

```
50 GOSUB 60 : POKE 781,0 : POKE 1144,
   90 : POKE TR%,0 : CALL IO% : POKE
   781,255 : POKE TR%,TK% : CALL IO% :
   RETURN
```

```
***** Print prompts and values *****
```

```
60 VTAB 23 : HTAB 2 : INVERSE : PRINT
   "SLOT"; : HTAB 10 : PRINT "DRIVE";
   : HTAB 19 : PRINT "TRACK"; : NORMAL
```

```
70 VTAB 23 : HTAB 7 : PRINT PEEK (S1%)
   / 16; : HTAB 16 : PRINT PEEK (DR%)
   - PEEK (S1%); : HTAB 25 : PRINT
   " " "B$B$B$B$ PEEK (TR%) / 2
```

```
80 RETURN
```

```
***** Menu - normal entry at 100 *****
```

```
90 GOSUB 540
```

```
100 IN% = PEEK (CT%) : VTAB 21 : HTAB
   32 : PRINT "PAGE "IN% - 31 : GOSUB
   60 : VTAB 23 : HTAB 30 : CALL
   - 868 : FLASH : PRINT ">COMMAND<"
   : NORMAL : GOSUB 20
```

```
110 IF KY% = 210 THEN GOSUB 480
120 IF KY% = 211 THEN GOSUB 390
130 IF KY% = 216 THEN GOSUB 410
140 IF KY% = 212 THEN GOSUB 420
150 IF KY% = 199 THEN GOSUB 270
160 IF KY% = 196 THEN GOSUB 230
170 IF KY% = 208 THEN GOSUB 290
180 IF KY% = 136 THEN GOSUB 250
190 IF KY% = 149 THEN GOSUB 370
200 IF KY% = 204 THEN GOSUB 490
210 IF KY% = 206 THEN GOSUB 510
220 GOTO 100
```

```
***** Change drive number *****
```

```
230 VTAB 23 : HTAB 30 : INVERSE :
   PRINT G$"SET DRIVE"; : HTAB 10 :
   FLASH : PRINT "DRIVE"; : NORMAL :
   HTAB 16 : PRINT " " CHR$ (8); :
   GET A$ : DR = VAL (A$) : IF DR < 1
   OR DR > 2 THEN 230
240 POKE DR% , PEEK (S1%) + DR : GOTO
   50
```

```
***** Decrement buffer *****
```

```
250 IN% = IN% - 1 : IF IN% < 32 THEN
   IN% = 32
260 POKE CT%,IN% : CALL MV% : RETURN
```

```
***** Toggle sound on and off *****
```

```
270 PRINT G$ : IF G$ = CHR$ (7) THEN
   G$ = "" : RETURN
```

```

280 IF G$ = "" THEN G$ = CHR$ (7) : ***** Read current track *****
    RETURN

***** Print screen contents *****
290 VTAB 23 : HTAB 30 : FLASH : PRINT
    G$>PRINTER<; : NORMAL
300 PR# 1
310 BUFFER% = PEEK (CT%) * 256
320 PRINT : PRINT "TRACK "TK%
330 FOR X = 0 TO 255 STEP 13 : FOR Y =
    0 TO 12 : POKE NM%, PEEK (BUFFER%
    + X + Y) : CALL HX% : PRINT " " ; :
    NEXT Y : PRINT
340 IF PEEK ( - 16384) = 155 THEN 360
350 NEXT X
360 PR# 0 : POKE - 16368,0 : RETURN

***** Increment buffer *****
370 IN% = IN% + 1 : IF IN% > 63 THEN
    IN% = 63
380 POKE CT%,IN% : CALL MV% : RETURN

***** Change slot number *****
390 VTAB 23 : HTAB 30 : INVERSE :
    PRINT G$"NEW SLOT?"; : HTAB 2 :
    FLASH : PRINT "SLOT"; : NORMAL :
    HTAB 7 : PRINT " " CHR$ (8); : GET
    A$ : KY% = VAL (A$) : IF KY% < 1
    OR KY% > 7 THEN 390
400 POKE S1%,KY% * 16 : POKE S2%,KY% *
    16 : GOTO 240

***** Exit to BASIC *****
410 TEXT : HOME : CALL 1002 : END

***** Change track value *****
420 C$ = "" : VTAB 23 : HTAB 30 :
    INVERSE : PRINT "SET TRACK"; :
    HTAB 19 : FLASH : PRINT G$"TRACK";
    : NORMAL : PRINT " " CHR$ (8)
    CHR$ (8) CHR$ (8); : GET A$ : C$ =
    C$ + A$ : PRINT A$; : GET A$ : C$
    = C$ + A$ : IF A$ = CHR$ (13) THEN
    460
430 PRINT A$;
440 GET A$ : C$ = C$ + A$ : PRINT A$;
450 IF A$ = "." THEN GET A$ : C$ = C$
    + A$ : PRINT A$;
460 KY = VAL (C$) : IF KY < 0 OR KY >
    35 THEN 420
470 TK% = KY * 2

***** Subtract .5 and read *****
480 POKE CT%,32 : VTAB 23 : HTAB 30 :
    FLASH : PRINT ">>>READ<<<"G$; :
    NORMAL : PRINT " " ; : POKE TR%,TK%
    : GOSUB 70 : CALL IO% : GOTO 60

***** Add .5 to track and read *****
490 TK% = TK% - 1 : IF TK% < 0 THEN
    TK% = 71
500 GOTO 480

***** Poke machine subroutine *****
***** define variables & return *****
540 FOR X = 768 TO 894 : READ X% :
    POKE X,X% : NEXT X
550 DATA 162,97,189,137,192,162,96,189
    ,137,192,160,5,169,255,32,168,252,
    136,16,248,169,0,32,160,185,189,
    142,192,169,0,133,30,169,32,133,31
    ,162,96,160,0,189,140,192,16,251,
560 DATA 145,30,230,30,208,245,230,31,
    165,31,201,64,144,237 189,136,192,
    169,1,133,37,32,34,252,169,0,133,
    36,133,30,169,13,133,31,162,1,32,
    74,249,166,30,189,0,32,32,218
570 DATA 253,162,1,32,74,249,230,30,
    240,7,198,31,208,235,76,75,3,32,
    156,252,230,37,32,34,252,169,22,
    133,34,96,169,172,32,218,253,96
580 S1% = 774 : S2% = 805 : DR% = 769
    : TR% = 789 : MV% = 830 : CT% =
    856 : B$ = CHR$ (8) : G$ = CHR$
    (7) : IO% = 768 : NM% = 890 : HX%
    = 889 : DR = 1
590 GOSUB 40 : VTAB 8 : HTAB 10 :
    PRINT "COPYRIGHT 1981 (C)" : PRINT
    : HTAB 10 : PRINT
    "ALL RIGHTS RESERVED" :PRINT: HTAB
    10 : PRINT "HARDCORE COMPUTING" :
    PRINT : HTAB 10 : PRINT
    "P.O. BOX 44549" : PRINT : HTAB 10
    : PRINT "TACOMA, WA 98444"
600 VTAB 22 : GOSUB 40 : GOTO 60

```



hardware solutions

by charles r. haight

Many of you with the A-II+ complained that you couldn't use Bobby's SOFTKEYs because you didn't have access to the Integer Firmware card. Well, here's one solution....

curing those auto-start ROM blues

The autostart ROM in the Apple II+ is a blessing. The autostart feature allows programmers to create a Turnkey System. The user need only insert a program disk and switch on the computer. The monitor ROM will automatically cause the disk to BOOT and the program will be up and running. The reset switch can be locked out. An unknowledgeable user cannot accidentally crash the program. This is great for computer users who have no need or desire to learn about computers.

But, for the hobbyist or businessman who is trying to modify lines or fix a bug in a program, the autostart ROM will make your life miserable. It is all but impossible to stop a running program.

figure 1

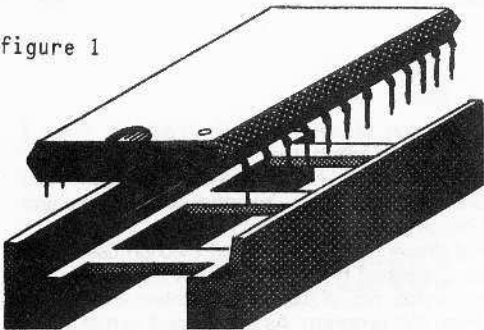
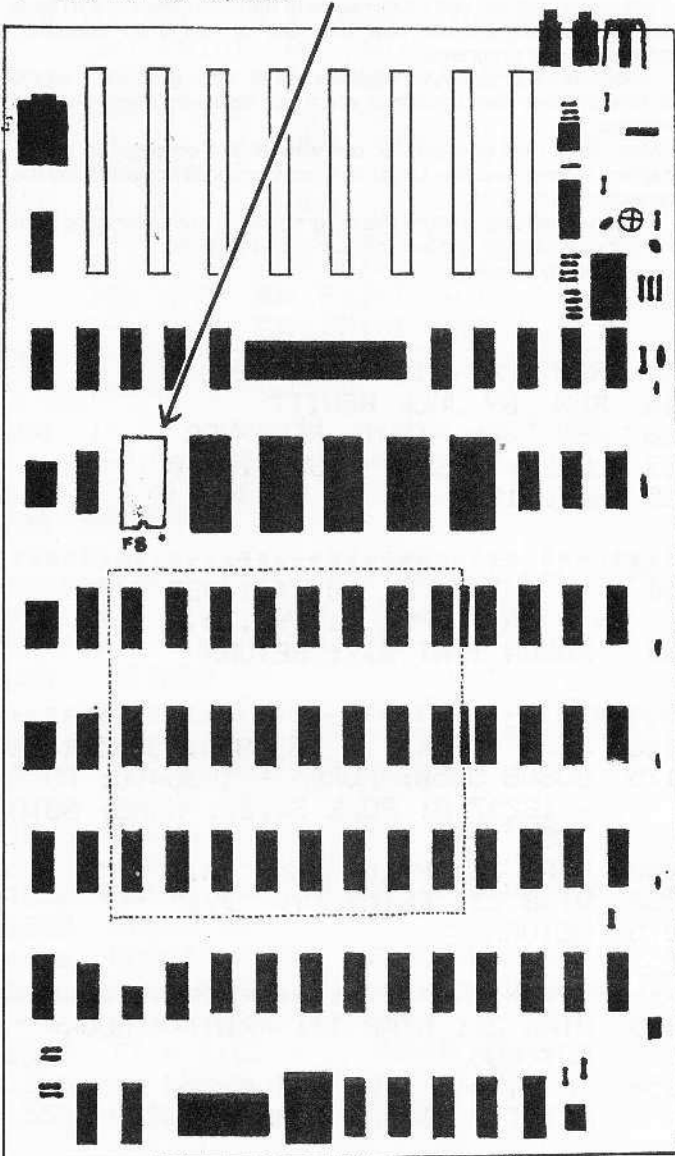


figure 2

↓ Your Apple Motherboard showing the the ROM to replace.



One solution is to not purchase software that cannot be modified.

Another solution is to purchase an Integer Firmware Card. (The old F-8 monitor ROM doesn't have the autostart feature.) The price for this can range from \$100 to \$200, depending on whether you purchase it new or used.

A less expensive solution is to purchase just the F-8 monitor ROM for an Apple II from your Apple dealer and replace the autostart ROM in your Apple II+ whenever you need to do program modifications.

The procedure is simple. However, care should be exercised because the pins on the Integrated Circuit (IC), chips are easily bent. If you decide to do this then read all the instructions before you start.

CAUTION: This will probably void your warranty if your dealer finds out.

1. Turn off the power to your computer. Remove the cord to be sure.
2. Open the top cover and set it aside.
3. Touch the metal power supply case to discharge any static from your body. (The power supply is the large box on the left side.)
4. Locate the F-8 ROM (see fig. 2). Using a small flat screwdriver (the kind your wife gets with her sewing machine) carefully pry up one end of the IC about 1/16th of an inch.

CAUTION: Be sure you are prying up on the IC chip and not on the socket. (see fig. 1)

5. Gently pry up the other end of the IC about 1/16th of an inch.
6. Repeat steps 4 and 5 until the IC is free.
7. Set the chip aside in a safe place. (If the F-8 ROM that you bought came with a case, then use that.)

REMEMBER: Static is your worst enemy. Handle the chip as little as possible.

8. Pick up the chip that you purchased and examine it. One end will have a notch. The chip must be inserted with the notched end pointing toward the keyboard.

CAUTION: Inserting the chip in the socket backwards will fry it and may damage other components on the motherboard.

9. Insert the chip in the socket. You can pre-align the pins on the chip by pressing them upon a flat surface. Be gentle and use even pressure. Insure that the chip is fully seated on its socket.

10. Replace the top cover and reconnect the power cord.
11. Turn on your computer. Your Apple II+ will now emulate an Apple II cold start. When you see the asterisk prompt, type:

```
6 ctrl P return
```

It will BOOT a disk in slot 6.

Follow these same steps when you wish to reinstall the autostart ROM.



HGR : huey's hi-res cor

new
cor



by jack hewitt

This is the beginning of a column which will be devoted to the description and use of the high-resolution graphics capabilities of the Apple computer.

When I first began to look into the world of home computers a couple of years ago, one of the first things which struck me as extremely interesting was the graphics capabilities of this Apple computer. I was fascinated by the demonstration of Applevision with the little man dancing a jig to the tune of "Turkey in the Straw."

However I soon learned, after unpacking my 16K Integer Apple II that high-resolution graphics was not only beyond the capability of my meager knowledge of computer programming but, for the present, was also beyond the capability of my computer's configuration. So as soon as possible I purchased the Programers Aid 1 which gave me high-resolution graphics capabilities.

Here began a love/hate relationship with my computer which has lasted until the present... A love for what this fantastic machine can do and a hate for the realization of how much I did not know and how much I would have to learn in order to accomplish the goals which I had set for myself in respect to my ability to manipulate this computer's graphics capabilities to my ends.

I still am a long way from my goals mainly due to various distractions which seem to crop up along the way, [like sunny summer weather which makes sitting at a keyboard seem somewhat out of place]. But I have managed to learn a few tricks along the way which might possibly be of use to those many people who, like myself, are fascinated by Apple's graphics.

This first installment will introduce you to the accompanying program which I call "artist's Easel". It is a fun drawing program which allows you to draw lines, boxes, solid rectangles, circles, and more.

This program is a compilation of several of the routines which I developed through my first trials in graphics use and began to incorporate into one drawing utility. Soon I had developed what seemed to be a useful set of routines and linked them together with a menu and disk access and error handling.

The main control of drawing is accomplished through use of the game paddles to move a 'cursor' to any position on the screen and from there select lines, circles, etc. A little experimenting will help you to get the full benefit from this program. As I have tried to make it as foolproof as possible, feel free to try anything.

By using the 'T' key you can see the available commands, and the 'F' key will allow you to see the full screen [without the coordinates] or switch back to mixed text and graphics.

In future columns I will explain some of the subroutines which I have used in this program and how they can be tied most effectively together in your programs.

Also, as we progress through the use of hi-res graphics, I will try to develop other useful routines which can be incorporated into this program.

Next issue I will give a quick overview of the organization of the graphics screens and how to get to them from both Applesoft and the monitor.

In the meantime, explore this program and try to understand each subroutine's function and expand your hi-res horizons.

—happy drawing.

Move cursor with paddles.
To end or save picture, type Z.

COMMANDS FOR ARTIST'S EASEL

A position (A,B)
B box from A,B (upper left) to X,Y (lower right).
C change color.
D draw points with keys.
E erase (or CTRL E)
F full/mixed screen switch
I inverse screen.
L draw line from X,Y to A,B.
P draw points with paddles.
R draw circle with center at cursor
S shade from A,B to X,Y (see B).
T switch back to draw mode.
W white (or CTRL W).
Z end/set color to background at cursor.
CTRL E see E
CTRL L draw line but A,B unchanged.
CTRL R draw circle below cursor.
CTRL W see W

```
15 REM COPYRIGHT FEB 1980
16 REM BY JACK HEWITT
20 REM ALL RIGHTS RESERVED
23 LOMEM: 16384: GOSUB 10000
25 GOTO 100
```

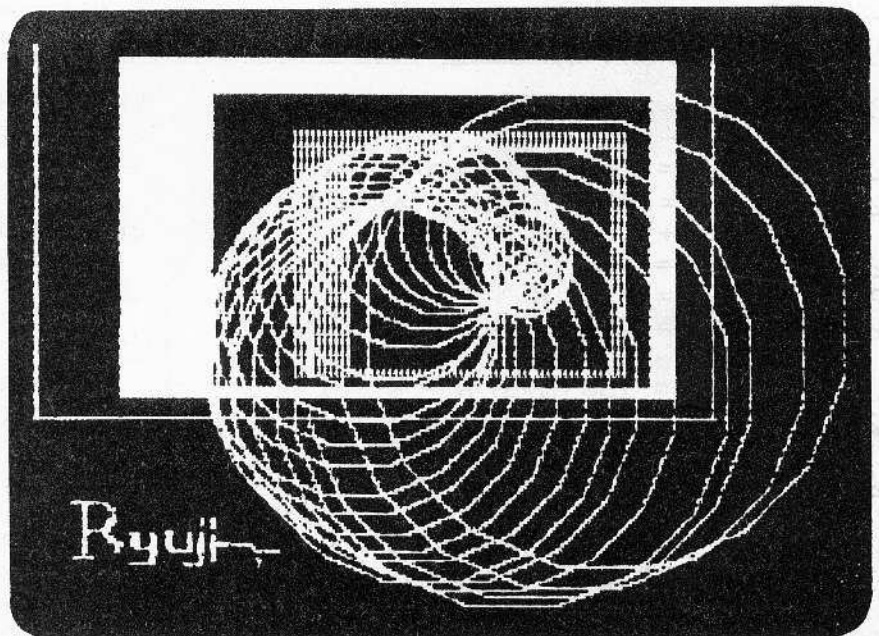
```
50 X = INT ( PDL ( 0 ) * 1.095 ) :
   Y = INT ( PDL ( 1 ) * .75 )
60 XDRAW 1 AT X,Y: RETURN
```

```
100 TEXT : POKE - 16300,0: GOTO 1000
175 GOSUB 5850: POKE - 16304,0: POKE
   - 16297,0: POKE 34,21: HOME: GOTO
   1350
200 UTAB 22: PRINT "X= ";X;" "
250 UTAB 23: PRINT "Y= ";Y;" "
275 RETURN
```

```
300 UTAB 22: HTAB 11: PRINT "COLOR= ";
   CO$(CO)
350 IF TT = 1 THEN UTAB 23: HTAB 9:
   PRINT " T-FOR COMMANDS "
```


] artist's easel

by
jack
hewitt



```

400 IF DD < > 1 THEN VTAB 22: HTAB
30: PRINT "A= ";A;" "
450 IF DD < > 1 THEN VTAB 23: HTAB
30: PRINT "B= ";B;" "
500 IF CS = 1 THEN VTAB 21: PRINT
RIGHT$(S$,37)
550 IF DD = 1 THEN VTAB 21: PRINT
RIGHT$(S$,20);"USE KEYS-": VTAB
21: HTAB 30: PRINT "QWE": VTAB
22: HTAB 30: PRINT "A D " : VTAB
23: HTAB 30: PRINT "ZXC "
600 IF DD = 1 THEN VTAB 23: HTAB 11:
PRINT "P-RETURN "
650 IF PP = 1 THEN VTAB 23: HTAB 11:
PRINT "ANY KEY ESCAPES"
675 KFLAG = 0
700 RETURN

```

```

*****
750 X = INT ( PDL (0) * 1.095):
Y = INT ( PDL (1) * .75)
800 RETURN

```

```

*****
1000 ONERR GOTO 9250
1040 GOSUB 7500
1050 GOSUB 7000: GOSUB 5850: GOSUB
1600
1250 GOSUB 1610: POKE 34,21: HOME
1350 GOSUB 50
1400 FOR T = 1 TO 10: NEXT : GOSUB 60
1450 K = PEEK ( - 16384): IF K > 127
THEN GOSUB 2850
1500 TT = 1:CS = 1: GOSUB 200
1525 IF KFLAG THEN GOSUB 300
1550 GOSUB 50: GOTO 1400

```

```

*****
Init & Variables
*****

```

```

1600 HGR :CO = 3: HCOLOR= CO: RETURN
1610 DIM CO$(7): CO$(0) = "BLACK ":
CO$(1) = "GREEN ": CO$(2) =
"VIOLET": CO$(3) = "WHITE ":
CO$(4) = CO$(0): CO$(5) =
"RED ":CO$(6) = "BLUE "
1620 CO$(7) = CO$(3)

```

```

*****
Init Exis screen
*****

```

```

1650 CO = 3: TS = 1: FS = 0: HCOLOR
= 3: POKE 216,0: ER = 1
1675 POKE - 16301,0
1700 POKE 232,0: POKE 233,3
1750 POKE 768,1: POKE 769,0: POKE
770,4: POKE 771,0: POKE 772,197:
POKE 773,6: POKE 774,0
1800 ROT= 24: SCALE= 8
1850 RETURN

```

```

*****
Draw Points With Keys
*****

```

```

2050 DD = 1: TT = 0: CS = 0:
GOSUB 200
2075 IF KFLAG THEN GOSUB 300
2100 DD = 0
2150 KB = PEEK ( - 16384)
2200 POKE - 16368,0

```

```

2250 IF KB = 209 THEN X = X - 1:
      Y = Y - 1
2300 IF KB = 193 THEN X = X - 1
2350 IF KB = 218 THEN X = X - 1:
      Y = Y + 1
2400 IF KB = 216 THEN Y = Y + 1
2450 IF KB = 215 THEN Y = Y - 1
2500 IF KB = 197 THEN X = X + 1:
      Y = Y - 1
2550 IF KB = 196 THEN X = X + 1
2600 IF KB = 195 THEN X = X + 1:
      Y = Y + 1
2650 IF KB = 208 THEN KFLAG = 1:
      RETURN
2660 IF Y > 191 THEN Y = 191
2665 IF Y < 0 THEN Y = 0
2675 IF X < 0 THEN X = 0
2677 IF X > 279 THEN X = 279
2700 HPLOT X,Y
2750 IF KB = 198 THEN POKE - 16302 +
      FS,0: FS = ABS (FS - 1)
2800 GOTO 2050

```

```

*****
Interpret - execute keyboard commands
*****

```

```

2850 POKE - 16368,0: IF K = 193 THEN
      A = X: B = Y
2875 KFLAG = 1
2900 IF K = 140 THEN HPLOT A,B TO X,Y
2925 IF K = 204 THEN HPLOT A,B TO
      X,Y: A = X: B = Y
2950 IF K = 208 THEN GOSUB 4650
2975 IF K = 133 THEN HCOLOR= 4:
      CO = 4
3000 IF K = 197 THEN HCOLOR= 0:
      CO = 0
3050 IF K = 201 THEN CALL 778
3100 IF K = 215 THEN HCOLOR= 3:
      CO = 3
3125 IF K = 151 THEN HCOLOR= 7:
      CO = 7
3150 IF K = 196 THEN GOSUB 2050
3200 IF K = 198 THEN POKE - 16302 +
      FS,0: FS = ABS (FS - 1)
3250 IF K = 212 THEN POKE - 16304 +
      TS,0: TS = ABS (TS - 1)
3300 IF K = 194 THEN HPLT A,B TO
      X,B TO X,Y TO A,Y TO A,B
3350 IF K = 210 THEN R1 = 1: GOSUB 3600
3360 IF K = 146 THEN R1 = 0: GOSUB 3600
3400 IF K = 211 THEN GOSUB 4500
3450 IF K = 195 THEN GOSUB 4950
3500 IF K = 218 THEN GOTO 4100
3550 RETURN

```

```

*****
'R' - Draw Circle With Inputed Scale
*****

```

```

3600 UTAB 21: INPUT "DRAW CIRCLE-WHAT
      SCALE ";SC
3625 IF SC < 1 OR SC > 35 THEN PRINT
      S$: GOTO 3600
3630 IF X - INT (SC / 2) < 0 OR Y -
      INT (2.5 * SC) < 0 THEN UTAB 21:
      PRINT "OFF SCREEN TRY AGAIN "
      : FOR D = 1 TO 1000: NEXT:
      RETURN
3650 SCALE= SC: IF R1 THEN 3675
3665 HPLOT X - INT (SC / 2),Y: GOTO
      3700
3675 HPLOT X - INT (SC / 2),Y - INT
      (2.5 * SC)
3700 POKE 772,5: POKE 773,0
3750 FOR R = 0 TO 64 STEP 4: ROT = R:
      DRAW 1: NEXT
3850 POKE 772,197: POKE 773,6: SCALE =
      8: ROT = 24: RETURN

```

```

*****
'Z' - Save Screen? or End?
*****

```

```

4100 POKE - 16301,0: HOME : UTAB 21:
      PRINT "SAVE THIS SCREEN (Y/N) ";
4150 INPUT A$: IF LEFT$ (A$,1) = "Y"
      THEN 4230
4200 GOTO 5500
4230 GOSUB 5600: REM GET DRIVE#
4250 IF E1 THEN PRINT "SAVE AS ";B$:
      INPUT A$: IF LEFT$ (A$,1) = "Y"
      THEN N$ = B$: GOTO 4300
4280 INPUT "WHAT FILE NAME? ";N$
4300 POKE - 16302,0
4350 D$ = CHR$ (4)
4400 PRINT D$,"BSAVE";N$,"",A$2000,
      L$2000,0";OR
4450 GOTO 5500

```

```

*****
'S' - Shade Rectangles
*****

```

```

4500 FOR BG = A TO X
4550 HPLT BG,B TO BG,Y: NEXT
4600 RETURN

```

```

*****
'P' - Draw Points With Paddles
*****

```

```

4650 GOSUB 750
4700 TT = 0: PP = 1: GOSUB 200

```

```

4725 IF KFLAG THEN GOSUB 300
4750 H PLOT X,Y
4800 PP = 0
4850 IF PEEK ( - 16384 ) > 127 THEN
RETURN
4900 GOTO 4650

```

```

*****
'C' - Change Color
*****

```

```

4950 POKE - 16301,0
5000 UTAB 21: INPUT "COLOR ?
(0-E 1-G 2-V 3-W 5-0 6-B)";CO
5050 IF CO < 0 OR CO > 7 THEN 4950
5100 HCOLOR= CO: RETURN

```

```

*****
End Or Continue Routine
*****

```

```

5500 HOME : POKE - 16301,0: UTAB 21:
INPUT "DO YOU REALLY WISH TO
END ? ";A$
5510 IF LEFT$(A$,1) < > "Y" THEN
GOTO 1350
5520 POKE - 16302,0: POKE 34,0: HOME:
UTAB 10: PRINT "GOTO 175 TO
REENTER WITH SCREEN INTACT": END
5600 IF D2 THEN HTAB 6: PRINT
"DRIVE # ";: INPUT DR$
5605 IF DR$ = "" THEN DR = 1: RETURN
5610 DR = VAL (DR$): IF DR > 2 THEN
5600
5620 RETURN

```

```

*****
Commands
*****

```

```

5850 HOME: INVERSE
5900 PRINT "***** ARTIST'S EASLE *****
**"
5950 NORMAL
6000 PRINT " *** BY JACK HEWITT ***"
6050 PRINT: FLASH
6100 PRINT " MOVE CURSOR WITH PADDLES"
6125 PRINT: NORMAL
6150 PRINT "W =WHITE (OR-CTRL W )"
6200 PRINT "E =ERASE (OR-CTRL E )"
6250 PRINT "C =CHANGE COLOR"
6300 PRINT "F =SWITCH FULL/MIXED
SCREEN"
6325 INVERSE
6350 PRINT "T =SWITCH BACK TO DRAW"
6375 NORMAL
6450 PRINT "D =DRAW POINTS WITH KEYS"

```

continued on page 50



WANTED
alive
&
running
!

original
PROGRAMS

HardCore Computing would like to publish your original programs both in the magazine and on diskette for our readers. If you have such a program and would like to share it with the readers of HardCore, please ...

1. GOTO page 59 and
2. READ the section labeled:
3. REM PROGRAMS

Do it for the notoriety.
Do it for the \$\$\$
Do it now!

```

6500 PRINT "P =DRAW POINTS WITH
      PADDLES"
6525 PRINT "I =INVERSE SCREEN"
6550 PRINT "A =POSITION (A,B)"
6600 PRINT "L =DRAW LINE FROM 'X,Y' TO
      'A,B'"
6650 PRINT "B =BOX FROM A,B(UL) TO
      X,Y(LR)"
6700 PRINT "S =SHADE FROM A,B TO X,Y
      (SEE BOX)"
6750 PRINT "R =CIRCLE WITH CENTER AT
      CURSOR"
6760 PRINT "CTRL R =CIRCLE BELOW
      CURSOR"
6780 PRINT "CTRL L =LINE BUT A,B NOT
      CHANGED"
6800 PRINT "Z =END-SET COLOR TO BKGND
      AT CURSOR"
6825 FLASH
6850 PRINT : PRINT "*** HIT ANY KEY TO
      BEGIN ***"
6875 NORMAL
6900 IF PEEK ( - 16384) < 128 THEN
6900: POKE - 16368,0: HOME
6950 RETURN

*****
      Load Existing Screen?
*****

7000 HOME : UTAB 10
7050 PRINT "DO YOU WISH TO LOAD AN
      EXISTING"
7100 PRINT "SCREEN OR START A NEW
      ONE "
7150 PRINT "(N =NEW E =EXISTING) ";
7200 INPUT A$
7250 IF LEFT$(A$,1) < > "E" THEN
      RETURN
7275 E1 = 1: D2 = 1: GOSUB 5600
7300 PRINT "WHAT FILE NAME?": PRINT
      "(C-CATALOG)": INPUT " ";B$
7350 D$ = CHR$(4)
7375 IF B$ = "C" THEN PRINT D$;
      "CATALOG": GOTO 7300
7400 PRINT D$;"BLOAD";B$;"A8192,D";OR
7450 POP: GOSUB 5850: POKE - 16304,0:
      POKE - 16297,0: POKE - 16300,0:
      POKE 230,32: GOTO 1250

*****
      Introduction
*****

7500 HOME: ER = 0: SPEED= 150
7550 HTAB 10: PRINT "WELCOME TO:"
7600 PRINT "*** ARTIST'S EASLE ***":
      SPEED= 255

7650 PRINT : PRINT
7700 PRINT "WITH THIS PROGRAM YOU
      WILL BE": PRINT "ABLE TO CREATE
      PICTURES ON THE": PRINT "HIGH-
      RESOLUTION SCREEN."
7750 PRINT " BY MOVING A 'CURSOR'
      WITH THE": PRINT "GAME PADDLES
      YOU CAN PLOT POINTS,": PRINT
      "DRAW LINES, BOXES, OR SOLID"
7800 PRINT "RECTANGLES IN ALL OF
      APPLE'S": PRINT "HIRES COLORS."
7850 PRINT "THE POSITION KEY (A) WILL"
7860 PRINT "ALLOW YOU TO PLOT THE END-
      POINT"
7870 PRINT "OF A LINE OR THE UPPER
      LEFT AND"
7880 PRINT "LOWER-RIGHT CORNERS OF A
      BOX OR"
7890 PRINT "A SOLID. IN ADDITION
      YOU CAN": PRINT "DRAW CIRCLES OR
      DOODLE IN POINTS."
7900 PRINT "THEN YOU CAN SAVE THE
      SCREEN TO": PRINT "DISK
      INSTANTLY."
7950 PRINT: FLASH
8000 PRINT "HIT ANY KEY TO CONTINUE"
8010 NORMAL
8050 IF PEEK ( - 16384) < 128 THEN
8050
8100 POKE - 16368,0: RETURN

*****
      Error Handling Routine
*****

9250 IF ER = 0 THEN PRINT " ***
      ***** NOT FOUND *****";G$:
      FOR D = 1 TO 500: NEXT
9270 GOTO 1050
9300 IF ER = 1 THEN PRINT "ERROR";
      G$;: FOR D = 1 TO 1000: NEXT:
      UTAB 22: PRINT RIGHT$(S$,38):
      GOTO 1400

*****
10000 FOR PK = 778 TO 806: READ ID:
      POKE PK, ID: NEXT
10010 DATA 169, 32, 133, 1, 169, 0,
      133, 0, 160, 0, 177, 0, 73,
      127, 145, 0, 230, 0, 208, 246,
      230, 1
10020 DATA 165,1,201,64,208,238,96
10030 FOR A = 1 to 40: S$ = S$ + " ";
      NEXT a
10040 G$= CHR$(7)
10090 RETURN

```



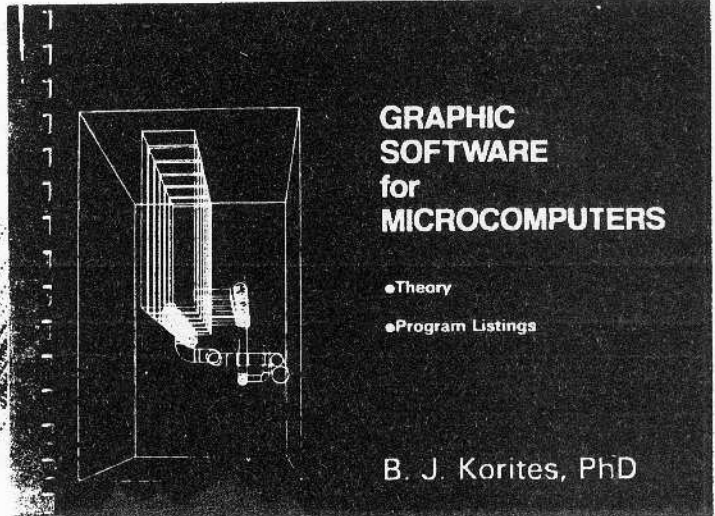
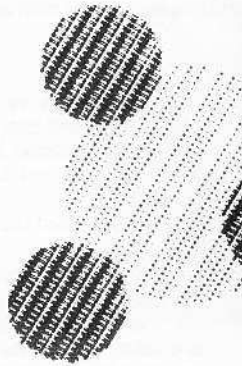
Graphic Software for Microcomputers
 by B. J. Korites, PhD
 ©1981 by Kern Publications

11" wide, 8" high, plastic spiral binding, soft-cover, 200 pages (printed on one side only)

A better title would have been "HI-RES GRAPHICS FROM 1 TO 3-DIMENSIONS" because the author takes the reader from the simplest HPLots to the creation of 3-D images that you can stretch, squeeze, move, turn and even shade, remove the "hidden" lines and clip off lines that HPLOT beyond the screen limits.

Written as a self-teaching guide for microcomputerists in general, this book should be of special interest to Apple-ites because all the programs are in AppleSoft BASIC (for 48K All+). The programs are in the most elementary format so that they can be translated more easily into other languages, the programs do not use most of the Apple's hi-res inherent subroutines such as: shape tables, DRAW, XDRAW, ROT, SCALE, nor does the author discuss hi-res coloring.

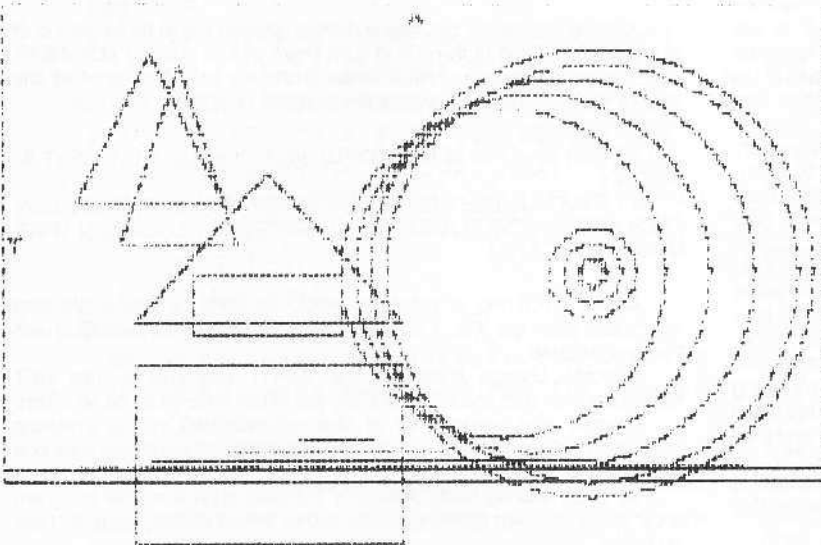
The emphasis, therefore, is not on the actual program, but rather on the algorithm: the logic and math behind such techniques as perspective and 3-D rotations.



Contents:

1. Basic plotting commands
2. Point drawings
3. Line drawings
4. 2D interactive graphics
5. 2D translation
6. 2D scaling and stretching
7. 2D clipping
8. 2D rotation

9. 3D rotation
10. 3D translation and rotation
11. Perspective
12. Hidden line removal
13. Shading
14. 3D shapes
15. Matrix concatenations
16. Graphic tablets
17. Applications
18. Practice problems



Using the 2D interactive graphics program.

The text comes with a floppy diskette of all the AppleSoft programs listed and explained in the book. (Available on another diskette are the same programs in machine language.) The diskette comes with normal DOS 3.3 and is filled to the brim with 61 little programs.

Because the programs are written at such an elementary level, they are not error-safe and I've had to rerun some of the programs because of various annoying error messages but as a vehicle for learning and as motivation to get you to write better software, the programs are great little teachers.

Although it appears to be somewhat hastily put together (I noticed such little items as a SYNTAX ERROR message listed along with the program), the meat is there for the graphics student to devour. The author, B.J. Korites is the president of Data Dynamics, Inc., and teaches a computer graphics course at Worcester Polytechnic Institute in Massachusetts. His book has been adopted as text by several universities.

HARD CORE BOOK REVIEW POLICY

The HardCore staff will review (for publication under this column heading) some of the books sent in for review. All such books (and their accompanying materials) become the property of the HardCore Library/Inventory for later use in comparative reviews, or use as reference material. Send book samples to BOOK REVIEW, HardCore Computing, 14404 East D Street, Tacoma, WA 98445.

my first "real" program:



There are many ways to write articles that instruct readers in the basics of BASIC. One can simply write an article revealing the functions of certain commands or give helpful programming hints. But I've found that the most helpful articles were those with actual programs. However, those programs were usually too complex for me to understand without a great deal of effort and time.

This column presents simple programs written by beginners (the authors might not be in that category right now, but way-back-then...when it was written...they were just beginners). The program is generally bug-free and runs well...but is ripe for improvement...These programs fall into the realm of this column: My First 'Real' Program.

If you have such a program, please submit it for publication along with an article describing the problems you had writing it and how you would write it now.

To get this new column off the ground, I've taken a look at my own "first 'real' full length, debugged program"...and I'm still a little surprised. It was written almost a year ago, so I'm surprised that it works as well as it does. After all, I wrote it when I was in the earliest stages of learning BASIC. Of course, that fact clearly shows through. Just look at the listing: One statement per line...

[I've renumbered it because the original line numbers would have made it difficult for you to make alterations. And I've gotten rid of nearly all of the CTRL G's scattered invisibly throughout the PRINT statements. Other than that, the listing is exactly as I wrote it a year ago.]

Like all "first programs", it could use a lot of improvement...still, it's a colorful lo-res two-player game that works well. And it was my "first"...and that makes it special. It meant that I knew how to use the Apple! And it meant that I had something to show for all the hours of tapping keys, all the silent curses at the author(s) of the tutorial, and all the equally silent praises when I did something new and different, something that wasn't "in the book". Sure, I played with the bouncing "ball" routine, even altered it to see what it would do. But somehow, I felt that all the variations were still products derived directly from "the book".

It began when my little niece came over and watched in fascination as the colors angled across the screen of the color T.V. that was normally noisily displaying her favorite cartoons: bugs bunny, the roadrunner. I let her tap the keys. She was entranced. Being only 5 at the time, she was easily entranced, but her attention span was measured in breaths. Yet the computer held her attention. It combined two of her most favorite "toys": the electric typewriter [she filled pages and pages with her name: A-M-B-E-R] and a color T.V. set!

I made a few simple programs that would interest her. However, because the alphabet was still something she was struggling to learn, most of the programs just prompted yawns after a while. Even the games on the master disk were too difficult for her [little brick out, etc].

Finally, she cornered me and asked if I wanted to play tic-tac-toe. She had just discovered the simple logic embodied in that child's game and was excitedly trying to show everyone...and usually that "everyone" was me. Being a writer, and working mostly at home, meant that your relatives had a tendency to use you as a convenient day-care center. Amber and I were using up an awful lot of my typing bond [she likes to use clean sheets for each game]. And since I had resolved to let the computer do away with all my paperwork...and that included tic-tac-toe...the next step was clear.

I went about using what little I knew to write a computer version of it. It's my guess that most of you have written your own version of this game in either text or lo-res.

Between the constant interruptions of Amber's supervisory questions, I completed it to her delight. I made it noisy. It beeped and buzzed at every turn. It let her type her name. It responded with a "hello", but only to her name. To play, she needed to know only the numbers from "1" to "9". And when she won [and she nearly always won when she played me] it noisily buzzed away. She loved it: the color, the noise, and the fact that she hardly ever lost...

At that time, I had just introduced myself, via the tutorial, to lo-res graphics. I had not even heard of hi-res yet. Yet the hardest thing to write was the part that looked to see if anyone had won yet, and that had to be written first. As you see, the solution was simple.

Why didn't I write one that would actually play against Amber, computer v.s. human? After all, if I wanted her to win most of the time, all I had to do was use the RND function to make the computer's

choice...Perhaps, now that she's older, she'll sit down and play the game by herself but, at the time, she played anything that someone else played. Human interaction was important. She would not play a machine, she had to play a person. Perhaps she had something there...

Well, I called the game: Amber's T's..

The program appears rather long but that is due to the enormous number of graphics involved. All the numbers as well as the "X" and "O" had to be HLINed, VLINed, and PLOTted. When these portions are taken into account, the program is quite reasonable, especially if you do a little compacting and eliminate extraneous line numbers.

Personally, I'd write the program much differently today. Having learned to use the hi-res page, I think that if I were to write that tic-tac-toe game today, I'd have it drawn using a cube of 27 blocks, a cube that could be rotated in 3-d [and I just might!]

Some of the changes that I'd make in this lo-res version would simply make it more compact.

For example: lines 2630 through 2710 can be written in just one line:

```
2630 ON S GOSUB 2500, 2360, 2200, 2070, 1910, 1750,
1610, 1440, 1270: RETURN
```

A more extensive, but space-saving, change would be to get rid of all the HLINing and VLINing and turn them all into a single FOR-NEXT loop that uses integer arrays contains the x,y coordinates of all the PLOTs that are used to create the number or letters. Example:

```
10 FOR A = 1 TO NUMBER%(0): PLOT X%(A),Y%(A): NEXT A:
RETURN
11 REM NUMBER%(0) HOLD THE NUMBER OF ELEMENTS IN
EACH ARRAY, X%( ) AND Y%( ) ARE THE X,Y LOCATION THAT
MUST BE PLOTTED
```

All the PLOTting of numbers would be done by this single loop subroutine after the X%, Y%, and Number% arrays had been filled with the proper data.

I would change most of the INPUT statements into GET statements so that the RETURN key would not have to be hit so often.

I would also remove all of the redundancies in the graphics routines. And, of course, I would add in a little intelligence so that one could play against the computer.

All in all, looking back, Amber's T's isn't such a simple program after all. In fact, I'm starting to feel rather proud of this, my first 'real' program.

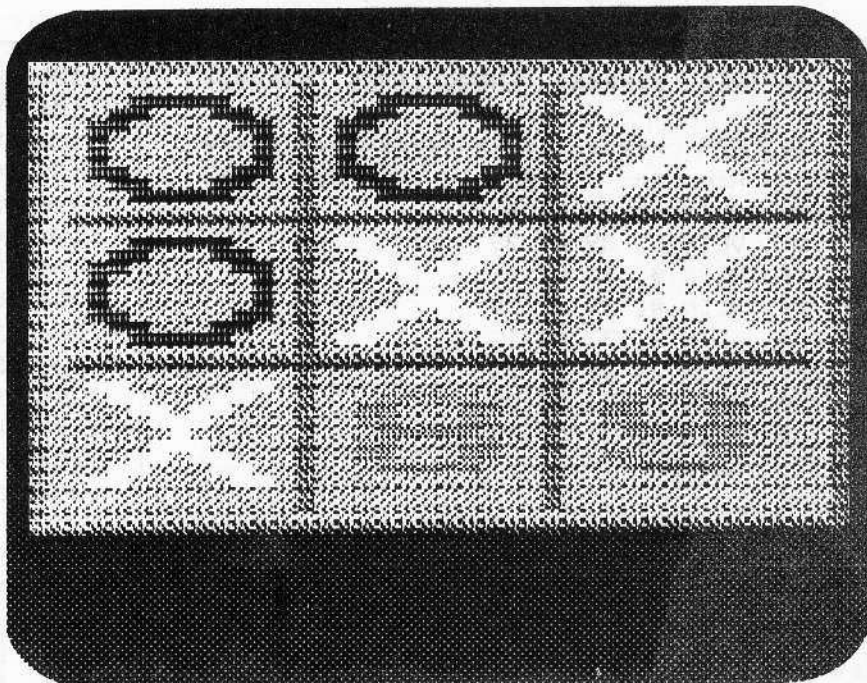
```
10 REM ** AMBER'S TIC TAC TOE **
20 REM ** CREATED DECEMBER 3, 1980 **
30 TEXT : HOME
40 DIM NAME$(75), ANAME$(75)
70 DIM SS(9)
80 DIM LA(3), LB(3), LC(3), LX(3)
90 DIM RA(3), RB(3), RC(3), RX(3)
100 VTAB 18: PRINT "T I C T A C
T O E"
```

AMBER'S

T's

by
Dev. r.

haight



```
110 VTAB 28: PRINT "PRESENTED BY UNCLE
120 VTAB 18: PRINT "PLEASE TYPE YOUR
130 INPUT "WHO IS 'X'"; NAME1$:
140 INPUT "WHO IS 'O'"; ANAME$
150 PRINT "THANK-YOU"
160 A = 7: B = 19: C = 31
170 GR
180 IF ( LEFT$ (NAME1$,5) = "AMBER")
190 OR ( LEFT$ (ANAME$,5) = "AMBER")
200 THEN GOSUB 4390
210 FOR YELLOW = 0 TO 39: COLOR= 13:
220 HLIN 0,39 AT YELLOW: NEXT YELLOW
230 REM **** ZERO DIM. VARIABLES ****
240 FOR TEST = 0 TO 9: SS(TEST) = 0:
250 NEXT TEST
260 GOSUB 4200
270 TIC = 0
280 REM *****
290 REM ***** DRAW TIC TAC TOE *****
300 COLOR= 1
310 GOSUB 680
320 REM *****
330 REM ** NUMBER THE SQUARES **
340 COLOR= 12
350 FOR S = 1 TO 9
360 GOSUB 2620
370 NEXT S: S = 1
380 REM *****
390 VTAB 23: PRINT : PRINT : PRINT
400 Q = 1: RR = 1
410 REM ***** THE NEXT TURN *****
420 IF Q > 0 THEN GOTO 420
430 IF Q < 0 THEN GOTO 390
440 PRINT "IT IS NOW "; ANAME$: "'S
450 TURN TO PLAY."
460 INPUT "ON WHAT NUMBER SQUARE DO
470 YOU WANT YOUR 'O' DRAWN..."; S
480 GOTO 440
490 PRINT "I THINK THAT IT'S ";
500 NAME1$: "'S TURN TO PLAY."
510 INPUT "UPON WHAT NUMBER SQUARE DO
520 YOU WISH ME TO DRAW YOUR 'X', ";
530 S: GOTO 440
540 IF (S < 1) OR (S > 9) THEN GOTO
550 780
560 ST = 1: REM ** is number taken? **
570 IF S = SS(ST) THEN GOTO 860
580 ST = ST + 1: IF ST > 9 THEN GOTO
590 490
600 GOTO 460
610 IF Q > 0 THEN GOTO 530
620 IF Q < 0 THEN GOTO 520
630 STOP
640 SIDE = 2: GOTO 540
650 SIDE = 1: GOTO 540
660 SS(RR) = S: COLOR= 13: GOSUB 2620
670 IF Q > 0 THEN GOTO 590
680 IF Q < 0 THEN GOTO 580
690 REM *****
700 COLOR= 6: GOSUB 880: GOTO 600
710 COLOR= 9: GOSUB 1070: GOTO 600
720 REM *** HAS ANYONE WON YET ****
730 IF RR > 4 THEN GOSUB 3550
740 IF TIC < 0 THEN GOTO 160
750 RR = RR + 1: Q = - Q
760 IF RR = 10 THEN GOSUB 4330:
770 GOTO 160
780 GOTO 360: REM ** next turn **
```

```

660 REM ***** SUBROUTINES *****

670 REM *****

680 REM ** DRAW TIC TAC TOE **
690 VLIN 0, 39 AT 0
700 VLIN 0, 39 AT 39
710 HLIN 0, 39 AT 0
720 HLIN 0, 39 AT 39
730 HLIN 2, 37 AT 13
740 VLIN 2, 37 AT 25
750 HLIN 2, 37 AT 25
760 VLIN 2, 37 AT 13
770 RETURN

780 PRINT "CHOOSE A SINGLE NUMBER FROM
1 TO 9 THAT IS NOT ALREADY TAKEN."
790 FOR PAUSE = 0 TO 1000: NEXT PAUSE
800 IF Q > 0 THEN GOTO 820
810 IF Q < 0 THEN GOTO 840
820 PRINT "IT'S STILL "; NAME1$; "'S
TURN ..."
830 GOTO 430
840 PRINT "..."; ANAME$; ", IT'S STILL
YOUR TURN..."
850 FOR PAUSE = 0 TO 600: NEXT PAUSE:
GOTO 400
860 PRINT "THAT SQUARE IS TAKEN ...":
PRINT "CHOOSE ANOTHER SQUARE..."
870 FOR PAUSE = 0 TO 1000: NEXT PAUSE:
GOTO 360

880 REM ***** MAKE AN "O" *****
890 HLIN X - 2, X + 2 AT Y + 4
900 HLIN X - 2, X + 2 AT Y - 4
910 VLIN Y - 2, Y + 2 AT X - 4
920 VLIN Y - 2, Y + 2 AT X + 4
930 PLOT X - 3, Y - 3
940 PLOT X - 2, Y - 3
950 PLOT X - 3, Y - 2
960 PLOT X + 2, Y + 3
970 PLOT X + 3, Y + 3
980 PLOT X + 3, Y + 2
990 PLOT X - 3, Y + 2
1000 PLOT X - 3, Y + 3
1010 PLOT X - 2, Y + 3
1020 PLOT X + 2, Y - 3
1030 PLOT X + 3, Y - 3
1040 PLOT X + 3, Y - 2
1050 GOSUB 3660
1060 RETURN

1070 REM ***** MAKE AN "X" *****
1080 VLIN Y - 1, Y + 1 AT X
1090 VLIN Y - 2, Y - 1 AT X - 1
1100 VLIN Y - 2, Y - 1 AT X + 1
1110 VLIN Y + 1, Y + 2 AT X - 1
1120 VLIN Y + 1, Y + 2 AT X + 1

```

```

1130 VLIN Y - 3, Y - 2 AT X - 2
1140 VLIN Y - 3, Y - 2 AT X + 2
1150 VLIN Y + 3, Y + 2 AT X - 2
1160 VLIN Y + 3, Y + 2 AT X + 2
1170 VLIN Y - 4, Y - 3 AT X - 3
1180 VLIN Y - 4, Y - 3 AT X + 3
1190 VLIN Y + 4, Y + 3 AT X - 3
1200 VLIN Y + 4, Y + 3 AT X + 3
1210 PLOT X + 4, Y + 4
1220 PLOT X + 4, Y - 4
1230 PLOT X - 4, Y + 4
1240 PLOT X - 4, Y - 4
1250 GOSUB 3660
1260 RETURN

1270 REM ***** MAKE A "9" *****
1280 X = C: Y = C
1290 RX (SIDE) = RX (SIDE) + 1
1300 LC (SIDE) = LC (SIDE) + 1
1310 RC (SIDE) = RC (SIDE) + 1
1320 GOSUB 3660
1330 HLIN X - 2, X + 3 AT Y
1340 HLIN X - 2, X + 2 AT Y - 3
1350 HLIN X - 2, X + 2 AT Y + 3
1360 HLIN X - 3, X - 2 AT Y - 2
1370 HLIN X - 3, X - 2 AT Y - 1
1380 HLIN X - 3, X - 2 AT Y + 2
1390 HLIN X + 2, X + 3 AT Y - 2
1400 HLIN X + 2, X + 3 AT Y - 1
1410 HLIN X + 2, X + 3 AT Y + 1
1420 HLIN X + 2, X + 3 AT Y + 2
1430 RETURN

1440 REM ***** MAKE AN "8" *****
1450 X = B: Y = C
1460 LC (SIDE) = LC (SIDE) + 1
1470 RB (SIDE) = RB (SIDE) + 1
1480 GOSUB 3660
1490 HLIN X - 2, X + 2 AT Y - 3
1500 HLIN X - 2, X + 2 AT Y
1510 HLIN X - 2, X + 2 AT Y + 3
1520 HLIN X - 3, X - 2 AT Y - 2
1530 HLIN X - 3, X - 2 AT Y - 1
1540 HLIN X - 3, X - 2 AT Y + 1
1550 HLIN X - 3, X - 2 AT Y + 2
1560 HLIN X + 2, X + 3 AT Y - 2
1570 HLIN X + 2, X + 3 AT Y - 1
1580 HLIN X + 2, X + 3 AT Y + 1
1590 HLIN X + 2, X + 3 AT Y + 2
1600 RETURN

1610 REM ***** MAKE A "7" *****
1620 X = A: Y = C
1630 LX (SIDE) = LX (SIDE) + 1
1640 LC (SIDE) = LC (SIDE) + 1
1650 RA (SIDE) = RA (SIDE) + 1
1660 GOSUB 3660

```


READER INPUT

continued from page 39

[Editors note.....

Disk-0-Doc is not a nibble copier because it can not distinguish between sync bytes and normal bytes.]

While on the subject of copy programs, I want to say that even though I agree totally with the censorship editorial by Bev. R. Haight, I think that any publication has the right to refuse whatever ads they see fit, and for whatever reasons they may have.

It would be interesting to know why Locksmith cancelled the ad in your magazine.

[Editor's answer: Not having seen the magazine, and after having seen some of our junk-mail flyers, Mr. Alpert felt that our publication was a pirate newsletter that endorsed all forms of software theft. It was a common misconception among software vendors.]

Because we do not agree with someone else is no reason to become unglued. I also think that now that nibble copy programs have become available, the program and disk "protection" problem will die a natural death. Did you know that even Locksmith has been unlocked. The best way to "protect" a program from tampering (but not from copying) is to have it in machine code. How many people can, or will, take one apart to see how it works?

The utility RWTS program, DiskEdit, is really neat and works beautifully. The example of how to use it was a big help in getting started. A printing routine to print out what is on the screen, or the whole track at once for that matter, would be helpful in analyzing all that stuff. Now if it would read and write nibbles...

Please pass on to "Bobby" that in Applesoft, it is pointless to do a check using LEFT\$(X\$,1) after GETting X\$. There is only one character to do something with anyway. See line 5 in the program on page 16 and line 40 in the program on page 29. Line 1682 of the program fragment on page 18 shows how to check the value of a numerick input and line 60210 of the same fragment shows how to check an alphabetic input.

[Editor's note... Bobby says that he suspects a bug in the GET routine. Sometimes, when you get a character, the program GETs a wrong one, or acts as if it got two characters instead of one. It can be solved by clearing the strobe before GETting consecutive characters or by using LEFT\$(X\$,1).]

Sincerely,

John W. Davison

Ft. Walton Beach, FL

Sirs,

Now that I've seen your first issue, I realize how much trouble the "software selling mags" must have given you. You sure tell it like it is. Congratulations.

Some time ago, in a weak moment, I subscribed to

```
1670 HLIN X - 3, X + 3 AT Y - 3
1680 HLIN X + 2, X + 3 AT Y - 2
1690 HLIN X + 1, X + 2 AT Y - 1
1700 HLIN X, X + 1 AT Y
1710 HLIN X - 1, X AT Y + 1
1720 HLIN X - 2, X - 1 AT Y + 2
1730 HLIN X - 2, X - 1 AT Y + 3
1740 RETURN
```

```
1750 REM ***** MAKE A "6" *****
1760 X = C: Y = B
1770 LB (SIDE) = LB (SIDE) + 1
1780 RC (SIDE) = RC (SIDE) + 1
1790 GOSUB 3660
1800 HLIN X - 2, X + 2 AT Y - 3
1810 HLIN X - 3, X + 2 AT Y
1820 HLIN X - 2, X + 2 AT Y + 3
1830 HLIN X - 3, X - 2 AT Y - 2
1840 HLIN X - 3, X - 2 AT Y - 1
1850 HLIN X - 3, X - 2 AT Y + 1
1860 HLIN X - 3, X - 2 AT Y + 2
1870 HLIN X + 2, X + 3 AT Y - 2
1880 HLIN X + 2, X + 3 AT Y + 1
1890 HLIN X + 2, X + 3 AT Y + 2
1900 RETURN
```

```
1910 REM ***** MAKE A "5" *****
1920 X = B: Y = B
1930 RB (SIDE) = RB (SIDE) + 1
1940 RX (SIDE) = RX (SIDE) + 1
1950 LB (SIDE) = LB (SIDE) + 1
1960 LX (SIDE) = LX (SIDE) + 1
1970 GOSUB 3660
1980 HLIN X - 3, X + 2 AT Y - 3
1990 HLIN X - 3, X + 2 AT Y
2000 HLIN X - 2, X + 2 AT Y + 3
2010 HLIN X - 3, X - 2 AT Y - 2
2020 HLIN X - 3, X - 2 AT Y - 1
2030 HLIN X - 3, X - 2 AT Y + 2
2040 HLIN X + 2, X + 3 AT Y + 1
2050 HLIN X + 2, X + 3 AT Y + 2
2060 RETURN
```

```
2070 REM ***** MAKE A "4" *****
2080 X = A: Y = B
2090 LB (SIDE) = LB (SIDE) + 1
2100 RA (SIDE) = RA (SIDE) + 1
2110 GOSUB 3660
2120 VLIN Y - 3, Y + 3 AT X + 2
2130 VLIN Y - 3, Y + 3 AT X + 1
2140 HLIN X - 3, X + 3 AT Y + 2
2150 HLIN X - 3, X - 2 AT Y + 1
2160 HLIN X - 2, X - 1 AT Y
2170 HLIN X - 1, X AT Y - 1
2180 PLOT X, Y - 2
2190 RETURN
```

continued on page 56

continued on page 57

```

2200 REM ***** MAKE A "3" *****
2210 X = C: Y = A
2220 LX (SIDE) = LX (SIDE) + 1

2230 LA (SIDE) = LA (SIDE) + 1
2240 RC (SIDE) = RC (SIDE) + 1
2250 GOSUB 3660
2260 HLIN X - 2, X + 2 AT Y - 3
2270 HLIN X - 2, X + 2 AT Y + 3
2280 HLIN X - 1, X + 2 AT Y
2290 HLIN X - 3, X - 2 AT Y - 2
2300 HLIN X - 3, X - 2 AT Y + 2
2310 HLIN X + 2, X + 3 AT Y - 2
2320 HLIN X + 2, X + 3 AT Y + 1
2330 HLIN X + 2, X + 3 AT Y + 2
2340 HLIN X + 2, X + 3 AT Y - 1
2350 RETURN

2360 REM ***** MAKE A "2" *****
2370 X = B: Y = A
2380 LA (SIDE) = LA (SIDE) + 1
2390 RB (SIDE) = RB (SIDE) + 1
2400 GOSUB 3660
2410 HLIN X - 2, X + 2 AT Y - 3
2420 HLIN X - 3, X + 3 AT Y + 3
2430 HLIN X - 3, X - 2 AT Y - 2
2440 HLIN X + 2, X + 3 AT Y - 2
2450 HLIN X + 2, X + 3 AT Y - 1
2460 HLIN X, X + 2 AT Y
2470 HLIN X - 2, X AT Y + 1
2480 HLIN X - 3, X - 2 AT Y + 2
2490 RETURN

2500 REM ***** MAKE A "1" *****
2510 X = A: Y = A
2520 LA (SIDE) = LA (SIDE) + 1
2530 RA (SIDE) = RA (SIDE) + 1
2540 RX (SIDE) = RX (SIDE) + 1
2550 GOSUB 3660
2560 VLIN Y - 3, Y + 3 AT X
2570 VLIN Y - 3, Y + 3 AT X + 1
2580 HLIN X - 1, X + 2 AT Y + 3
2590 PLOT X - 1, Y - 2
2600 RETURN

2610 REM *****
2620 REM ** DETERMINE SQUARE **
2630 IF S = 1 THEN GOSUB 2500
2640 IF S = 2 THEN GOSUB 2360
2650 IF S = 3 THEN GOSUB 2200
2660 IF S = 4 THEN GOSUB 2070
2670 IF S = 5 THEN GOSUB 1910
2680 IF S = 6 THEN GOSUB 1750
2690 IF S = 7 THEN GOSUB 1610
2700 IF S = 8 THEN GOSUB 1440
2710 IF S = 9 THEN GOSUB 1270
2720 RETURN
    
```

```

2730 REM *****
2740 REM ***** RIGHT SLASH WINS *****
2750 TIC = TIC - 2
2760 GOSUB 3930: REM ** STARS **
2770 FOR WINNER = 0 TO 3
2780 GOSUB 3660
2790 X = A: Y = A: GOSUB 4000
2800 X = B: Y = B: GOSUB 4000
2810 X = C: Y = C: GOSUB 4000
2820 NEXT WINNER
2830 RETURN

2840 REM ***** LEFT SLASH WINS *****
2850 TIC = TIC - 2
2860 GOSUB 3930
2870 FOR WINNER = 0 TO 3
2880 GOSUB 3660
2890 X = C: Y = A: GOSUB 4000
2900 X = B: Y = B: GOSUB 4000
2910 X = A: Y = C: GOSUB 4000
2920 NEXT WINNER
2930 RETURN

2940 REM **** LEFT COLUMN WINS *****
2950 TIC = TIC - 2
2960 GOSUB 3930
2970 FOR WINNER = 0 TO 3
2980 GOSUB 3660
2990 X = A: Y = A: GOSUB 4000
3000 X = A: Y = B: GOSUB 4000
3010 X = A: Y = C: GOSUB 4000
3020 NEXT WINNER
3030 GOSUB 3730
3040 RETURN

3050 REM **** CENTER COLUMN WINS ****
3060 TIC = TIC - 2
3070 GOSUB 3930
3080 FOR WINNER = 0 TO 5
3090 GOSUB 3660
3100 X = B: Y = A: GOSUB 4000
3110 X = B: Y = B: GOSUB 4000
3120 X = B: Y = C: GOSUB 4000
3130 NEXT WINNER
3140 RETURN

3150 REM ***** RIGHT COLUMN WINS *****
3160 TIC = TIC - 2
3170 GOSUB 3930
3180 FOR WINNER = 0 TO 3
3190 GOSUB 3660
3200 X = C: Y = A: GOSUB 4000
3210 X = C: Y = B: GOSUB 4000
3220 X = C: Y = C: GOSUB 4000
3230 NEXT WINNER
3240 RETURN
    
```

READER INPUT

```
3250 REM ***** TOP ROW WINS *****
3260 TIC = TIC - 2
3270 GOSUB 3930
3280 FOR WINNER = 0 TO 3
3290 GOSUB 3660
3300 X = A: Y = A: GOSUB 4000
3310 X = B: Y = A: GOSUB 4000
3320 X = C: Y = A: GOSUB 4000
3330 NEXT WINNER
3340 RETURN

3350 REM ***** MIDDLE ROW WINS *****
3360 TIC = TIC - 2
3370 GOSUB 3930
3380 FOR WINNER = 0 TO 3
3390 GOSUB 3660
3400 X = A: Y = B: GOSUB 4000
3410 X = B: Y = B: GOSUB 4000
3420 X = C: Y = B: GOSUB 4000
3430 NEXT WINNER
3440 RETURN

3450 REM ***** BOTTOM ROW WINS *****
3460 TIC = TIC - 2
3470 GOSUB 3930
3480 FOR WINNER = 0 TO 3
3490 GOSUB 3660
3500 X = A: Y = C: GOSUB 4000
3510 X = B: Y = C: GOSUB 4000
3520 X = C: Y = C: GOSUB 4000
3530 NEXT WINNER
3540 RETURN

3550 REM **** WHO IS THE WINNER? ****
3560 IF RX(SIDE) = 3 THEN GOSUB 2740
3570 IF LX(SIDE) = 3 THEN GOSUB 2840
3580 IF LA(SIDE) = 3 THEN GOSUB 3250
3590 IF LB(SIDE) = 3 THEN GOSUB 3350
3600 IF LC(SIDE) = 3 THEN GOSUB 3450
3610 IF RA(SIDE) = 3 THEN GOSUB 2940
3620 IF RB(SIDE) = 3 THEN GOSUB 3050
3630 IF RC(SIDE) = 3 THEN GOSUB 3150
3640 IF TIC < 0 THEN GOTO 3730

3660 REM ** NOISE MAKER **
3670 P = - 16336
3680 FOR NOISE = 0 TO 10
3690 H = PEEK (P) - PEEK (P) + PEEK
(P) - PEEK (P)
3700 NEXT NOISE
3710 RETURN

3720 REM *****
3730 REM * CONGRATULATIONS TO WINNER *
3740 IF Q > 0 THEN GOTO 3760
3750 IF Q < 0 THEN GOTO 3810
```

continued on page 58

continued from page 55

C.C., and you are right, it's just a software advertisement cover to cover.

Also, I was aware that Dave Alpert was getting a bad time, but I didn't realize it was that bad. I would like to congratulate Allen Emery for having the courage to advertise in *HARDCORE*, as well as the other few advertisers you have. All Apple people should support these advertisers as much as possible...

Thanks,
Allen Chaikin
Fallbrook, CA

Dear Editor,

Received your Premeir issue of *HARDCORE*. Holy Digital! It is explicit and down-right hard-core. It is like watching topless and bottomless for the first time. I have keyed in the Akalabeth and Walla! It works. I love it. I love it!

Keep up your good work. I would like to see in the future an article on "how to" cram all the short commercial games on one diskette.

Brothers, there will be hardship on your part for your underground work. But keep it up.

Thank-you for your kind fore-play.

P.S. Why not include all programs listings of a single issue on one diskette for all those who hate to type and do "digital manipulation"?

sincerely,
(name withheld by request)

Dear HardCore,

Even with the presence of bit-copiers, the need for a magazine like yours is tremendous since the resulting copy disk is still "protected". I find it very annoying that I must first boot the *BASICS* disk to run protected 3.2 disks on DOS 3.3. Any articles that will allow one to transfer files from protected 3.2 disks onto normal DOS 3.3 disks will be greatly appreciated. The article "boot 3.2 Diskettes on 3.3" which appeared in *Nibble* helps, but it doesn't work on some of the protected 3.2 disks.

I purchased *Locksmith* version 3.0 in order to back up my library. It worked on 8 out of 10 protected disks. Overall, I am quite happy with the *Locksmith* although I find that the documentation leaves something to be desired. Nowhere does it state in the documentation that it supports synchronized or unsynchronized tracks. Perhaps you could explain these terms in a future issue so that the users can understand the difference between them.

Which version of *Locksmith* did you test in your review? In any review, it would be helpful to indicate the version number and then have follow-up re-

continued on page 58...

```

3760 REM ***** X WINS *****
3770 HTAB 5: PRINT "CONGRATULATIONS !"
3780 PRINT "IT LOOKS LIKE "; NAME1$;
      " WINS THIS GAME."
3790 FOR PAUSE = 0 TO 1000: NEXT PAUSE
3800 GOTO 3850

3810 REM ***** 0 WINS *****
3820 PRINT "YOU WON,"; ANAME$; "!!";
      PRINT "YOU BEAT "; NAME1$; "!!"
3830 GOTO 3850
3840 FOR PAUSE = 0 TO 1000: NEXT PAUSE
3850 REM ***** ANOTHER GAME ??? *****
3860 PRINT "HOW ABOUT ANOTHER GAME?"
3870 INPUT "IF YOU DO, THEN TYPE 0";
      AGAIN
3880 IF AGAIN = 0 THEN GOTO 160
3890 PRINT "WELL.....SO LONG, THEN."
3900 TEXT : HOME : PRINT "THE END"
3910 END

3920 REM *****
3930 REM ** STARS **
3940 PRINT:PRINT
3950 FOR STAR = 1 TO 40: PRINT "*"::
      NEXT STAR
3980 RETURN

3990 REM *****
4000 REM *** DISPLAY WINNING ROWS **
4010 IF Q > 0 THEN GOTO 4030
4020 IF Q < 0 THEN GOTO 4070

4030 REM ***** FLASH THE 'X' *****
4040 COLOR= 15: GOSUB 1070
4050 COLOR= 0: GOSUB 1070
4060 GOTO 4110

4070 REM ***** FLASH THE 'O' *****
4080 COLOR= 15: GOSUB 880
4090 COLOR= 0: GOSUB 880
4100 GOTO 4110
4110 RETURN

4190 REM *****
4200 REM ** ZERO VARIABLES **
4210 FOR SIDE = 1 TO 2
4220 RX(SIDE) = 0
4230 RA(SIDE) = 0
4240 RB(SIDE) = 0
4250 RC(SIDE) = 0
4260 LX(SIDE) = 0
4270 LA(SIDE) = 0
4280 LB(SIDE) = 0
4290 LC(SIDE) = 0
4300 NEXT SIDE
4310 RETURN

4320 REM *****
4330 REM ***** NO ONE WON *****
4340 PRINT "IT LOOKS LIKE NO ONE WON
      THIS GAME...": PRINT "SO LET'S
      TRY ANOTHER GAME"
4350 FOR PAUSE = 0 TO 3000: NEXT PAUSE
4360 PRINT "AND THIS TIME,"; NAME1$;
      " AND "; ANAME$; ", TRY A LITTLE
      HARDER!"
4370 FOR PAUSE = 0 TO 2000: NEXT PAUSE
4380 RETURN

4390 REM ***** HELLO AMBER *****
4400 PRINT "HELLO AMBER"
4410 RETURN

```

READER INPUT

continued from page 57

views as new versions become available.

In your review of Locksmith, you state that it takes 5 to 6 minutes to copy a disk. With a single drive system, it takes about 30 minutes!! Perhaps you could indicate copy times for users with only a single disk drive.

Best of luck in your "Crusade" and I will be looking forward to trying out some of the methods you suggest and to receiving future issues.

Sincerely yours,
Richard Sylvester
Belgium

Sir:

Your new magazine was shown to me, and I was asked my opinion about the inside back cover. I realize there is the problem of unauthorized copying, but I do not want programs that I cannot back up. And I'm not too enthusiastic about programs that I cannot modify to my own tastes. When I write programs for other people I write in FP Basic, so that they can modify them. I have a couple of programs that I purchased that did not suit my purposes and it is not practical to modify them, so I just do not use them (money wasted.)

Sincerely,
Harry A. Moneyhun
Anderson, IN

ARE YOU A HARDCORE WRITER ?

HARDCORE COMPUTING is looking for hard-core computerists to write columns, articles or programs.

COLUMNS

If you have an idea for a column and would like to become a HARDCORE columnist or feature writer, then drop the Publisher a note. Please include a sample of your proposed column and a letter briefly introducing yourself, your background, your interests and expertise. If you haven't written before, don't let that stop you from dropping us a letter.

PROGRAMS

If you have an original program and would like to share it with other HARDCORE readers then send us a cassette or diskette with the entire debugged program and the accompanying external documentation or an article describing what it is and how it does what it does. We pay from \$20 to \$100 to print the article and listing in HARDCORE.

Some readers would rather buy the tape or diskette instead of typing it all in themselves, so we offer you TWO means of selling your software:

1. Do your own mail-order. The listing and the article will act as advertising, and those interested in purchasing the program can send their bucks to you directly. It's a great way to start your own software business.

2. Join the HARDCORE Program Library. We offer the program on diskette along with any others in the HARDCORE Program Library. We charge our readers for the diskette, postage and handling, and then offer each program at what we call the Direct Royalty Charge (DRC) of \$1 to \$5, depending upon demand. The DRC is what you receive for each program of yours that we sell through the HARDCORE Program Library. You would receive a quarterly royalty check (the amount is announced in the magazine). Our readers would then be able to get many programs on the same diskette. It is less expensive for the readers and less work for the writer.

Our diskettes are not copy-protected, so your programs are bound to be bartered. One advantage of this method is the potential of program improvement through reader input. The improvements are then included in the updated versions. Copy-protection would slow or halt such program evolution.

ARTICLES

What is a HARDCORE article?

It is a no-frills how-to step-by-step trip through some aspect of Apple-dom.

\$

We work closely with our writers and we generally pay from \$20 to \$100 for articles, columns or programs. The bigger bucks go to the authors of in-depth, HARDCORE articles full of apple-aids and useful, well-written programs. The lesser bucks go to writers who submit hints, small programs or articles that have less HARDCORE content.

Why not become a HARDCORE writer?

All it takes is a touch of genius, and a no-nonsense hard-core determination to find out everything you can about your Apple . . . plus an honest willingness to share that hard-won knowledge with others in this Apple Free Press magazine.

LIBRARIAN

exchange subscriptions

There were at least 4 other periodicals that were left out of the last LIBRARIAN list:

Dr. Dobb's Journal

For users of small computer systems

a monthly magazine by People's Computer Company, Box E, 1263 El Camino Real, Menlo Park, CA 94025.

\$21 per year (U.S.), more elsewhere

Peelings II

The magazine of Apple Software Evaluation

6 times a year by Peelings II, Inc., 2260 Oleander, Las Cruces, NM 88001

\$15 per year (U.S.), more elsewhere

Computerist's Directory

The Who's Who of People and Computers

published semi-annually (Jan, July) with updates by Alternet, Inc., P.O. Box 405, Forestville, CA 95436.

\$10 per year (see ad on page 30)

Computer Shopper

The Nationwide Marketplace for Computer Equipment

a monthly by Patch Publishing Company, Inc., P.O. Box F, Titusville, FL 32780.

\$10 per year (U.S., more elsewhere)

If we still left you out of our Library, just send us a recent issue of your periodical and begin an exchange subscription with us. Join our library of reference literature.

Diskedit version 2.1 is listed here for those of you who like to type. The program has undergone some rather extensive changes. The speed has been optimized and this version is compilable. The commands have been expanded and changed. Default values are used extensively. The commands are listed below and are in the following format:

KEY (MNEMONIC) MODE PROMPT

ESC (ESCAPE)--

This is the "I changed my mind" key. Press this key to reset defaults and exit back to the command mode.

RTN (RETURN)--

The RETURN key when used to answer an input prompt, will set the current default and continue. [Example: when prompted for the track and sector during a read command, pressing RETURN twice will cause the current track and sector to be read.]

A (ASCII) COMMAND

Sets the [ASC] screen format and prints the buffer contents on the screen-display as all ASCII characters. Control characters are shown as periods. [See "Z" filter command]

C (CATALOG) CATALOG

Calls the catalog from the disk using the default slot and drive.

D (DRIVE) SET-DRIVE

Prompts you for a new drive number. Valid entries are 1 or 2. [Default drive value at BOOT is 1.]

E (EDIT) EDIT

A continuous-edit mode, this mode allows you to type in changes just like a typewriter. If you are in [ASC] format then all keys are valid except the ESC key. In [HEX] format only valid hex digits are accepted as input. Press the ESC key to exit this mode.

F (FORMAT) COMMAND

Steps the EDIT format from normal to flashing to inverse and back to normal in a continuous cycle each time you press the 'F' key. [Example: in normal mode, press the 'F' key to change mode to flashing. Press the 'F' key again to get inverse and again to go back to normal.] The [ASC] prompt will reflect the format that you're in. It is only visible in [ASC] screen format].

G (BELL) COMMAND

Turns the sound on and off each time you press the "G" key. [Defaults at BOOT is ON]

H (HEX) COMMAND

Writes the buffer contents to the screen as all HEX characters. This is useful for editing binary files.

I (UP) COMMAND

Moves cursor up.

J (LEFT) COMMAND

Moves cursor left.

K (RIGHT) COMMAND

Moves cursor right.

M (DOWN) COMMAND

Moves cursor down.

L (LAST) COMMAND

Reads last sector.

N (NEXT) COMMAND

Reads next sector.

O (CURSOR) CURSOR

Allows cursor to be jumped to any absolute position in the displayed sector.

P (PRINT) PRINTER

Dumps the buffer contents to your printer. The format is 16 HEX bytes and 16 ASCII characters. A header is printed first which shows the track, sector and volume. This routine supports the SilenType printer. If you use another printer you will need to alter this routine. [see Making Changes]

R (READ) COMMAND

Prompts you for Track and Sector to read [The ESC key resets defaults and exits while the RETURN key sets defaults and continues]

S (SLOT) SET-SLOT

Prompts you for new slot. [Default slot at BOOT is 6]

U (UPDATE) COMMAND

Flips input/output mode between hex and decimal updates the display information. Only the track, sector and pointer are affected by this key. [Default at BOOT is Hex]

W (WRITE) WRITE

Prompts you for Track and Sector to write to [The ESC key sets defaults and exits while the RETURN key sets defaults and continues]. After entering the Track and Sector, DiskEdit will beep three times and pause. This is your chance to change your mind. Press RETURN to WRITE or any other key to escape.

X (EXIT) COMMAND

Clears screen and exits to BASIC.

Z (FILTER) COMMAND

Switches the filter on and off each time you press the "Z" key. The filter changes inverse, flashing and lower case characters on the screen display to normal ASCII. Control characters are displayed as inverse. Entering the EDIT mode automatically switches the filter off. [Default at BOOT is on.]

```

2 TEXT : HOME : LOMEM : 8194 : GOSUB
  272 : GOSUB 134 : GOTO 160
4 HTAB (HT% - 1) * 3 + 1 : VTAB VT% +
  2 : RETURN
6 GOSUB 4 : PRINT " " ; : VT% = INT
  (KY% / 13) : HT% = KY% - (VT% * 13)
  + 1 : GOSUB 134 : GOTO 70
8 PRINT MID$ (A$,KY% + 1,1) ; : RETURN
10 GET N$ : KY% = ASC (N$) + 128 :
  GOTO 16
12 GOSUB 4
14 KY% = PEEK (- 16384) : IF KY% < 128
  THEN INVERSE : PRINT ">" CHR$ (8) ;
  : GOSUB 42 : NORMAL : PRINT " "
  CHR$ (8) ; : GOSUB 42 : INVERSE :
  PRINT ">" CHR$ (8) ; : GOSUB 42 :
  NORMAL : PRINT " " CHR$ (8) ; : GOTO
  14
16 POKE -16368,0 : IF KY% < > 155 THEN
  RETURN
18 POKE FL%,FT% : TK% = T0% : SE% =
  S0% : GOSUB 134 : GOSUB 144 : GOSUB
  142 : POKE CM%,RD%

```

*LINS 160
ADD GOSUB 144*

*PRINT
POS*

Print HEX

*Shifting
cursor*

at key

Edit 2.1

```
20 CALL - 10621 : GOTO 160
22 GOSUB 12 : IF NOT FX% THEN 34
24 IF KY% < 160 THEN 34
26 IF KY% < 192 THEN KY% = KY% + 32 :
  GOTO 30
28 KY% = KY% - 32
30 IF FX% = 1 THEN KY% = KY% - 96 :
  GOTO 34
32 KY% = KY% - 160
34 POKE 7936 + PT%,KY% : GOSUB 4 :
  PRINT " "; : POKE NM%,KY% : CALL
  HX% : GOTO 60
36 GOSUB 4 : PRINT "> "; : GOSUB 22 :
  VTAB 23 : HTAB 35 : IF IX% THEN
  PRINT PT% : GOTO 36
38 POKE NM%,PT% : CALL HX% : PRINT " "
  ; : GOTO 36
40 GOSUB 4 : GOSUB 84 : IF KY% > 15
  THEN PRINT G$; : GOTO 40
42 RETURN
44 GOSUB 40 : A1% = KY% : PRINT " "; :
  GOSUB 8 : GOSUB 40 : A2% = KY% :
  GOSUB 100 : GOTO 34
46 GOSUB 4 : PRINT "> "; : GOSUB 44 :
  VTAB 23 : HTAB 35 : IF IX% THEN
  PRINT PT% : GOTO 46
48 POKE NM%,PT% : CALL HX% : PRINT " "
  ; : GOTO 46
50 GOSUB 4 : PRINT " " : KY% = KY% - 8
  : ON KY% GOTO 52,56,60,160,66
52 VT% = VT% - 1 : IF VT% < 0 THEN VT%
  = 19 : GOTO 68
54 GOTO 70
56 HT% = HT% - 1 : IF HT% < 1 THEN HT%
  = 13 : GOTO 52
58 GOTO 70
60 HT% = HT% + 1 : IF HT% > 13 THEN
  HT% = 1 : GOTO 66
62 IF VT% = 19 AND HT% > 9 THEN HT% =
  1 : GOTO 66
64 GOTO 70
66 VT% = VT% + 1 : IF VT% > 19 THEN
  VT% = 0
68 IF VT% = 19 AND HT% > 9 THEN HT%
  = 9
70 GOSUB 4 : PRINT ">" : PT% = VT% *
  13 + HT% - 1 : RETURN
72 PRINT G$G$; : RETURN
74 KY% = PEEK (- 16384) : IF KY% < 128
  THEN 74
76 POKE - 16368,0 : RETURN
```

```
78 ERR = PEEK (ER%) : POKE 34,1 : POKE
  35,21 : HOME : VTAB 12 : HTAB 12 :
  IF ERR = 16 THEN PRINT
  "UNABLE TO WRITE"
80 IF ERR < > 16 THEN PRINT
  "DISK DRIVE ERROR"
82 GOSUB 72 : FOR X = 1 TO 2000 : NEXT
  : POKE 756,0 : POKE 35,24 : GOTO 18
84 IF NS% THEN GOSUB 10 : GOTO 88
86 GOSUB 14
88 IF KY% = 141 THEN RETURN
90 KY% = KY% - 176 : IF KY% < 0 OR KY%
  > 22 THEN GOSUB 72 : GOTO 84
92 IF KY% > 9 THEN KY% = KY% - 7 : IF
  KY% < 10 OR KY% > 15 THEN GOSUB 72
  : GOTO 84
94 IF ED% THEN RETURN
96 IF IX% AND KY% > 9 THEN GOSUB 72 :
  GOTO 84
98 RETURN
100 IF ED% OR NOT IX% THEN KY% = A1% *
  16 + A2% : RETURN
102 KY% = A1% * 10 + A2% : RETURN
104 NS% = 1 : VTAB 23 : HTAB 1 : FLASH
  : PRINT*"TRK"; : NORMAL : HTAB 5 :
  GOSUB 84 : IF KY% > 15 THEN KY% =
  TK% : GOTO 114
106 IF NOT IX% AND KY% > 2 THEN 114
108 IF KY% > 3 THEN 114
110 A1% = KY% : GOSUB 8 : GOSUB 84 :
  IF KY% > 15 THEN KY% = A1% : GOTO
  114
112 A2% = KY% : GOSUB 100
114 IF KY% < 0 OR KY% > 34 THEN PRINT
  G$; : GOTO 104
116 TO% = TK% : TK% = KY% : GOSUB 144
118 NS% = 1 : VTAB 23 : HTAB 1 :
  INVERSE : PRINT "TRK"; : HTAB 9 :
  FLASH : PRINT "SCT"; : NORMAL :
  HTAB 13 : GOSUB 84 : IF KY% > 15
  THEN KY% = SE% : GOTO 128
120 IF NOT IX% THEN 128
122 A1% = KY% : GOSUB 8 : IF KY% > 1
  THEN 128
124 GOSUB 84 : IF KY% > 15 THEN KY% =
  A1% : GOTO 128
126 A2% = KY% : GOSUB 100 : IF KY% < 0
  OR KY% > 15 THEN PRINT G$; : GOTO
  118
128 SO% = SE% : SE% = KY% : VTAB 23 :
  HTAB 9 : INVERSE : PRINT "SCT"; :
  NORMAL : GOSUB 144 : IF PEEK (CM%)
  = 2 THEN VTAB 24 : HTAB 30 : FLASH
  : PRINT "??WRITE??"G$G$G$; :
  NORMAL : GOSUB 74 : IF KY% < > 141
  THEN 18
130 POKE SC%,SE% : POKE TR%,TK% : GOTO
  140
```

```

132 FOR X = 1 TO 40 : PRINT "-"; :
    NEXT : RETURN
134 VTAB 23 : HTAB 1 : CALL - 958 :
    INVERSE : PRINT "TRK"; : HTAB 9 :
    PRINT "SCT"; : HTAB 17 : PRINT
    "VOL"; : HTAB 31 : PRINT "CRS" :
    PRINT "SLOT"; : HTAB 9 : PRINT
    "DRIVE"; : HTAB 18 : PRINT "I/O";
136 VTAB 24 : HTAB 30 : FLASH : PRINT
    "COMMAND"; : NORMAL : PRINT " "; :
    RETURN
138 GOSUB 144
140 CALL IO%
142 CALL MV% : RETURN
144 VTAB 23 : HTAB 5 : IF IX% THEN
    PRINT TK% " "; : HTAB 13 : PRINT
    SE% " "; : HTAB 21 : PRINT PEEK
    (VO%) " "; : HTAB 35 : PRINT
    PT% " ";
146 IF NOT IX% THEN POKE NM%,TK% :
    CALL HX% : HTAB 13 : POKE NM%,SE%
    : CALL HX% : HTAB 21 : PRINT PEEK
    (VO%); : HTAB 35 : POKE NM%,PT% :
    CALL HX% : PRINT " ";
148 HTAB 25 : IF HA% = 1 THEN PRINT
    "(HEX"; : GOTO 156
150 IF FX% = 0 THEN PRINT "(ASC"; :
    GOTO 156
152 IF FX% = 1 THEN PRINT "("; : FLASH
    : PRINT "ASC"; : GOTO 156
154 IF FX% = 2 THEN PRINT "("; :
    INVERSE : PRINT "ASC";
156 NORMAL : PRINT ")"; : VTAB 24 :
    HTAB 6 : PRINT PEEK (SL%) / 16; :
    HTAB 15 : PRINT PEEK (DR%); : HTAB
    22 : IF NOT IX% THEN PRINT "HEX";
    : RETURN
158 PRINT "DEC"; : RETURN
160 POKE 216,0 : ED% = 0 : NS% = 0 :
    IN# 0 : PR# 0 : GOSUB 136 : IF
    PEEK (756) > 0 THEN GOSUB 78
162 TO% = TK% : SO% = SE% : GOSUB 144
    : GOSUB 12 : KY% = KY% - 192 : IF
    KY% > 0 AND KY% < 27 THEN ON KY%
    GOSUB 166,72,264,170,168,176,184,
    180,50,50,50,220,50,226,234,188,72
    ,248,250,72,182,72,256,258,72,270
164 GOTO 160
166 PRINT G$ : HA% = 2 : POKE FM%,HA%
    : GOTO 142
168 VTAB 24 : HTAB 30 : FLASH : PRINT
    G$">>EDIT<<"; : ED% = 1 : NORMAL :
    POKE FL%,0 : CALL MV% : ON HA%
    GOTO 46,36
170 NS% = 1 : VTAB 24 : HTAB 30 :
    INVERSE : PRINT G$"SET DRIVE"; :

```

```

HTAB 9 : FLASH : PRINT "DRIVE"; :
    NORMAL : HTAB 15 : GOSUB 84 : IF
    KY% > 15 THEN GOTO 134
172 IF KY% < 1 OR KY% > 2 THEN 170
174 POKE DR%,KY% : GOTO 134
176 PRINT G$ : FX% = FX% + 1 : IF FX%
    > 2 THEN FX% = 0
178 RETURN
180 PRINT G$ : HA% = 1 : POKE FM%,HA%
    : GOTO 142
182 PRINT G$ : IX% = NOT IX% : RETURN
184 PRINT G$ : IF G$ = CHR$ (7) THEN
    G$ = "" : RETURN
186 IF G$ = "" THEN G$ = CHR$ (7) :
    RETURN
188 VTAB 24 : HTAB 30 : FLASH : PRINT
    G$"PRINTER"; : NORMAL : CALL 1002
    : VTAB 23 : HTAB 39 : PRINT : IF
    NOT PR THEN GOSUB 260
190 IF NOT PR THEN PRINT G$G$G$; :
    RETURN
192 LI% = 1 : Z% = 7936 : PRINT CHR$
    (4)"PR#"PR : PRINT "TRACK : "; :
    IF IX% THEN PRINT TK%;
194 IF NOT IX% THEN POKE NM%,TK% :
    CALL HX%
196 PRINT " SECTOR : "; : IF IX% THEN
    PRINT SE%;
198 IF NOT IX% THEN POKE NM%,SE% :
    CALL HX%
200 PRINT " VOLUME : " PEEK (VO%) :
    FOR X = 1 TO 16 * LI% : CT =
    (X - 1) * 16 / LI% : IF NOT IX%
    THEN POKE NM%,CT : CALL HX%
202 IF IX% THEN PRINT CT;
204 HTAB 5 : PRINT "-"; : HTAB 7 : FOR
    L = 0 TO 15 / LI% : POKE NM%, PEEK
    (Z% + L) : CALL HX% : PRINT " "; :
    NEXT L : PRINT " "; : FOR Y = 1 TO
    16 / LI% : CHAR = PEEK (Z%) : IF
    CHAR < 32 THEN CHAR = CHAR + 192 :
    GOTO 214
206 IF CHAR < 64 THEN CHAR = CHAR +
    128 : GOTO 214
208 IF CHAR < 96 THEN CHAR = CHAR +
    128 : GOTO 214
210 IF CHAR < 128 THEN CHAR = CHAR +
    64 : GOTO 214
212 IF CHAR < 160 THEN CHAR = 174
214 PRINT CHR$ (CHAR); : Z% = Z% + 1 :
    NEXT Y : PRINT : IF PEEK (- 16384)
    = 155 THEN 218
216 NEXT X
218 PRINT CHR$ (4)"PR#0" : RETURN
220 SE% = SE% - 1 : IF SE% > - 1 THEN
    POKE SC%,SE% : GOTO 138

```



```

222  SEX = 15 : POKE SC%,SEX : TK% =
      TK% - 1 : IF TK% >= 0 THEN POKE
      TR%,TK% : GOTO 138
224  TK% = 34 : POKE TR%,TK% : GOTO 138
226  SEX = SEX + 1 : IF SEX < 16 THEN
      POKE SC%,SEX : GOTO 138
228  SEX = 0 : POKE SC%,SEX : TK% = TK%
      + 1 : IF TK% < 35 THEN POKE TR% ,
      TK% : GOTO 138
230  TK% = TK% + 1 : IF TK% < 35 THEN
      POKE TR%,TK% : GOTO 138
232  TK% = 0 : POKE TR%,TK% : GOTO 138
234  NS% = 1 : PRINT G$ : VTAB 24 :
      HTAB 30 : INVERSE : PRINT "CURSOR"
      ; : NORMAL : PRINT " " ; : VTAB 23
      : HTAB 31 : FLASH : PRINT "CRS" ; :
      NORMAL : HTAB 35 : GOSUB 84 : IF
      KY% > 15 THEN GOSUB 134 : GOTO 70
236  A1% = KY% : GOSUB 8 : GOSUB 84 :
      IF KY% > 15 THEN KY% = A1% : GOTO 6
238  A2% = KY% : GOSUB 8 : GOSUB 100 :
      IF NOT IX% THEN 6
240  IF KY% > 25 THEN 6
242  A1% = KY% : GOSUB 84 : IF KY% > 15
      THEN KY% = A1% : GOTO 6
244  A2% = KY% : GOSUB 8 : GOSUB 100 :
      IF KY% < 0 OR KY% > 255 THEN PRINT
      G$ : GOTO 234
246  GOTO 6
248  VTAB 24 : HTAB 30 : INVERSE :
      PRINT G$">READ<"; : NORMAL : PRINT
      " " ; : GOSUB 104 : GOTO 136
250  NS% = 1 : VTAB 24 : HTAB 30 :
      INVERSE : PRINT G$"SET SLOT"; :
      HTAB 1 : FLASH : PRINT "SLOT"; :
      NORMAL : HTAB 6 : GOSUB 84 : IF
      KY% > 15 THEN GOTO 134
252  IF KY% < 1 OR KY% > 7 THEN 250
254  POKE SL%,KY% * 16 : GOTO 134
256  POKE CM%,WR% : VTAB 24 : HTAB 30 :
      INVERSE : PRINT G$"WRITE<"; :
      NORMAL : GOSUB 104 : POKE CM%,RD%
      : GOTO 134
258  TEXT : HOME : CALL 1002 : END
260  FOR XX = 1 TO 7 : AD = - 16384 +
      256 * XX : IF PEEK (AD + 23) = 201
      AND PEEK (AD + 55) = 207 AND PEEK
      (AD + 76) = 234 THEN PR = XX : XX
      = 7
262  NEXT XX : RETURN
264  CALL 1002 : ONERR GOTO 268
266  VTAB 24 : HTAB 30 : FLASH : PRINT
      "CATALOG"; : NORMAL : POKE 34,1 :
      POKE 35,21 : HOME : PRINT : PRINT
      CHR$ (4)"CATALOG,D" PEEK (DR%)" ,S"
      PEEK (SL%) / 16 : PRINT : POKE 35,
      24 : PRINT
      "PRESS ANY KEY TO CONTINUE " ; :
      GOSUB 74 : GOTO 142
268  POKE 216,0 : VTAB 12 : HTAB 12 :
      PRINT "DISK DRIVE ERROR"GGGG$ :
      FOR X = 1 TO 1000 : NEXT : POKE 35
      ,24 : GOSUB 142 : GOTO 20
270  PRINT G$ : FT% = NOT FT% : POKE
      FL%,FT% : GOTO 142
272  FOR X = 752 TO 974 : READ A% :
      POKE X,A% : NEXT : DATA 1,0,0,0,0,
      1,169,3,160,0,32,217,3,176,22,96,
      1,96,1,0,0,0,17,3,0,31,0,0,1,0,0,
      96,1,0,1,239,216,162,255,142,244,
      2,96,173,243,2,76,43,3,173,243,2,
      174,240,2,224,2,240,4
274  DATA 32,218,253,96,32,237,253,96,
      169,1,133,37,32,34,252,169,0,133,
      36,141,242,2,169,13,141,241,2,162,
      1,32,74,249,174,242,2,189,0,31,
      174,240,2,224,2,240,9,32,218,253,
      32,74,249,76,181,3,172,245,2,192,0
276  DATA 240,61,24,216,201,32,176,5,
      105,192,76,175,3,201,64,176,5,105,
      128,76,175,3,201,96,176,5,105,128,
      76,175,3,201,128,176,5,105,64,76,
      175,3,201,160,176,6,56,233,128,76,
      175,3,201,224,176,3,76,175,3,56,
      233
278  DATA 32,76,175,3,201,128,48,6,201,
      160,16,2,169,174,32,237,253,32,74,
      249,238,242,2,240,8,206,241,2,208,
      140,76,65,3,32,156,252,230,37,32,
      34,252,169,22,133,34,96
280  SL% = 769 : DR% = 770 : TR% = 772
      : SC% = 773 : CM% = 780 : ER% =
      781 : VO% = 782 : RD% = 1 : WR% =
      2 : IO% = 758 : MV% = 819 : FM% =
      752 : HA% = 1 : FX% = 0 : TK% = 0
      : SE% = 0 : HT% = 1 : VT% = 0 :
      HX% = 795 : IX% = 0 : A$ =
      "0123456789ABCDEF" : HH% = 801 :
      NM% = 755 : G$ = CHR$ (7) : FL% =
      757 : FT% = 1
282  GOSUB 132 : VTAB 8 : HTAB 10 :
      PRINT "COPYRIGHT 1981 (C)" : PRINT
      : HTAB 10 : PRINT
      "ALL RIGHTS RESERVED" :PRINT: HTAB
      10 : PRINT "HARDCORE COMPUTING" :
      PRINT : HTAB 10 : PRINT
      "P.O. BOX 44549" : PRINT : HTAB 10
      : PRINT "TACOMA, WA 98444" : VTAB
      22 : GOSUB 132: PRINT "INSERT DISK
      - PRESS ANY KEY TO CONTINUE"; :
      GOSUB 74 : GOTO 140

```

last
 cursor
 text
 cursor
 read
 set slot
 write
 catalog



advertising index

Computerist's Directory.....	30
D.R. Jarvis Computing, Inc.	7
Omega Microware, Inc.	15
Sensible Software, Inc.	17
Soft CTRL Systems	31
Stellation Two	28
Sympathetic Software	16

hard



..... programs for sale

If you want the most recent version of a previously published program, or if you simply do not have the time to type in a program's long listing, you may be able to purchase it from the HardCore Program Library.

the honor royalty system

Our diskettes are not copy-'protected' and we encourage you to let other computerists make copies of them. But please ask them to help compensate the authors. Just ask them to participate in our Honor Royalty System.

Those who make copies of these programs are urged to send \$1 for each program copied to the program's author (address in the program listing) especially if they appreciated the program. Write to us for updated versions of those same programs, or watch the pages of HardCore computing for changes and additions.

PROGRAM LIBRARY CATALOG

code	program name	language	documentation	DRC
U-001B	DiskEdit 2.3 (updated)	A-soft*	booklet \$2	\$4
U-002	DiskView 1.0 (new)	A-soft*	booklet \$1	\$3
A-001	Artist's Easel (new)	A-soft*	none yet	\$3
G-001	Amber's T's (new)	A-soft	none	\$2
G-002	Text Invaders 2.0 (new)	A-soft	none	\$2
G-003	Relief Mapper 1.0 (new)	A-soft	none	\$1

Note: If, after you order, an updated version was received by the Library, you will be sent the most recent versions.

HOW TO ORDER HARDCORE PROGRAMS

1. See if it's available by checking the list of programs for sale.
2. List the code and program names along side its actual cost (the DRC or Direct Royalty Charge which the authors receive). Documentation is an extra charge if you desire it.
3. Total the charges and then,
4. Add in the cost of the disk, first class postage, and handling... which now is \$6.00 (due to the rise in postage).
5. Total all of this, and add a note telling us whether you want 13 or 16 sector formats, and send the list along with a check for that total to:

HardCore Program Library
P.O. Box 44549
Tacoma, WA 98444

Be sure to include your address neatly printed



CORE computing

Subscription Rates

USA . . . \$20.00
Canada . . . \$28.50
Mex., Central America . . . \$32.50
S. America . . . \$37.50
All others . . . \$42.00

Dealer Inquiries Invited

Hardcore Computing
Subscriptions
14404 East "D" Street
Tacoma, WA 98445
U.S.A.

lift-off



the most
outspoken and controversial
computer magazine in apple-dom:

HARDCORE computing
P.O. Box 44549
Tacoma, WA 98444
(206) 531-5690



step
far beyond the
CENSORED limits of
passive computerling...



step
into new worlds of
the **LIBERATED** apple.*

you're at the back door.

step
inside!

