

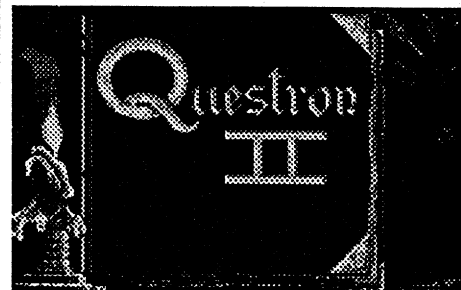
# 87 COMPUTIST

U.S. \$3.75

Canada & Mexico \$7

## Contents

Most Wanted softkeys .....	23	Context Clues: Hidden		Skills Bank II .....	22
RDEX Contributors .....	23	Treasure Red Level .....	21	Snack Attack .....	17
unClassifieds .....	23	Drawing Conclusions: Chief of		Spectre .....	19
The Product Monitor .....	4	Detectives Blue Leve .....	21	Superkey .....	19
Big-Mem Fix for Appleworks 2.0 .....	5	Drawing Conclusions: Chief of		TEAM Series: General Interest .....	20
Budokan: The Martial Spirit .....	5	Detectives Red Level .....	21	TEAM Series: Making Choices .....	20
Build Your Own Stereo Board! .....	5	Estimation Activities (3.5") .....	20	TEAM Series: People and Places .....	20
Happy CRT, The .....	5	Estimation Activities (5.25") .....	20	TEAM Series: Science and Health .....	20
I Want My GS RAM III! .....	4	Fact or Opinion: Smart		Time Navigator Around	
Magic Candle II Support .....	6	Shopper Blue Level .....	21	the World (3.5") .....	20
Missing Persons .....	4	Fact or Opinion: Smart		Time Navigator Around the World	
Platinum Painting Made Easy .....	6	Shopper Red Level .....	21	(5.25") .....	20
Their Finest Hour:		Field Zoologist, The .....	20	Twerps .....	17
The Battle Of Britain .....	5	Following Directions: Behind the		Vacation Nation Travel (3.5") .....	20
Wraith .....	4	Wheel Red Level .....	21	Vacation Nation Travel (5.25") .....	20
<b>Features, Notes and such:</b>		Getting the Main Idea: Around the		Where in the World is	
Basics of Kracking Part #14, The .....	15	World Blue Level .....	21	Carmen Sandiego? .....	18
Build Your Own Stereo Board! .....	5	Getting the Main Idea: Around the		Woolly Bounce (3.5") .....	20
Kracking SSI's RDOS .....	15	World Red Level .....	21	Woolly Bounce (5.25") .....	20
MECC 3.5" disk softkeys .....	20	Grammar Gazette (3.5") .....	20	<b>Bugs:</b>	
MECC 5.25" disk softkeys .....	20	Grammar Gazette (5.25") .....	20	Bug in Algebra vol 3 & 4 .....	21
Notes on Test Drive II cars disks .....	22	Grammar Monsters (3.5") .....	20	Bug in Matchmaker	
Questron II CDA v2.0 .....	7	Inference: School Days Blue Level .....	21	World Geography Facts .....	21
		Inference: School Days Red Level .....	21	Bug in PSAT Word Attack Skills .....	21
		Journey to the Age of the Dinosaurs ....	20	Bug in PSAT/SAT Analogies .....	21
		Littletown Zoo (3.5") .....	20	Bug in SAT Word Attack Skills .....	21
		Littletown Zoo (5.25") .....	20	<b>Advanced Playing Techniques:</b>	
		Living Cell (3.5"), The .....	20	Questron II .....	7
		Living Cell (5.25"), The .....	20	<b>Playing Tips:</b>	
		Outpost .....	17	Questron II .....	7
		Paper Plane Pilot (3.5") .....	20	<b>Questions:</b>	
		Paper Plane Pilot (5.25") .....	20	Ⓢ .....	16, 21
		Pensate .....	18	<b>IBM Softkeys:</b>	
		Problem Solving With Nim (3.5") .....	20	Aardvark McGraw-Hill	
		Problem Solving With Nim (5.25") .....	20	Financial Software .....	22
		Reading for Detail: Race		Ancient Art of War .....	22
		Track Blue Level .....	21	Astrilis .....	22
		Reading for Detail: Race		ATI Training Software .....	22
		Track Red Level .....	21	Faery Tale, The .....	22
		Sequence: What Comes		Gauntlet .....	22
		First Blue Level .....	21	Grave Yardage .....	22
		Sequence: What Comes		Gun Boat .....	22
		First Red Level .....	21		
		Seven Cities of Gold .....	18		



### Softkeys:

Blue Grass Bluff (5.25") .....	20
Bluegrass Bluff (3.5") .....	20
Botanist's Apprentice (The) .....	20
Cause & Effect: Mountain	
Climbing Blue Level .....	21
Cleanwater Detectives (3.5") .....	20
Cleanwater Detectives (5.25") .....	20

**Subscribe to  
COMPUTIST**

**We give you  
More!**

**Only \$25 for 8 issues**

I am:

- Renewing my current subscription
- Changing my address (Please include last label)
- A new subscriber.

Subscription rates:

- \$27 For new subscribers (and late renewals that missed 1 issue. U.S. regular/3rd Class Mail) Includes an extra \$2 for postage and handling. We will send your Starter Kit and 1st issue by 1st Class mail.
- \$25 U.S. regular renewal (3rd Class mail)
- \$35 1st Class (U.S. / Canada / Mexico)
- \$54 All other Foreign (Air mail)
- \$68 COMBO (1st Class plus Disk)
- \$95 Foreign COMBO (Air mail plus Disk)

**VISA**

**Charge It!**

**MC**

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Country \_\_\_\_\_ Phone \_\_\_\_\_

VISA/MC \_\_\_\_\_ Exp. \_\_\_\_\_

Signature \_\_\_\_\_

#87

• Send US funds drawn on US bank. • For regular subscriptions, please allow 4-8 weeks for 1st issue or add \$2 for postage and we will send your 1st issue by 1st Class mail. • Send check/money order to:

COMPUTIST 33821 E Orville Rd Eatonville WA 98328-9590 (206) 832-3055

**COMPUTIST(87)**  
**PO Box 242**  
**Kapowsin WA 98344-0242**

Forwarding postage guaranteed/Address correction requested

**BULK RATE**  
U.S. Postage  
**PAID**  
Kapowsin WA  
Permit No. 6

# COMPUTIST

Charles R. Haight  
Jeff Hurlburt  
Vincent Andrews

Editor  
Reviews  
BBS

COMPUTIST is published by SoftKey Publishing. Address all inquiries to:

COMPUTIST  
33821 East Orville Road  
Eatonville, WA 98328-9590  
(206) 832-3055

• COMPUTIST does NOT purchase editorial material. The entire editorial content consists of information submitted to COMPUTIST for publication in the shared interests of all COMPUTISTS.

• Unsolicited material (manuscripts, letters to the editor, softkeys, A.P.T.s, playing tips, questions, etc.) are assumed to be submitted as letters-to-the-RDEX-editor for publication with all and exclusive rights belonging to COMPUTIST.

• Entire contents copyright 1990 by SoftKey Publishing. All rights reserved. Copying done for other than personal or internal reference (without express written permission from the publisher) is prohibited.

• The SoftKey Publishing assumes no liability or responsibility for the products advertised in this newsletter. Although we are usually pretty much in agreement, any opinions expressed by the authors are not necessarily those of COMPUTIST magazine or SoftKey Publishing.

**SUBSCRIPTIONS:** Rates (for 8 issues):

U.S. ....\$25 Canada/Mexico ..\$35  
U.S. 1st Class ..\$35 Other Foreign .....\$54

• Subscriptions are sold by number of issues and not by month or year. An 8 issue subscription means that you will receive 8 issues before you need to renew. It's when you will receive each issue that we're a little erratic about.

• Domestic Dealer rates: Call (206) 832-3055 for more information.

• Change Of Address: Let the U.S. Postal Service know that you are moving. Tell them that you want your mail forwarded. If your issue does not come to you in an envelope then you have a regular subscription and you must tell the USPS to forward your third class mail. Notify us as soon as you know your new address. When we receive your notice of change of address, we will send you an acknowledgement card. If you do not receive the acknowledgement card after 2 weeks, send another notice or call us direct.

*Issues missed due to non-receipt of change of address may be acquired at the regular back issue rate.*

*We are not responsible for missing issues 90 days after mailing date. If you do not receive an issue at the usual time each month, please call or write.*

Apple® is a trademark of Apple Computers. IBM® is the IBM trademark.

## Software recommendations

The Starter Kit contains most of the programs that you need to "Get started". In addition, we recommend that you acquire the following:

- Applesoft program editor such as "Global Program Line Editor (GPLE)".
- Assembler such as "Merlin/Big Mac".
- Bit-copy program such as "Copy II Plus", "Locksmith" or "Essential Data Duplicator".
- Word-processor (such as AppleWorks).
- "COPYA", "FID" and "MUFFIN" from the DOS 3.3 System Master disk.

## Super IOB and Controllers

This powerful deprotection utility (in the COMPUTIST Starter Kit) and its various Controllers are used in many softkeys. (It is also on each Super IOB Collection disk.)

## Reset into the Monitor

Softkeys occasionally require the user to stop the execution of a copy-protected program and directly enter the Apple's system monitor. Check the following list to see what hardware you will need to obtain this ability.

**Laser 128:** Your ROM includes a forced jump to the monitor. Press **ctrl return** reset.

**Apple II+, //e, compatibles:** 1) Place an Integer BASIC ROM card in one of the Apple slots. 2) Use a non-maskable interrupt (NMI) card such as Replay or Wildcard.

**Apple II+, compatibles:** 1) Install an F8 ROM with a modified reset-vector on the computer's motherboard as detailed in the "Modified ROM's" article (COMPUTIST #6 or Book Of Softkeys III) or the "Dual ROM's" article (COMPUTIST #19).

**Apple //e, //c:** Install a modified CD ROM on the computer's motherboard that changes the open-apple **ctrl** reset vector to point to the monitor. (*This will void an Apple //c warranty since you must open the case to install it.*)

**Apple //gs:** If you have the 2x ROM, there is a hidden Classic Desk Accessory (CDA) that allows you to enter the monitor. In order to install the new CDA, you should enter the monitor (CALL -151) before running any protected programs and press **# return**. This will turn on two hidden CDAs, Memory Peeker and Visit Monitor. Thereafter press **openapple ctrl esc** to go to the Desk Accessories menu. Select Visit Monitor and there you are. Use **ctrl Y** to exit.

## Recommended literature

- Apple II Reference Manual (or IIe, IIc, etc.)
- DOS 3.3 & ProDOS manual
- Beneath Apple DOS & Beneath Apple ProDOS, by Don Worth and Pieter Lechner, from Quality Software

## Typing Applesoft programs

BASIC programs are printed in a format that is designed to minimize errors for readers who key in these programs. If you type:

```
10HOME:REMCLEAR SCREEN
```

The LIST will look like:

```
10 HOME : REM CLEAR SCREEN
```

Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces don't pose a problem except when they are inside of quotes or after a DATA command. There are two types of spaces: those that have to be keyed and those that don't. Spaces that must be typed appear in COMPUTIST as special characters (◊). All other spaces are there for easier reading.

**NOTE:** If you want your checksums to match, only type spaces within quotes or after DATA statements if they are shown as (◊) characters. SAVE the program at periodic intervals using the name given in the article. All characters after a REM are not checked by the checksum program so typing them is optional.

## Typing Hexdumps

Machine language programs are printed in COMPUTIST as hexdumps, sometimes also as source code.

Hexdumps are the shortest and easiest format to type in. You must first enter the monitor:

```
CALL -151
```

Key in the hexdump exactly as it appears in the magazine, ignoring the four-digit checksum (\$ and four digits) at the end of each line. When finished, return to BASIC with:

```
3D0G
```

BSAVE the program with the filename, address and length parameters given in the article.

## Typing Source Code

The source code is printed to help explain a program's operation. To enter it, you need an

"Assembler". Most of the source code in older issues is in S-C Assembler format. If you use a different assembler, you will have to translate portions of the source code into something your assembler will understand.

## Computing checksums

Checksums are 4-digit hexadecimal numbers which tell if you typed a program correctly and help you locate any errors. There are two types of checksums: one created by the CHECKBIN program (for machine language programs) and the other created by the CHECKSOFT program (for BASIC programs). Both are on the "Starter Kit".

If your checksums do not match the published checksums then the line where the first checksum differs is incorrect.

**CHECKSOFT instructions:** Install Checksoft (BRUN CHECKSOFT) then LOAD your program. Press **&** to get the checksums. Correct the program line where the checksums first differ.

**CHECKBIN instructions:** Enter the monitor (CALL -151), install Checkbin at some out of the way place (BRUN CHECKBIN, A\$6000), and then LOAD your program. Get the checksums by typing the Starting address, a period and the Ending address of the file followed by a **ctrl Y**.  
**SSSS.EEEE ctrl Y**

Correct the lines where the checksums differ.

## Writing to the RDEX editor

RDEX (are-decks) stands for: Reader's Data EXchange. We print what you write. When you send in articles, softkeys, APTs, etc., you are submitting them for free publication in this magazine. RDEX does not purchase submissions nor do we verify data submitted by readers. If you discover any errors, please let us know so that we may inform our other readers.

Remember that your letters or parts of them may be used in RDEX even if not addressed to the RDEX editor. Correspondence that gets published may be edited for clarity, grammar and space requirements.

Because of the great number of letters we receive and the ephemeral and unpredictable appearance of our volunteer staff, any response to your queries will appear only in RDEX, so it would be more appropriate for you to present technical questions to the readers and ask for their responses which will then be placed in the Apple-RDEX.

## How to get a free library disk

Whenever possible, send everything on Apple format (5.25" - DOS/ProDOS or 3.5" - ProDOS) or IBM format (3.5") disks. Other formats are acceptable but there may be some delay as we look for someone to translate it for us. (*If you use a 5.25" disk, when we print your letter, we will return your disk with the current library disk copied onto it.*) Use whatever text editor you like, but tell us which one. Put a label on the disk with your name (or pseudonym) and address (if you want to receive mail). Don't reformat any programs or include them in the text of your letter. Send Applesoft programs as normal Applesoft files and machine language programs as normal binary files. We have programs to convert them to the proper format for printing. If you are

sending source code files, and you are not using the S-C Assembler, send them as normal text files.

## When to include a printed letter

Don't include hardcopy (printout) unless:

- a. You are writing about a bug or other printing error.
- b. You are writing to ask for help.
- c. You are answering another readers help request.
- d. You are writing about your subscription or sending an order for back issues or software.

Bugs, requests for help and answers to requests for help are bumped to the head of the line and go in the very next issue. All other letters are printed in the order that we receive them.

## Writing to get help

When writing to request help, be sure to include ALL relevant information. The more information you include, the easier it is to find a solution. There's an old saying that goes "A properly framed question includes 90% of the answer".

## How to get mail

If you are interested in receiving mail from other readers, be sure that we have a current address. If you use a pen name and want to receive mail, we need to have your address. Our readers privacy is important, so we will not print your address unless you specifically say too.

## How to write to RDEX authors

When writing to one of the RDEX authors. Write your letter and seal it in an envelope. Put your return address, the authors name (as it appears in RDEX) and the correct postage on the envelope. Put this envelope into another and send it to RDEX. We will put the correct address on your letter and mail it for you. Check to the right of the authors name to see if the author is writing from a foreign country and include the proper postage.

## Help Line

These readers have volunteered their time to help you. Please call only within the given time frames (corrected for your time zone). No collect calls. (You can write anytime!)

Jack Nissel (Disk Protection, 7-10PM EST)  
(215) 365-8160

Marc Batchelor, 6025 Coker St., Cocoa, FL 32927

Rich Etarip, 824 William Charles Ct. #2, Green Bay, WI 54304-4018

## The BBS

(Bulletin Board System)

Dave Goforth is the sysop for the Computist BBS. The number is: (206) 581-9292. If you already have a User ID# and password, sign-on using the User ID#. If you are a new user, it may take a day or so to validate your new ID# and password.

## Readers Data EXchange

New COMPUTIST readers using Apple IIs are advised to read this page carefully to avoid frustration when attempting to follow a softkey or entering the programs printed in this issue.

### What is a softkey, anyway?

Softkey is a term which we coined to describe a procedure that removes, or at least circumvents, any copy-protection on a particular disk. Once a softkey procedure has been performed, the resulting backup copy can usually be copied by the normal copy programs (for example: COPYA, on the DOS 3.3 System Master disk).

### Commands and control keys

Commands which a reader is required to perform are set apart by being in boldface and on a separate line. The return key must be pressed at the end of every such command unless otherwise specified. Control characters are preceded by "ctrl". An example of both is:

```
6 ctrl P
```

Type 6. Next, place one finger on the ctrl key and then press P. Don't forget to press the return key.

Other special combination keypresses include ctrl reset and open-apple ctrl reset. In the former, press and hold down the ctrl key then press the reset key. In the latter, press and hold down both ctrl and open-apple then press reset.

## You have a LEGAL RIGHT to an unlocked backup copy of your commercial software.

*Our editorial policy is that we do NOT condone software piracy, but we do believe that users are entitled to backup commercial disks they have purchased. In addition to the security of a backup disk, the removal of copy-protection gives the user the option of modifying programs to meet his or her needs. Furthermore, the copyright laws guarantee your right to such a DEPROTECTED backup copy:*

... "It is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner."

United States Code title 17, §117

## Table of Contents

**Most Wanted softkeys** .....23

**RDEX Contributors**.....23

**unClassifieds** .....23

**The Product Monitor** .....4

Big-Mem Fix for Appleworks 2.0 .....5

Budokan: The Martial Spirit .....5

Build Your Own Stereo Board! .....5

Happy CRT, The .....5

I Want My GS RAM III! .....4

Magic Candle II Support .....6

Missing Persons .....4

Platinum Painting Made Easy .....6

Their Finest Hour: The Battle Of Britain .....5

Vendors .....6

Wraith .....4

**Bugs:**

Bug in Algebra vol 3 & 4 .....21

Bug in Matchmaker

    World Geography Facts .....21

Bug in PSAT Word Attack Skills .....21

Bug in PSAT/SAT Analogies .....21

Bug in SAT Word Attack Skills .....21

## Features, Notes and such:

Basics of Kracking Part #14, The .....15

Build Your Own Stereo Board! .....5

Kracking SSI's RDOS .....15

MECC 3.5" disk softkeys .....20

MECC 5.25" disk softkeys .....20

Notes on Test Drive II cars disks .....22

Questron II CDA v2.0 .....7

## Softkeys:

Blue Grass Bluff (5.25") .....20

Bluegrass Bluff (3.5") .....20

Botanist's Apprentice (The) .....20

Cause & Effect: Mountain

    Climbing Blue Level .....21

Cleanwater Detectives (3.5") .....20

Cleanwater Detectives (5.25") .....20

Context Clues: Hidden

    Treasure Red Level .....21

Drawing Conclusions: Chief of

    Detectives Blue Leve .....21

Drawing Conclusions: Chief of

    Detectives Red Level .....21

Estimation Activities (3.5") .....20

Estimation Activities (5.25") .....20

Fact or Opinion: Smart

    Shopper Blue Level .....21

Fact or Opinion: Smart

    Shopper Red Level .....21

Field Zoologist, The .....20

Following Directions: Behind the

    Wheel Red Level .....21

Getting the Main Idea: Around the

    World Blue Level .....21

Getting the Main Idea: Around the

    World Red Level .....21

Grammar Gazette (3.5") .....20

Grammar Gazette (5.25") .....20

Grammar Monsters (3.5") .....20

Inference: School Days Blue Level .....21

Inference: School Days Red Level .....21

Journey to the Age of the Dinosaurs .....20

Littletown Zoo (3.5") .....20

Littletown Zoo (5.25") .....20

Living Cell (3.5"), The .....20

Living Cell (5.25"), The .....20

Outpost .....17

Paper Plane Pilot (3.5") .....20

Paper Plane Pilot (5.25") .....20

Pensate .....18

Problem Solving With Nim (3.5") .....20

Problem Solving With Nim (5.25") .....20

Reading for Detail: Race Track Blue Level ...21

Reading for Detail: Race Track Red Level ....21

Sequence: What Comes First Blue Level .....21

Sequence: What Comes First Red Level .....21

Seven Cities of Gold .....18

Skills Bank II .....22

Snack Attack .....17

Spectre .....19

Superkey .....19

TEAM Series: General Interest .....20

TEAM Series: Making Choices .....20

TEAM Series: People and Places .....20

TEAM Series: Science and Health .....20

Time Navigator Around the World (3.5") .....20

Time Navigator Around the World (5.25") ....20

Twerps .....17

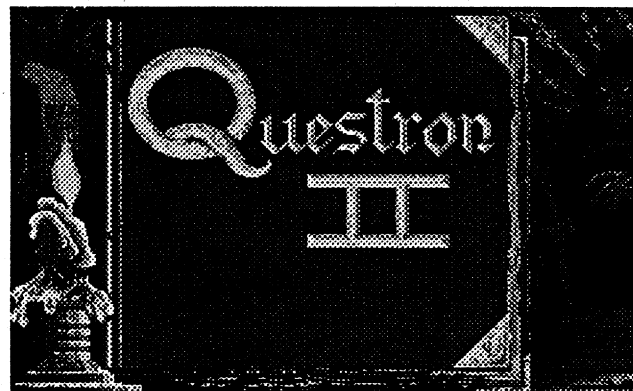
Vacation Nation Travel (3.5") .....20

Vacation Nation Travel (5.25") .....20

Where in the World is Carmen Sandiego? ....18

Woolly Bounce (3.5") .....20

Woolly Bounce (5.25") .....20



## Advanced Playing Techniques:

Questron II .....7

## Playing Tips:

Questron II .....7

## Questions:

Ⓟ .....16, 21, 22

## IBM Softkeys

Aardvark McGraw-Hill Financial Software ...22

Ancient Art of War .....22

Astrilis .....22

ATI Training Software .....22

Faery Tale, The .....22

Gauntlet .....22

Grave Yardage .....22

Gun Boat .....22

## Editorial Notes

Hey, All you Computist writers! I've only got enough material for another one and a half issues. How about finishing up what your working on and sending it in to the RDEX editor? And while we're on the subject, I'd like to see some BASIC programs submitted. Nothing too complex, just some common but useful items.

### In the News—some moldy old flies

Bugged by household flies? It seems that someone has finally come up with a non-chemical way to zap houseflies—with mold. Sprayed from an aerosol container, it is breathed by the flies into their bodies, where it multiplies and kills them. And it's harmless to mammals, birds and the home owner. What's next?

### High-Definition TV (HDTV)

Called "Hi-Vision" in Japan, it means wide 200 inch screens and 1125 line resolution. In Europe it's 1250 lines. And in the U.S., well we don't have it yet, but they say it's coming.

The American digital standard should be able to surpass the quality of the European and Japanese versions of HDTV. Plans are afoot to start HDTV broadcasts by 1998 and to phase out regular TV by 2008. The days of regular TV are numbered and the countdown has begun.

In the meantime, buying a large-screen TV may not be such a bad idea. You'll get plenty of use out of it before you need to switch over to HDTV

# The PRODUCT MONITOR

## RATINGS

- Superb ★★★★★
- Excellent ★★★★★
- Very Good ★★★
- Good ★★
- Fair ★
- Poor ☹
- Bad ●\*
- Defective ✖

## Missing Persons

Some months ago a caller to the popular Rush Lembaugh talk show complained that computing had become a male dominated 'club' and seemed to imply some conspiracy to exclude women. Rush, an avid Mac user and promoter, argued this wasn't so. Since, by his own admission, Rush is right 94.5% of the time, that was that. 'The Host' had spoken!

More recently, one of the early morning TV news programs did a report on "Kids and Computer Games". As usual, the piece zeroed in upon users of handheld cartridge machines and the threatened onset of mindlessness. Standard stuff, except that none of the experts interviewed seemed especially worried! Facing the prospect of a no crisis, low pizzazz wrap-up, the reporter happened to notice that virtually all of the zombiehood candidates were boys. "Where are the girl zombies?" she asked. "Ah, well," the game seller responded, glancing back at shelf upon shelf of glistening cartridges. "We do have a game, named 'Barbie', for girls."

Obviously, women use computers. Just as obviously, very few are 'into computing'. Users group meetings, hardware swap turnouts, program publication, even BBS postings—all are overwhelmingly male. It looks like a conspiracy; but, it isn't. Those 'into computing' would be pleased as pie to have more girls and women join in.

"Wait just a minute! If the talk show caller is typical and 'insiders' are no obstacle, what's the problem?" The problem is incentive. You have to enjoy 'computer stuff'. Otherwise, it's just plain work. In the '70's a lot of us got started by tinkering with hardware. Since the '80's, the usual 'starting off' incentive has been game playing. (i.e. Real games on real computers) With few exceptions, men wrote the games, played the games, and became THE market for still more and better entertainment software. 'What's fun' in computer use is, largely, what's fun for males.

If more games were written for women, more women might play games. For example, in Origin's "Wing Commander II", female Space Aces would find that Col. Devereaux is a romantic Frenchman. An SSI AD&D adventure might permit identifying a "lead character"

and then adjust events according to gender. (The player can rescue a handsome prince, etc..) Since today's designers, artists, and scenario writers ARE more skilled than ever at 'playing the gender card', tailoring major entertainment products for women could be very profitable.

Computing, the kind that goes beyond merely viewing the computer as a business tool, involves tinkering, experimentation, and playing. Some of this (e.g. club meetings, trading ideas with friends, and modeming) entails social interaction; but, a lot doesn't. It's just you, your machine, the software, and a cup of coffee. Much of the time, any distinction between work and play is ambiguous at best. Unfortunately, the "womanly behavior" stereotype does not encourage play, except in social situations. It is "okay" for children and men to build telescopes, fly RC models, AND 'mess around' with computers. For women such behavior is "self indulgent" and "immature". Where, ever, have you seen a movie/TV mom sit down, alone, for a session of computer gaming? The notion of a 30-year old female account executive coming home after a rough day on the job, plopping into a chair next to her new '486 PC, and booting "Pools of Darkness" is positively alien.

You can't have it both ways. To "get into computing" one must be ready to play. Mastering the 'do everything machine', enhancing self-expression, career advancement—all of the 'serious stuff'—will come in time. Women enforce the stereotype; it's up to women to permit exceptions. For the talk show caller the solution may not be easy; but, at least, it's simple. Buy a computer, go to users group meetings, try some games, subscribe to Computist. In short, hop in! You really don't have to remain among the 'missing persons'!

media with scenario, directions, and over-view map all on-diskette. No frills, true; but, with three large towns plus assorted castles and dungeons, you get loads of 'olde Ultima'-style play! This means crisp, top-down view hires displays showing grasslands, water, woods, etc. when traveling outdoors. Inside, you see walls, doors, altars, counters, signs, etc.. You show up as a partially-animated hero figure as do merchants, hotel clerks, tavern keepers, guards, ... and monsters. A few keyboard inputs handle purchases, combats, and movement on the smooth-scrolling maps.

Despite its "Ultima"'s III-V 'look', Wraith avoids puzzles, allows 'conversations' only with barkeepers, seers, etc. behind counters, and offers very limited choices for improving weapons, armor, and magical artifacts. Aside from an amulet to detect secret doors and Recall to teleport out of tight spots, magic chiefly consists of Fireball plus a few additional attack spells. Similarly, all players—there's room for three on the game diskette—begin with the same character: a male human fighter/mage. You pick a name; but have no say about profession, race, gender, and starting attributes.

On the other hand, new adventure gamers are likely to appreciate several features aimed at streamlining play. These include auto-equipping of purchased weapons and armor, bows which never run out of arrows, low-cost healing potions and spells, unlimited carrying capacity, and automatic Level advancement based upon Experience won by mashing monsters.

One automatic feature of which to be wary is the Game Save update performed whenever you enter "Q"uit OR move to a new map (e.g. enter a castle, go down a stairway, etc.). Should you be killed, you are chastised by your patron goddess, zipped back to 'start' in Tarot City,

adversaries in a running battle through a dungeon's rooms and corridors. Nor do 'monsters' always attack simply because they are on-screen. (Some monsters see better than others, are more aggressive, or have guard duties.) However, if you attack someone or someTHING, any other guards or monsters which see the attack will move against you. Another clever wrinkle is the lavish use of 'signs' to supply hints (e.g. "You're Getting Close!") and to identify dungeon areas, like the "Royal Gardens" and a boulder "Throwing Range" for giants.

Spanning several large, well thought out mazescapes, Wraith can, thanks to the numerous landmarks, be played with or without mapping. "With" is better in the long run. Line-and-box sketches take little effort and have proved to be great time savers. Bugs? Just one is worth noting; and, it may be a case of purposeful deception: look North, not South for the "secret base". Overall, expect 30-40 hours of smooth-running, medium-difficulty challenge. Getting past the Wraith's minions is not easy; you CAN get killed down there!! But, then, how long has it been since you've gotten into a for-real Ultima-style adventure? Well, that's too long!

## I Want My GS RAM III!

★★★★

\$199-\$499 (1MB-4MB)

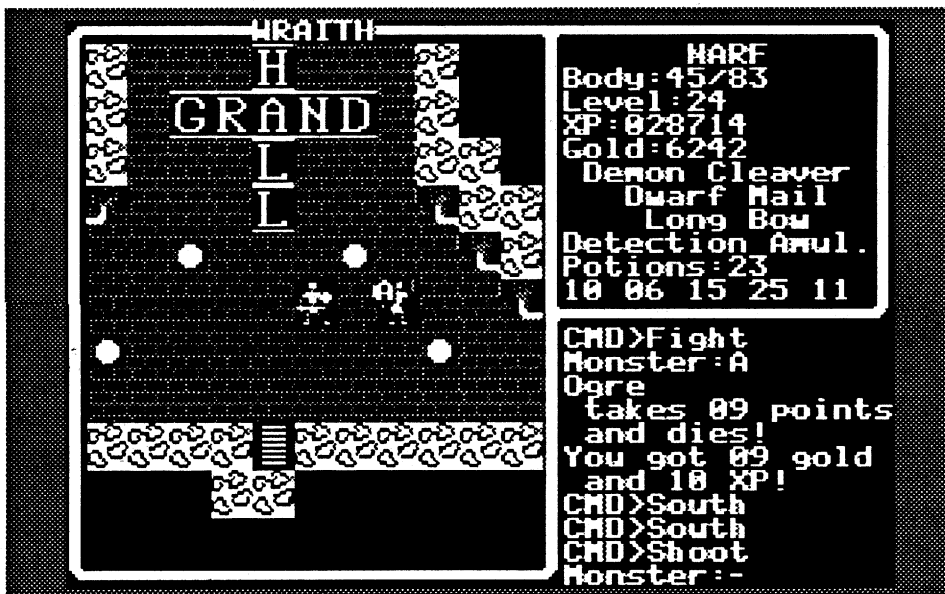
\$139 (2MB) with trade-in, for Apple IIs

Applied Engineering

If M-TV were as good as the (now classic) rock hit, I might actually watch it. Applied Engineering's GS RAM III IS a real goodie; and I am 'watching' it even as I type. Supplied as a compact 4" x 2" card, 'RAM III can add up to 4MB (6MB with special expander) of fully DMA-compatible memory AND leave plenty of room for a component crammed card in Slot 7. Since AE knows you're going to ask "Well, what about the 1MB memory board I've got now?", you can trade in (i.e. dump) your oldie and get a price break on the new board with 2MB installed.

GS RAM III uses the new 1MB x 4 zip package RAMs (e.g. 80 ns. NEC D424400V or Toshiba TC514402). Each is about the size of an old 16K RAM IC but with 'all the little feets' along one edge. A 1MB set, two chips, currently sells for about \$60 from suppliers like Quality Computers. When the board is installed the zips work like vertical fins to draw up air and dissipate heat. (Incidentally, this feature was pointed out by QC's Walker Archer, who noted their tech department "likes the 'RAM III very much".) At just 50-60 ma. of current per set, even a full 4MB draws far less than the 600 ma. allocated to the memory expansion slot.

The AE board package includes a manual describing Installation, How to add more zips, and RAM Disk use. (The manual does NOT, unfortunately, supply technical specifications—like chip info, current requirements, etc.—or a schematic.) You also get the "AW 3 Expander" diskette plus manual. It has a rigorous Memory Test program, Auto-Copy utility (e.g. for boot-loading a diskette or selected files onto RAM Disk), and utilities for adding various enhancements to "Appleworks" 1.2-3.0.



## Wraith

★★

\$9.95 for 128K Apple IIe, IIs

Nite Owl

The once peaceful island of Araithia is in serious trouble. A dark force has called up orcs, ogres, evil mages, hydras, and other horrors to roam the countryside. As leading citizens and lords fall beneath the dread influence, even the towns and castles have become centers of evil. Now YOU, protector of the last temple of Metiria, have been called. Find the great weapons, armor, and magical items, develop your powers, and destroy the Wraith!

No, the price is correct. Nite Owl's map/maze adventure comes on 3.5"

and divested of any gold. (Other possessions, Level, and attributes are not affected.) Naturally, the gold-less game is also saved! (Hint: If you have acquired a pile of gold and your demise seems imminent, REMOVE the game diskette from the drive. At worst, you will have a choice of putting it back in when prompted or rebooting and resuming play from the last Save.)

Typical of games by John Carmack, Wraith does not fail to introduce some nice improvements on the format it borrows. Since you are guiding a single character, the game allows full tactical combat on the current map. While the graphics and sound closely match those of the later Ultima's (except, there's no music), 'wide open' combat results in more challenging, more entertaining encounters which may involve several

On a IIGs, the most valuable mods increase maximum document size and number of Data Base records. As might be expected, "Expander" and "Time Out" do not 'get along' well. The AE enhancements must be made to an unmodified copy of "Appleworks"!

Of the recent upgrades made to our "Limited Edition" Woz IIGs, it is not at all easy to say which produced the larger gain in computing Power. Zip-GS speed is VERY nice, especially for graphics-intensive applications. GS-RAM III space is absolutely addictive! (Kind of like the supply of beakers in a chem lab; you end up using and needing whatever you've got.) For sure, being able to put GSOS on bootable RAM disk (and still have plenty of memory left for applications) delivers an 'order of magnitude' boost in user convenience. With RAM Disk boot times under 10 seconds, just think back to YOUR last multi-application, swap-in/swap-out computing session. OR take a look at memory size recommendations for some of the latest IIGs products. What's left to say but "I want my GS RAM III, too!"?

## Guest Reviews

### Their Finest Hour: The Battle Of Britain

review by Douglas Hecht

You rev-up your Supermarine Spitfire Mk I, tune in your radio, and take off. It is the Summer of 1940. You and the other RAF pilots in your squad have just learned that there is a Luftwaffe bomber group converging on one of your airfields. The Junker '88A-1's are being escorted by a squadron of Messerschmitt 'Bf 109E-3s. Your mission is to intercept the enemy force over the English Channel and shoot down as many aircraft as possible. Off in the distance you see the German formation approaching the North side of the channel, so you climb up to about 10,000 feet and sneak up on them from above and behind - get the first bomber in your gunsites and .....

Fly the RAF's Spitfire or Hawker Hurricane Mk I; or try a Luftwaffe fighter or one of four German bombers! Lucasfilm's 'Battle Of Britain' (\$54.95 for CGA-VGA 640K PC) PC) is packed with excitement. Catching a 'schmitt in your tracers, you watch it blossom into churning smoke and fire. Guiding a German bomber, you sit in the Bombarrier's seat looking down to site-in your targets; they look great before you drop your load of bombs and wipe them out! Graphics and control response are just outstanding. The AdLib/SB effects will blow you away!

Supplied with user-friendly, photo-filled 'collectors quality' manual, 'Battle of Britain' includes 72 pre-defined missions PLUS the ones you design using the Mission Builder. The MB allows you to create you own objectives and battles based on historical data. To sum it all up: there's a lot of fun contained in that classy-looking box; and, it just keeps coming!

### Budokan: The Martial Spirit

review by Douglas Hecht

You say that you're 'into' the video game Martial Arts experience; but, you are tired of having to go to the local arcade parlor every night to play. Well, maybe you should save those quarters

and try the Budokan! Once past the gates of the 'Tobiko-Ryu Dojo', you can enter training halls for 'Karate', the art of hand-to-hand combat, 'Kendo', the art of sword fighting, 'Nunchaku', fighting with Nunchak's, or 'Bo', the art of fighting with staffs. Practice alone, with an instructor, or take on a second player at the 'Free Spar Mat'. Control options include KB-only, KB and joystick, or two joysticks. (I prefer using a joystick; it seems to make it easier to move your man.) During your training, you also have the option to visit the 'Tobiko-Sensei' to receive words of wisdom about tactics and strategy along with philosophical guidance.

When you feel that you have mastered the arts well enough to hold your own in a tournament, it's off to the Budokan where you will fight students from other Dojo's. They can teach you skills not taught by your instructors, so long, that is, as you keep winning. There are 12 other students that you can fight at the tourney; but, you must beat one before you can fight the next.

Electronic Arts' Budokan' combines a great challenge with superb graphics. I really felt like I was playing a video game at my local mall while the package was running (including no Game Save). The only difference was that I didn't have to keep feeding my PC quarters every time I lost a match! Starting with a "fable" (about some 'Monks', some 'Grain', and some 'Software'), the manual describes each move and adds some interesting background 'color' on the Martial Arts tradition. (\$49.95 for CGA-VGA 640K PC)

## Fast Frames, Updates, Etc.

### Big-Mem Fix for Appleworks 2.0

Okay, I know every IIGs owner is supposed to be running "Appleworks 3.0" or "Appleworks /GS" or "Word Perfect/GS". Many do; but, lots of users like their multi-patched, "Time Out"-enhanced '2.0's... at least, for 'everyday stuff'. Bolstered by Zip GS speed and 2 MB of AE's GS-RAM III, my Appleworks 2.0 was looking better than ever. THEN I popped in another MB and (Zonko!): bombsville! Starting from the GSOS finder or booting straight from ProDOS 1.8, the program makes it into the "Pre-loading Appleworks into language card" message and crashes.

"Most likely you got some bad RAM zips." Logical; but, AE's memory test program said: "Uh, uh. Your RAM is working fine." Besides, swapping around the zips and going back to 2 MB— everything worked again— seemed to nail the culprit as memory Size. Possibly, the program has trouble dealing with ANY memory larger than 2.25 MB; for sure, it does not like 3.25 MB.

Note: Shortly after finishing this piece, I happened to be talking with AE's Nole Holmes about GS-RAM III and mentioned the problem. Sure enough, it turns out Appleworks 2.0 is noted for not "liking anything over 2 MB"!

My Appleworks 2.0 was exhibiting a Big-Mem Bug. I wanted to squash the critter; and, if your '2.0 has the same problem, so will you. As to finding the bug...? The fix I arrived at is the result of a guess that a call to the Total Mem-

ory tool might be involved. The critical machine code looks like this:

```
A2 02 1D (Load X: low= Toolset
           #2; high= tool #29)
22 00 00 E1 (JSL to Tool Dis-
           patcher routine)
```

In my version, the code shows up at \$24F4 after BLOADing the Appleworks system file (APPLEWORKS.SYS) at \$1000. To find it in your Appleworks 2.0, BLOAD the system file, go to the monitor, and enter  
VA2 02 1D<1000.4FFFF

The JSL instruction routes execution to Tool #29 in the Memory Manager Toolset (#2) in the IIGs ROM Toolbox. It tells your program how much memory is installed. Since this Total Memory number is the one Appleworks' doesn't seem to like, I substituted \$1B for \$1D at \$24F6 and saved the newly patched file: BSAVE APPLEWORKS.SYS, TSYS, A\$1000, L12390 (length of file as shown in CATALOG). Now, the Dispatcher call goes to Tool #27 in Toolset #2, the Memory Manager's Available Memory tool.

A disadvantage of the 'black box' approach to debugging is that one can never claim that a bug is "eliminated". Maybe the TM call IS the BM Bug; maybe not. In any case, with 3.25 MB installed, my 'original' copy still bombs. The BMB-patched Appleworks 2.0 boots without a hitch!

### The Happy CRT

No doubt about it, among the major components of your PC and/or Apple II system, the CRT display takes the most abuse. Day after day it gets turned ON and Off and run for long periods, all the while building up heat and collecting dust. You can simply wait for all of this to take its inevitable toll; OR, you can take a few simple steps and have a much happier, longer-lived CRT display:

1. Wash me, Please! Dust reduces airflow, increases heating, clogs controls, and can produce arcing from high voltage wiring. If your monitor has been in service for more than three or four years, it's about time it was cleaned. Since, just for starters, a thorough cleaning means removing the case and disconnecting control cables, it is not a job for someone lacking experience with CRT display innards. (Aside from HV shock and tube implosion hazards, there are just too many ways to blow it.) A novice might get a CRT-wise friend to help, contact a maintenance shop, or settle for running a vacuum cleaner hose over the case vents (top, back, sides, AND bottom) with the display turned OFF.

2. Cool it. Displays rely upon convection to move air in, from lower vents, and out through the top. Avoid crowding the display monitor into a cramped hutch and never block off the top with manuals, papers, etc.. On a IIGs the monitor normally sits just above the fan output vent of Kensington's "System Saver IIGs"; so, these users will get the additional benefit of some forced-air cooling.

3. Glitch not; switch not. Naturally, you are already using a "System Saver IIGs", one of the top-of-PC units, and/or some other AC line deglitcher/surge protector. Surges also occur each time you turn ON the display. (Switches arc, coils produce inductive spikes, capacitors suck in large currents, etc..) For any piece of electrical equipment, Turn-ON is a stressful instant, the moment when

failure is most likely. You cannot do much about the stress on your monitor's components; but, you can reduce the number of times you 'roll the dice'. From now on, when you must turn OFF your computer briefly to obtain a 'cold start' or to change some sensitive I/O connection, avoid hitting the Master Switch on your "System Saver" ("Power Console", etc.). Turn OFF only the computer.

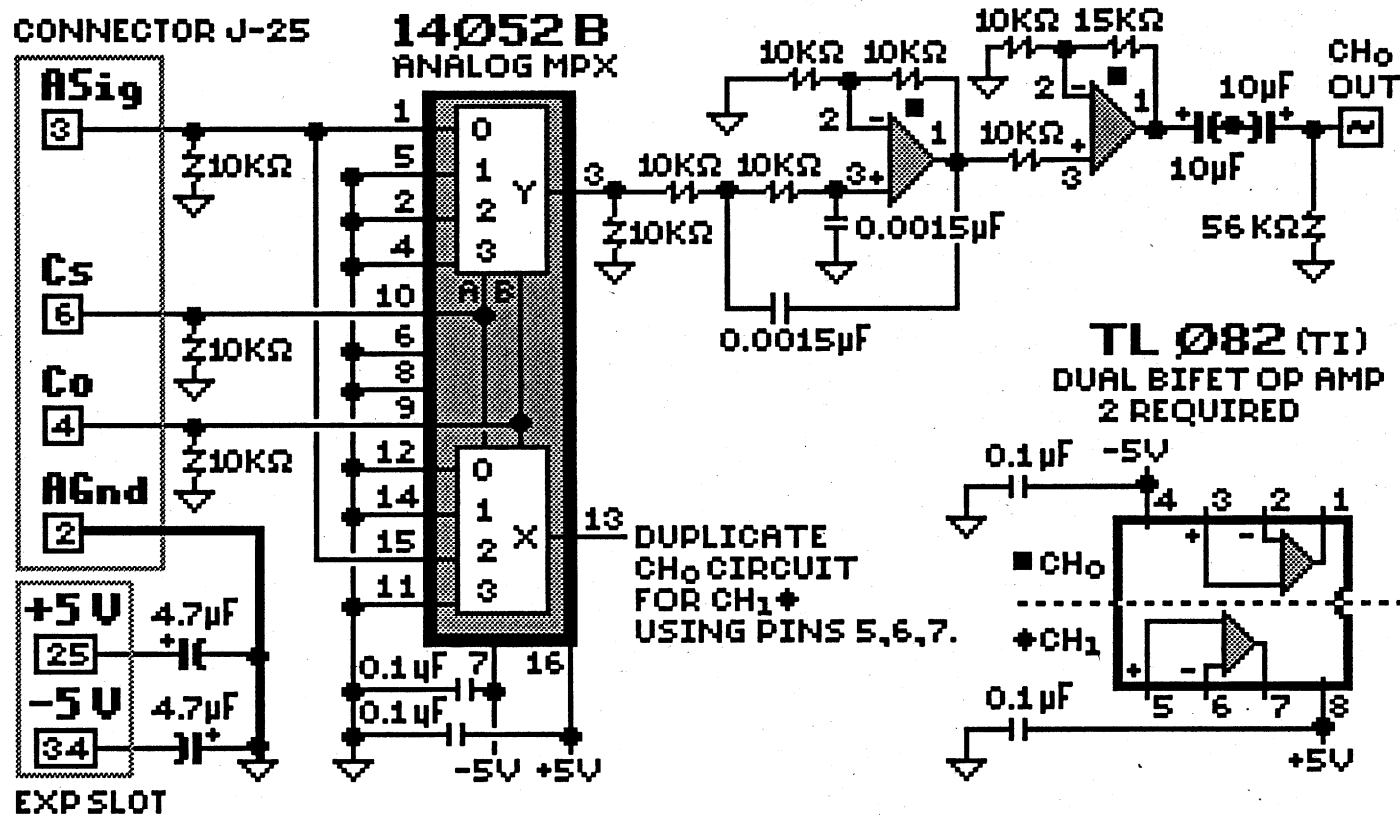
4. Learn not to burn. Use "screen saver" TSR's (CDA's, NDA's). Should a user leave the system running and forget to return for a few hours, a saver will blank the screen or animate the display to prevent burning of a fixed image into the screen phosphor. If you are the only regular user and would prefer not to install saver software, just get into the habit of turning down screen intensity whenever you get up "for a few moments" to answer the phone, brew some coffee, or check a back issue of Computist. The final 'no burn' hint relates to YOU: stay away from the screen. Yes, people have been using CRT displays for years; but, there is some concern about health effects, the need for reduced radiation CRT's, etc.. It's just common sense to place the screen at least 18"-24" away.

5. Seeing is believing. If your CRT monitor has slipped out of focus, what you WILL believe is how much better the display looks IN focus! "Good idea. I'll adjust the focus next time I start up the system." Ahem, what you WON'T believe is how much bother is involved. On today's CRT monitors, focus is, generally, a 'set it and forget it' adjustment performed at the factory. Many monitors do not even supply an access hole, let alone a knob, to adjust focus. For instance, on the IIGs you must remove the case to get to a control molded into the HV power supply. (i.e. This is another job for which you may want some expert help.) If your monitor has been in service only a year or so, it probably needs no adjustment. Otherwise, especially if the display actually looks a tad fuzzy, a focus touch-up is in order.

### Build Your Own Stereo Board!

Since "Hyperstudio 3.0" comes with its own low-noise A/D input board, the main reason for adding a IIGs 'sound board' is to obtain two-channel output. "Hey, my IIGs has a stereo cable plugged right into the Sound Output jack. Isn't THAT two-channel sound?" Nope; and, it is not "stereo" either. The jack is a standard stereo jack; but, the "Left" and "Right" outputs are connected to a single 'composite' source. Possibly, Apple originally intended to supply two-channel sound; perhaps, the idea was just to simplify connections to stereo systems. (Stereo devotees get upset when you give them just one 'channel' to plug in.) Whatever, the fact remains: for multi-channel output you need to add a "sound board".

Why bother? Some programs offer true stereo— effects and music are lifted from stereo sources or 'recorded' using two mikes— others deliver simulated stereo. Many products output some effects (like a bow twang) through one channel and other effects (the THUNK! of an arrow hit) through the second channel. Stereo, of course, produces spacious, '3-D sound'; but, even the 'separated channels' approach can spread out the action and add realism.



**IIGS Stereo Board - TDX** TURDHIL DSG. 1992

From input to output, the TDX design is straight-forward and simple. First, IIGS audio enters the 14052 where the "Left" and "Right" channels are separated using the C0 input to turn ON the appropriate section (output X or Y) when its channel is valid. (This happens at supersonic speeds so that the user doesn't notice that each channel is ON half of the time.) Then, each output goes to a pair of op amps where it's filtered and amplified.

Design objectives were low noise, low distortion, and low power consumption. From the start I expected that on-board power amp IC's would be too puny to drive our speakers to desired volume levels and maintain low distortion; so, the board includes no power amp IC's and is not intended to directly drive low impedance loads such as speakers. Like a tuner, CD deck, or other hi-fi source, it connects to a stereo amplifier's AUX inputs (or "Tuner", "Tape", etc. inputs) or to the inputs of speaker

units with built-in amplifiers. TDX has plenty juice to drive any decent stereo system at 'blow out the windows' volume levels.

As shown in the diagram, nearly all connections between TDX and the computer, including ground, are made over the J-25 lines. Power (+5 Volts and -5 Volts) comes from the slot into which the board is plugged. The outputs go to "RCA-type" hi-fi jacks mounted on the rear of the board for easy access via standard audio cables. When placing the jacks, be sure to allow space (between the jacks and rear of the computer) for the cable plugs or arrange for the jacks to line up with an opening.

The J-25 connection is via a 7-pin mini-molex ribbon cable. It can go to J-25 (located near the memory expansion slot); or, if J-25 is being used by the Hyperstudio A/D input board, it plugs in there. (On both J-25 and its extension on the A/D board, pin #1 is nearest the front of the computer, pin #2 is next, etc..) All

of the parts, including the Apple-compatible circuit board, are commonly available.

I built the TDX stereo board near the start of the IIGS era just as games like "Tower of Myraglen" were beginning to appear. It sounded great then and it sounds great today running "Dungeon Master", "Instant Music", "Jam Session", and Hyperbole MIDI-synth pieces. If your IIGS is still in mono mode, why not make '92 the year you 'go stereo'. Add a commercial unit or build the TDX. Either way, when you spread out the sound you open up the fun!

**Magic Candle II Support**

Anyone who played the first "Magic Candle" will appreciate that 'working the kinks' out of Magic Candle II/PC might take some doing. While a long awaited mouse-input version remains in development, Mindcraft is now shipping the 1.50 revision which squishes bugs related to magic scrolls, mind-

stones, and characters who disappear while on assignment. (Other bugs which allowed walking on water and over mountains, denied access to ship travel, etc. were fixed in a series of earlier revisions.) After some experiences with having to restart play from scratch in other adventures, you can bet I asked about any similar requirement for MC2 fixes. According to Mindcraft's Dan Riddle, the game was specifically designed to allow players to retain characters and not suffer any loss in progress as upgrades occur. The 1.50 update diskette is free to game owners who request it.

**Platinum Painting Made Easy**

The one 'necessary item' not included in Beagle Bros's Platinum Paint package is a Quick Reference Card for keyboard commands. Particularly when working in full-screen mode, KB commands (e.g. to change colors, switch between Marquee and Pen tools, etc.) are great time savers. After yet another trip to the shelves (to find the manual and look up some key sequence) I decided enough was enough and put together the official Computist "Frequently Used KB Commands" card. Cut out, copy, or whatever. Slap the 'card' into a plastic cover, and enjoy!

**Next**

Look for Roger Wagner's HyperStudio 3.0 and Microsoft's Visual Basic plus Nite Owl's slip-on battery. SSI's Eye of the Beholder II' is one of several games 'on-deck', along with ... more!

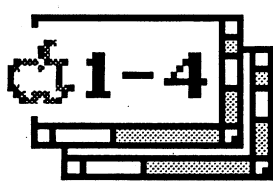
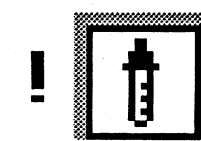
**Vendors**

APPLIED ENGINEERING: 3210 Beltline, Dallas, TX 75234 atten: Nole Holmes (214-241-0055, orders: 214-241-6060)  
 BEAGLE BROS.: 6215 Ferris Square, Suite 100, San Diego, CA 92121 atten: Bevey Minarovich (800-345-1750)

**Platinum Paint**

**FREQUENTLY USED KB CMDS**

▲ = SHIFT



F Fatbits

E Palette

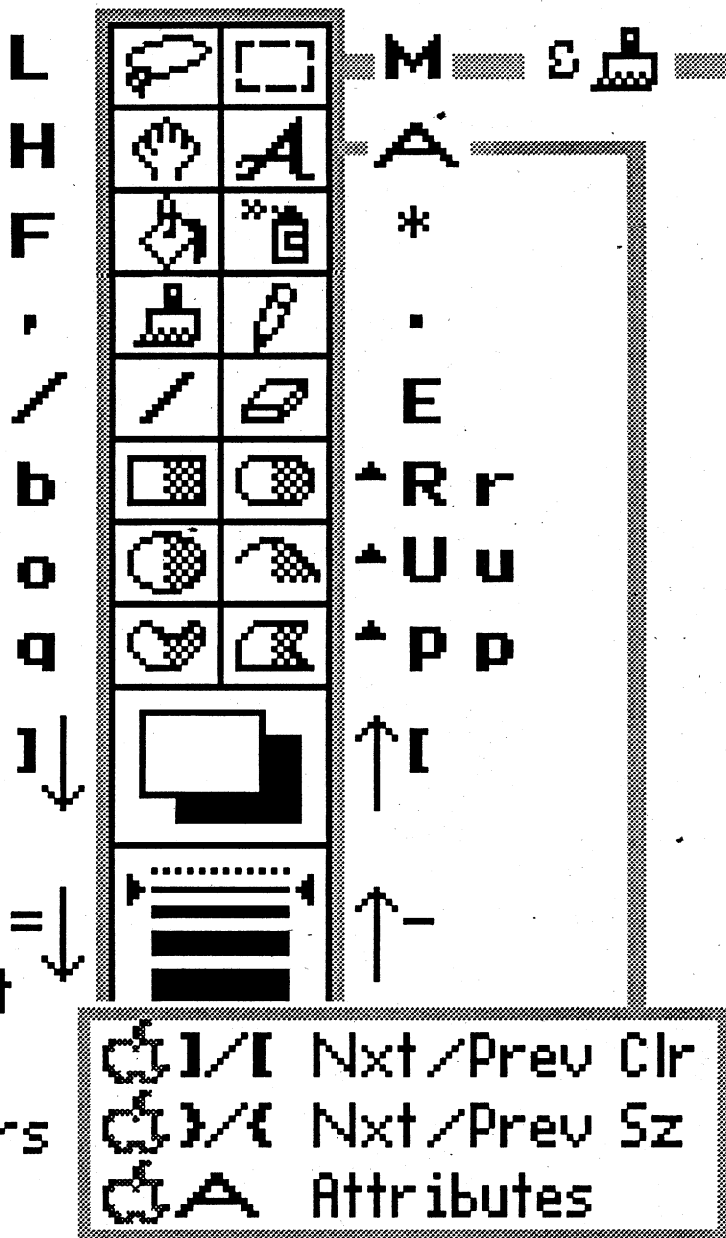
SPC Full Scrn =

ESC. Exit Text

I Info Bar

+ Crosshairs

++ Move



**ELECTRONIC ARTS:** 1450 Fashion Island Blvd., San Mateo, CA 94404  
 atten: Marci Galea & Lisa Higgins  
 (415-571-7171/ orders: 800-245-4525)

**Douglas Hecht:** 14724D Perthshire Road, Houston, TX 77079 (713-589-2176)

**HYPERBOLE:** 2402 Yoakum #2, Houston, TX 77006 atten: Greg Roach, Ed. (Subs: phone or mail to Resource Central)

**LUCASFILM GAMES:** P.O. Box 10307, San Rafael, CA 94912 atten: Sue Sesserman (415-721-3334; orders: 1-800-STARWARS) dist: EA

**MINDCRAFT:** 2291 205th Street, Suite 201, Torrance, CA 90501 atten: Dan Riddle (213-320-5215) dist: Electronic Arts 92

**NITE OWL PRODUCTIONS:** 5734 Lamar Blvd., Mission, KS 66202-2646 atten: Bob Shofstall (913-362-9898)

**PRODUCT MONITOR:** 7814 Santa Elena, Houston, TX 77061 atten: Jeff Hurlburt (713-645-8680)

**QUALITY COMPUTERS:** 20200 Nine Mile Road, St. Clair Shores, MI 48080 atten: Walker Archer (313-774-7200; orders: 800-443-6697)

**RESOURCE CENTRAL:** P.O. Box 11250, Overland Park, KS 66207 atten: Tom Weishaar (913-469-6502)

**ROGER WAGNER:** 1050 Pioneer Way, Suite P, El Cajon, CA 92020 atten: Garland Buckingham (619-442-0522)

**RUSH LEMBAUGH TALK SHOW:** EIB Network, 2 Penn Plaza, 17th Floor, NY City, NY 10121 atten: Rush Lembaugh

**STRATEGIC SIMULATIONS INC.:** 675 Almandor Ave, Sunnyvale, CA 94086 atten: Kathleen Watson (408-737-6800) dist: Electronic Arts

**ZIP TECHNOLOGY:** 5601 Slauson Ave. #264, Culver City, CA 90230 atten: PR/Mktg. (800-937-9737)

Vince Andrews WA

Advanced Playing Technique for...

### Questron II

?

### Questron II CDA v2.0

I have written this program for those that would like to alter their character while they play.

#### Information:

12/16/91 v1.0 - I started to write this program.

12/21/91 v1.1 - Fixed the search routine. It would lock up memory.

12/23/91 v1.2 - Fixed the Weapons & Armor routines. They now show the correct item you are currently equipt with.

12/30/91 v2.0 - Rewrote the search memory routines. Now it searches memory 10 times quicker than before.

#### Installing the CDA cheat

This can be done in two different ways. The first is by copying the file 'QUESTRON.CDA' to the folder \*/SYSTEM/DESK.ACCS. Then reboot the computer.

The second way is by using a CDA installer, such as the one I have included. Just launch 'QUESTRON.CHT' and it will install the CDA for you. (Both

files must be in the same directory or it will not work.)

### Using the CDA

Once the CDA is installed then you may launch the file 'Q2.SYS16' on the disk /Q2. After you have started to play the game, enter the Control-Panel and choose 'Questron II'. It will do a one time search so please be patient (about 10 seconds.)

You may now alter your character. Just select what you want to edit and press <Return>. Use the arrow keys to increase/decrease values. Hold Open-Apple key down while you press the arrow key and it will increase/decrease by groups of 5 or 100.

### Address where the game data is stored:

Hold the Open-Apple key down when you select the CDA cheat in the Desk Accessory Menu. It will display the address of where the game data is stored in the upper left-hand corner of the screen. I put that there for those that would like to snoop about memory.

### Questron.CDA Source

```

* Questron II CDA Cheat v2.0
* A classic desk accessory
* Vince C. Andrews 19 Dec 91
* Merlin-16+ Assembler
* Use OA-6 to Assemble
.
REL
LST off ;Turn listing on/off here
XC ;65C02 mode
XC ;65816 mode
MX %00 ;Full gs mode

```

```

* Equates and Macros used:
.
SET_80COL EQU $92
MTXT_OFF EQU $98
MTXT_ONEQU $9B
PNT EQU $E0 ;Also E1,2,3
KEY EQU $E0C000 ;Get key
STROBE EQU $E0C010 ;Clear Keyboard Strobe
_WriteChar MAC
Tool $180C
<<<
@TextWriteBlock MAC
PHL j1
PxW j2;j3
Tool $1E0C
<<<
_SetOutputDevice MAC
Tool $100C
<<<
_InitTextDevice MAC
Tool $150C

```

```

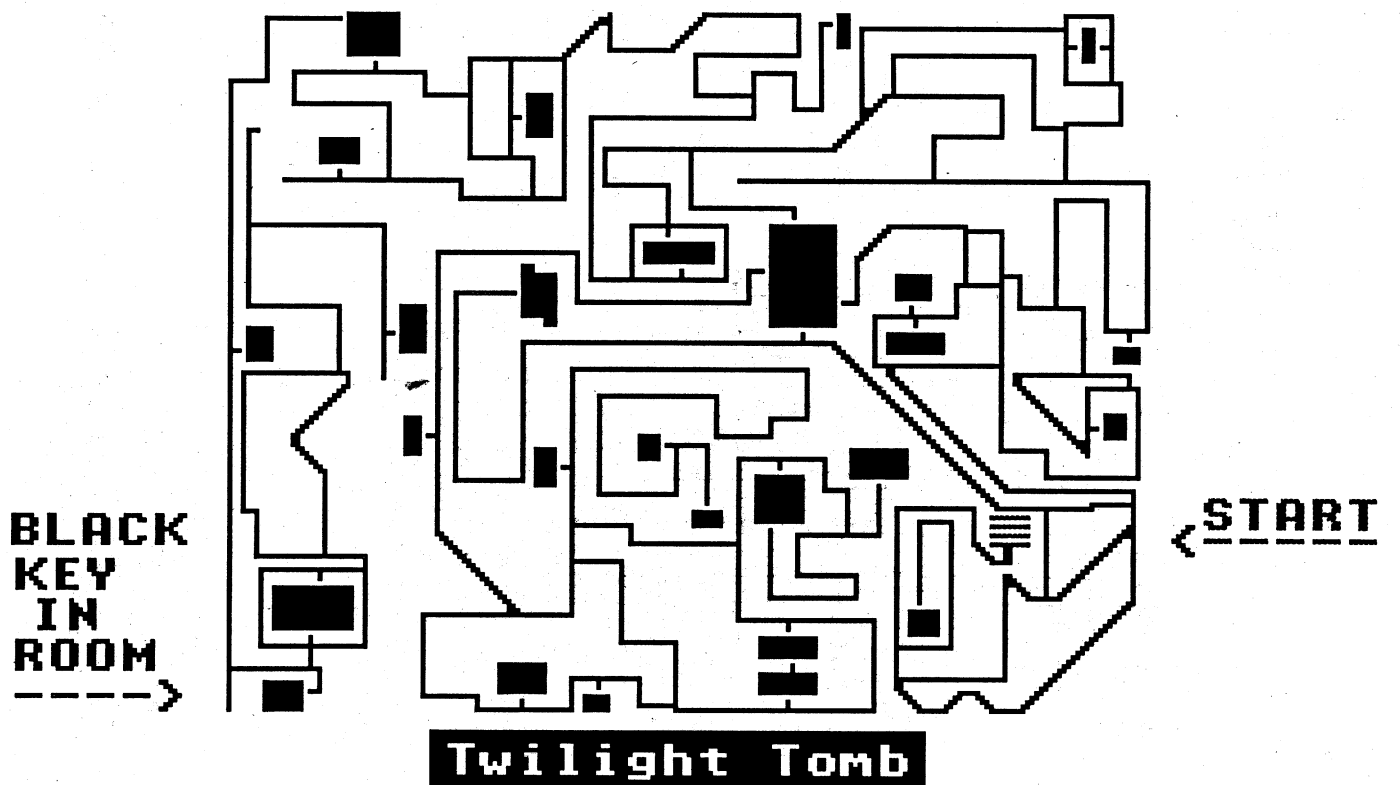
<<<
_WriteCString MAC
Tool $200C
<<<
PxW MAC
DO j0/1
PHWj1
DO j0/2
PHWj2
DO j0/3
PHWj3
DO j0/4
PHWj4
FIN
FIN
FIN
FIN
<<<
PHL MAC
IF #=j1
PEA ^j1
ELSE
PHWj1+2
FIN
PHWj1
<<<
PHW MAC
IF #=j1
PEA j1
ELSE
IF MX/2
LDA j1+1
PHA
FIN
LDA j1
PHA
FIN
<<<
Tool MAC
LDX #j1
JSL $E10000
<<<
.
* Required stuff of all CDA's:
* The string is what goes on the
* Control panel listing.
.
BEGIN STR 'Questron II'
ADRL START ;Entry point
address
ADRL EXIT ;Exit "routine"
address
; (just an RTL).
START PHB ;Save current data bank
PHK ;Get our bank
PLB ;and set data bank to it
PEI PNT ;Save the 4 zp locs we use
PEI PNT+2 ;and zero the bank byte
STZ PNT+2 ;for long addressing
pea 1 ;Pascal device type
pea 0 ;in slot 3
pea 3
_SetOutputDevice
pea 1 ;initialize standard output

```

```

_InitTextDevice
PEA SET_80COL
_WriteChar
pea $06 ;turn off cursor
_WriteChar
jsr Fast_Speed
stz Screen
jsr Border
lda Address
beq :nope
jmp Found
:nope pea #^WaitLine
pea #WaitLine
_WriteCString
ldx #1 ;start at bank 1
ldy #$1000 ;address 1000
stz PNT
stx PNT+2
NByte iny ;next byte
beq NBank
iny
beq NBank
cpy #bffd
beq :1
cpy #8ffe
bne CByte
cpx #1
bne CByte
inx
stx PNT+2
cpx #$10 ;quit at bank
bne NByte
pea #^NotFoundLine
pea #NotFoundLine
_WriteCString
jsr Get1Key
stz Address
jmp QUIT
CByte
lda [PNT],y
cmp Bytes2Scan
beq CByte1
cmp Bytes2Scan+1
bne NByte
dey
CByte1
lda [PNT],y
cmp Bytes2Scan
bne NByte
iny
iny
lda [PNT],y
cmp Bytes2Scan+2
bne NByte
iny
iny
lda [PNT],y
cmp Bytes2Scan+4
bne NByte
iny
iny
phy

```



```

phx
lda [PNT],y
pha
lda #0
sep $30
pla
plx
rep $30
plx
ply
cmp #$af

```

```

PLB ;Restore original data bank
PLA
STA PNT+2
PLA
STA PNT
EXIT RTL
*
* Print the Border Routine:
*
Border
pea MTXT_ON ;turn on MouseText

```

```

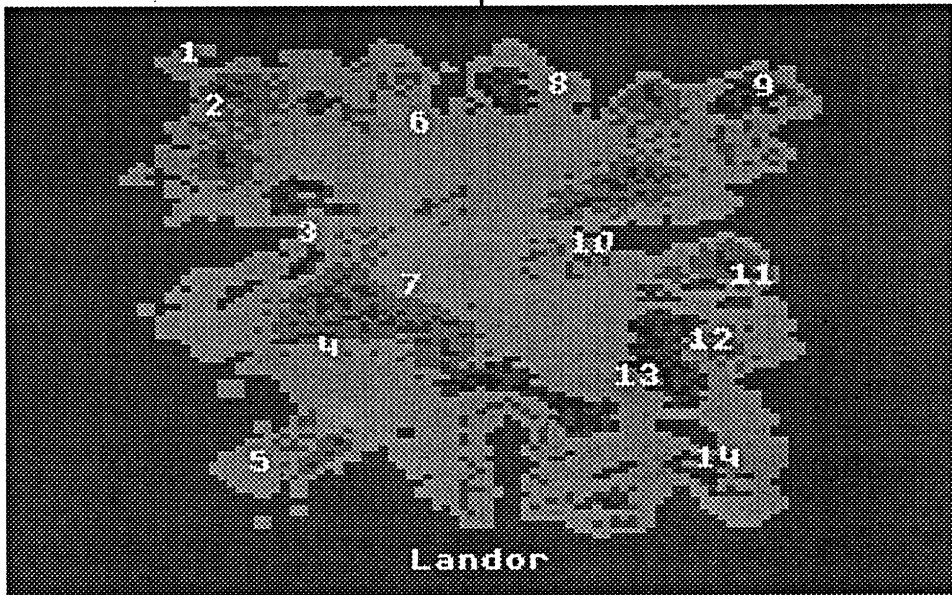
cpx #0
beq FE3
FE1 lda (PNT),y ;Table constantly changing
beq FE2
iny
bra FE1
FE2 iny
dex
bra FindEntry
FE3 rep $30
sty CurOffset
pea #f1
_WriteChar
FE4 lda PNT+2 ;Get hi addr
pha
lda PNT ;Get lo addr
clc
adc CurOffset ;Adjust it
pha
_WriteCString ;Print it
pea #e
_WriteChar
rts

```

```

dfb 42+32,5+32
asc 'Readied'
brk
*
* Gets a keypress:
*
SetBit ds 2
GetKey
jsr Get1Key
lda Char
cmp #$8a
beq GK_DownArrow
cmp #$8b
beq GK_UpArrow
cmp #$8d
beq GK_Return
cmp #$9b
bne GetKey
clc
jsr FE4 ;turn off the inverse line
sec ;set to point at QUIT
rts
CheckOA pha
sep $20
clv
stz SetBit
ldal $e0c061
bpl :1
lda #$60
sta SetBit
bit SetBit
:1 rep $30
pla
rts
Get1Key sep $20

```



```

bne :1
tya
sec
sbc #15
tay
lda [PNT],y
sta Address
iny
iny
iny
lda [PNT],y
sta Address+2
bra Found
:1 Found
jmp NByte
lda #0
sep $20
ldal $e0c061
bpl :1
rep $20
jsr Where
:1 sep $20
lda #7
sta BlankLine+2
rep $30
pea #BlankLine
pea #BlankLine
_WriteCString
stz Temp
stz Screen
Restart lda #0
ldx Screen
cpx #0
bne :submenu
lda Temp
:submenu sta Cursor
stz CurOffset
jsr FixTable
jsr Menu
:1 jsr FixTable
jsr SetCursor
jsr Readied
jsr FixTable
jsr GetKey
bcc :1 ;clear so reset cursor
lda Char
cmp #$8d
beq Restart
cmp #0 ;quit if char=0 and
beq QUIT ;carry was set
lda Screen
asl
tax ;otherwise, put cursor
lda Table5,x ;over quit because
sta Cursor ;esc was pressed.
clc
bra :1
QUIT CLC
REP %00110000 ;16-bit mode

```

```

_WriteChar
◊TextWriteBlock
;#Block0;#0;#Block0End-
Block0
◊TextWriteBlock
;#Block1;#0;#Block1End-
Block1
◊TextWriteBlock
;#Block2;#0;#Block2End-
Block2
pea MTXT_OFF;turn off MouseText
_WriteChar
rts
*
* Fix and update screen and table pointers:
*
FixTable
lda Screen ;What screen to show
asl
;Address every 4 bytes
tax
lda Table,x ;Get first 2 bytes
sta PNT
lda Table+2,x ;Get next 2 bytes
sta PNT+2
rts
*
* This will clear and draw the whole menu:
*
Menu rep $30
lda PNT+2
pha
lda PNT
clc
adc CurOffset
pha
_WriteCString
sep $30
ldy CurOffset
]ScrLoc1 lda (PNT),y
beq :1
iny
bra ]ScrLoc1
:1 iny
lda (PNT),y
beq :2
sty CurOffset
bra Menu
:2 rep $30
rts
*
* This will inverse current line:
*
SetCursor
lda Cursor
tax
ldy #0
FindEntry
sep $30

```

```

* Check and Print what is Readied:
*
Readied
lda Address
sta PNT
lda Address+2
sta PNT+2
lda Screen
cmp #2
beq :1
cmp #3

```

# Questron

## 2

- 1 OCTAPOINT
- 2 RIVERCREST CATHEDRAL
- 3 SEACREST
- 4 LYTON
- 5 REDSTONE CASTLE
- 6 CRAMFORD
- 7 LONG VIEW
- 8 GREAT PLAINS CATHEDRAL

- 9 HIDDEN ROCK
- 10 BAY VIEW
- 11 FOLMAN
- 12 ONTAGA
- 13 SANCTUARY CATHEDRAL
- 14 CROOKED PINE
- 15 SANTOR

## Cities of LANDOR

```

beq :2
rts
:1 sep $20
ldy #12
lda [PNT],y
cmp #$ff
bne :skip1
lda #0
bra :skip2
:skip1 inc
:skip2 clc
adc #4
sta :Text+2
rep $20
pea #^:Text
pea #:Text
_WriteCString
rts
:2 sep $20
ldy #13
lda [PNT],y
cmp #$ff
bne :skip3
lda #0
bra :skip4
:skip3 inc
:skip4 clc
adc #5
sta :Text+2
rep $20
pea #^:Text
pea #:Text
_WriteCString
rts
:Text hex 1e

```

```

ldal $e0c000
bpl Get1Key
stal $e0c010
rep $20
cmp #$e0
bcc :2
and #$df
:2 sta Char
clc
rts
GK_DownArrow
jsr FE4 ;turn off the inverse line
lda Screen
asl
tax
lda Cursor ;compare current line
cmp Table5,x ;to maximum line
beq :1
inc Cursor
clc
rts
:1 stz Cursor
clc
rts
GK_UpArrow
jsr FE4 ;turn off the inverse line
lda Cursor
bne :1 ;branch if not 0
lda Screen
asl
tax
lda Table5,x ;compare maximum
line
sta Cursor ;store maximum line here
bra :2

```



```

:1 dec Cursor
:2 clc
   rts
GK_Return
  lda Screen ;get screen #
  asl
  tax
  lda Cursor ;get line #
  cmp Table5,x
  bne :1 ;branch if not at quit
  lda Screen
  bne :skip ;branch if not main screen
  stz Char
  bra :skip2
:skip jsr Blank_Screen
:skip2 stz Screen
      ;(update screen)
      rts
:1 lda Screen ;get screen #
  beq :main ;branch if main screen
  lda Screen
  dec
  asl
  tax
  jmp (GK_Table0,X) ;execute routine
:main lda Cursor ;get current position
  sta Temp ;save cursor
  asl ;adjust by 2
  ldy Screen ;get current screen
]lup beq :done
  asl ;adjust by 2
  dey
  bne ]lup ;loop until y=0
:done tax
  jmp (GK_Table,X) ;execute routine
**
TO_C0
  pea #^:Text
  pea #:Text
  _WriteCString
  ldy #0 ;Change at this location
  ldx #9999 ;Max value
  jsr Change
  clc
  rts
:Text hex 1e
  dfb 42+32,5+32
  asc 'hp: '
  hex 1e
  dfb 48+32,5+32
  brk
**
TO_C1
  pea #^:Text
  pea #:Text
  _WriteCString
  ldy #4 ;Change at this location
  ldx #32767 ;Max value
  jsr Change
  clc
  rts
:Text hex 1e
  dfb 42+32,6+32
  asc 'gp: '
  hex 1e
  dfb 48+32,6+32
  brk
**
TO_C2
  jsr Blank_Screen
  lda #1 ;initialize status menu
  sta Screen
  stz Cursor

```

```

      sec ;return with carry set
      rts
**
TO_C3
  jsr Blank_Screen
  lda #2 ;initialize weapon menu
  sta Screen
  stz Cursor
  sec ;return with carry set
  rts
**
TO_C4
  jsr Blank_Screen
  lda #3 ;initialize armor menu
  sta Screen
  stz Cursor
  sec ;return with carry set
  rts
**
TO_C5
  jsr Blank_Screen
  lda #4 ;initialize spells menu
  sta Screen
  stz Cursor
  sec ;return with carry set
  rts
**
TO_C6
  lda Address ;rope/hooks
  sta PNT
  lda Address+2
  sta PNT+2
  bra :2
:again jsr Get1Key
  cmp #$8a
  beq :1
  cmp #$8b
  beq :2
  cmp #$8d
  bne :again
  pea #^chr
  pea #chr
  _WriteCString
  jsr FixTable
  clc
  rts
:1 sep $20
  lda #0
  ldy #$45
  sta [PNT],y
  rep $20
  pea #^No
  pea #No
  _WriteCString
  bra :again
:2 sep $20
  lda #1
  ldy #$45
  sta [PNT],y
  rep $20
  pea #^Yes
  pea #Yes
  _WriteCString
  bra :again
Yes hex 1e
  dfb 44+32,11+32
  asc 'Yes'
  brk
  brk
No hex 1e
  dfb 44+32,11+32
  asc 'No'
  brk
  brk
clr hex 1e

```

```

  dfb 44+32,11+32
  asc ' '
  brk
**
TO_C7
  lda #17
  jsr B_S
  TextWriteBlock
      #About;#0;#About_End-
      About
  jsr Get1Key
  lda #17
  jsr B_S
  lda #$8d
  sta Char
  sec ;quick answer
  rts
**
Stats
  lda Cursor
  asl
  tax
  jmp (:Stats,x)
:Stats dw Chr
  dw Str
  dw Agl
  dw Sta
  dw Int
  dw Qt
**
Chr
  pea #^:Text
  pea #:Text
  _WriteCString
  ldy #7 ;Change at this location

```

```

  brk
**
Sta
  pea #^:Text
  pea #:Text
  _WriteCString
  ldy #10 ;Change at this location
  ldx #200 ;Max value
  jsr Change2
  clc
  rts
:Text hex 1e
  dfb 44+32,8+32
  asc 'Sta: '
  hex 1e
  dfb 48+32,8+32
  brk
**
Int
  pea #^:Text
  pea #:Text
  _WriteCString
  ldy #11 ;Change at this location
  ldx #200 ;Max value
  jsr Change2
  clc
  rts
:Text hex 1e
  dfb 44+32,9+32
  asc 'Int: '
  hex 1e
  dfb 48+32,9+32
  brk
**

```

# Questron

## 2

- 7 DEMPH
- 8 LOOKOUT POINT
- 9 BIG OAK
- 10 GRISSOLD
- 11 SLIPPERY ROCK
- 12 DUNGEON

# Realm of Sorcerers

- 1 TWILIGHT CATHEDRAL
- 2 BRANTOWN
- 3 GREAT FORTRESS
- 4 ORCHARD LAKE
- 5 BURNSIDE
- 6 DUNGEON OF DESPAIR

```

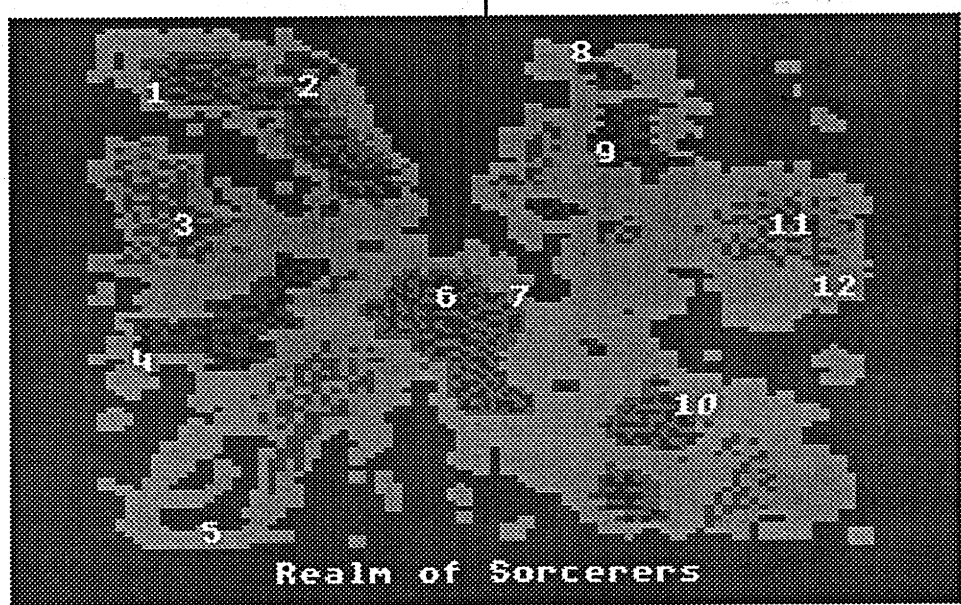
  ldx #200 ;Max value
  jsr Change2
  clc
  rts
:Text hex 1e
  dfb 44+32,5+32
  asc 'Chr: '
  hex 1e
  dfb 48+32,5+32
  brk
**
Str
  pea #^:Text
  pea #:Text
  _WriteCString
  ldy #8 ;Change at this location
  ldx #200 ;Max value
  jsr Change2
  clc
  rts
:Text hex 1e
  dfb 44+32,6+32
  asc 'Str: '
  hex 1e
  dfb 48+32,6+32
  brk
**
Agl
  pea #^:Text
  pea #:Text
  _WriteCString
  ldy #9 ;Change at this location
  ldx #200 ;Max value
  jsr Change2
  clc
  rts
:Text hex 1e
  dfb 44+32,7+32
  asc 'Agl: '
  hex 1e
  dfb 48+32,7+32

```

```

  Qt stz Screen
  sec
  rts
**
Weapons
  lda Address
  sta PNT
  lda Address+2
  sta PNT+2
  lda Screen
  asl
  tax
  lda Cursor
  cmp Table5,x
  beq :quit
  sep $20
  ldy #12
  lda [PNT],y
  cmp #$ff
  bne :skip1
  lda #0
  bra :skip2
:skip1 inc
:skip2 clc
  adc #4
  sta :Text+2
  rep $20
  pea #^:Text
  pea #:Text
  _WriteCString
  sep $30
  lda Cursor
  bne :skip3
  lda #$FF
  bra :skip4
:skip3 dec
:skip4 ldy #12
  sta [PNT],y
  lda Cursor

```



```

beq :1
clc
adc #24 ;position of weapon
dec
tay
lda [PNT],y
bne :1
inc
sta [PNT],y
:1 rep $30
jsr FixTable
clc
rts
:quit stz Screen
sec ;(update screen)
rts
:Text hex 1e
dfb 42+32,5+32
asc
brk
**
Armor
lda Address
sta PNT
lda Address+2
sta PNT+2
lda Screen
asl
tax
lda Cursor
cmp Table5,x
beq :quit
sep $20
ldy #13
lda [PNT],y
cmp #$ff
bne :skip1
lda #0
bra :skip2
:skip1 inc
:skip2 clc
adc #5
sta :Text+2
rep $20
pea #^:Text
pea #:Text
_WriteCString
sep $30
lda Cursor
bne :skip3
lda #$FF
bra :skip4
:skip3 dec
:skip4 ldy #13
sta [PNT],y
lda Cursor
beq :1
clc
adc #39 ;position of armor
dec
tay
lda [PNT],y
bne :1
inc
sta [PNT],y
:1 rep $30
jsr FixTable
clc
rts
:quit stz Screen
sec ;(update screen)
rts
:Text hex 1e
dfb 42+32,5+32
asc
brk
**
Spells
lda Cursor
asl
tax
jmp (:Spells,x)
:Spells dw MM
dw FB
dw SW
dw TS
dw Qt
MM pea #^:Text
pea #:Text

```

```

_WriteCString
ldy #19 ;Change at this location
ldx #200 ;Max value
jsr Change2
clc
rts
:Text hex 1e
dfb 42+32,5+32
asc 'MM: '
hex 1e
dfb 47+32,5+32
brk
**
FB pea #^:Text
pea #:Text
_WriteCString
ldy #20 ;Change at this location
ldx #200 ;Max value
jsr Change2
clc
rts
:Text hex 1e
dfb 44+32,6+32
asc 'FB: '
hex 1e
dfb 48+32,6+32
brk
**
SW pea #^:Text
pea #:Text
_WriteCString
ldy #21 ;Change at this location
ldx #200 ;Max value
jsr Change2
clc
rts
:Text hex 1e
dfb 44+32,7+32
asc 'SW: '
hex 1e
dfb 48+32,7+32
brk
**
TS pea #^:Text
pea #:Text
_WriteCString
ldy #22 ;Change at this location
ldx #200 ;Max value
jsr Change2
clc
rts
:Text hex 1e
dfb 44+32,8+32
asc 'TS: '
hex 1e
dfb 48+32,8+32
brk
**
Where mx $00
php
phy
phx
pea #^:Text
pea #:Text
_WriteCString
plx
ply
:done lda Address+2
jsr doWord
lda Address
jsr doWord
plp
rts
:Text hex 1e
dfb 1+32,1+32
asc 'Address: '
brk
*
doWord phx ;save
phy ;save
pha ;push word
sep #$30
ply ;pull two bytes
plx
jsr PRWORD
rep #$30
ply ;restore
plx ;restore

```

```

rts
mx %11
PRWORD TXA
JSR PRBYTE
TYA
PRBYTE PHA
LSR
LSR
LSR
LSR
JSR :DIG
PLA
AND #$F
:DIG ORA #0*
CMP #9*+1
BLT Cout
ADC #A*-9*-2
mx %00
Cout PHA
PHX
PHY
PHP
REP %00110000
PHA
_WriteChar
PLP
PLY
PLX
PLA
RTS
*
* Enter with Y-reg pointing to location to change,
* X-reg maximum value to change
*
Change lda Address
sta PNT
lda Address+2
sta PNT+2
sep $20
lda Cursor
clc
adc #5
sta :Text+2
sta :Text+12
sta :Text2+2
rep $20
:again phx
phy
pea #^:Text
pea #:Text
_WriteCString
ply
lda [PNT],y
pha
lda #0
sep $30
pla
plx
rep $30
clc
plx
jsr PrDec16
:wrong jsr Get1Key
lda Char
cmp #$8a
beq :2
cmp #$8b
beq :1
cmp #$8d
bne :wrong
phx
phy
pea #^:Text2
pea #:Text2
_WriteCString
ply
plx
jsr FixTable
clc
rts
:1 jsr Incr2
bra :again
:2 jsr Subc2
bra :again
:Text hex 1e
dfb 48+32,5+32
asc
hex 1e
dfb 48+32,5+32
brk
:Text2 hex 1e
dfb 44+32,5+32
asc
brk
**
Incr jsr CheckOA
stx Max
lda [PNT],y
bvc :No_OA
clv
clc
adc #99
:No_OA inc
cmp Max
bcc :less
beq :less
lda Max
:less sta [PNT],y
rts
Incr2 jsr CheckOA
stx Max
lda [PNT],y
bvc :No_OA
clv
clc
adc #4
:No_OA inc
sep $20
brk

```

```

* Enter with Y-reg pointing to location to change,
* X-reg maximum value to change
*
Change2 lda Address
sta PNT
lda Address+2
sta PNT+2
sep $20
lda Cursor
clc
adc #5
sta :Text+2
sta :Text+12
sta :Text2+2
rep $20
:again phx
phy
pea #^:Text
pea #:Text
_WriteCString
ply
lda [PNT],y
pha
lda #0
sep $30
pla
plx
rep $30
clc
plx
jsr PrDec16
:wrong jsr Get1Key
lda Char
cmp #$8a
beq :2
cmp #$8b
beq :1
cmp #$8d
bne :wrong
phx
phy
pea #^:Text2
pea #:Text2
_WriteCString
ply
plx
jsr FixTable
clc
rts
:1 jsr Incr2
bra :again
:2 jsr Subc2
bra :again
:Text hex 1e
dfb 48+32,5+32
asc
hex 1e
dfb 48+32,5+32
brk
:Text2 hex 1e
dfb 44+32,5+32
asc
brk
**
Incr jsr CheckOA
stx Max
lda [PNT],y
bvc :No_OA
clv
clc
adc #99
:No_OA inc
cmp Max
bcc :less
beq :less
lda Max
:less sta [PNT],y
rts
Incr2 jsr CheckOA
stx Max
lda [PNT],y
bvc :No_OA
clv
clc
adc #4
:No_OA inc
sep $20

```

```

cmp Max
bcc :less
beq :less
lda Max
:less sta [PNT],y
rep $20
rts
Subc jsr CheckOA
stx Max
lda [PNT],y
bvc :No_OA
clv
sec
sbc #99
:No_OA dec
cmp Max
bcc :less
beq :less
lda #0
:less sta [PNT],y
rts
Subc2 jsr CheckOA
stx Max
lda [PNT],y
bvc :No_OA
clv
sec
sbc #4
:No_OA dec
sep $20
cmp Max
bcc :less
beq :less
lda #0
:less sta [PNT],y
rep $20
rts
Max ds 2

```

\* PRDEC16 = 2-byte decimal output routines.  
 \* PRDEC16 enters with long A and prints A.

```

PrDec16 phx
phy
php
tax ;Entry to print 16-bit A
xba
sep $30
sta NH ;Entry to print 8-bit A,X
stx NL
lda #0 ;Left justify if CC
bcc :OV
lda #* ;Right justify if CS
:OV sta JUST
ldx #8
stx NFL
:LOAD0 ldy #'0'-1
rep %00100000 ;Long M
lda NL ;index length no matter
sec ;Needed 1st time
:DIV iny
sta NL
sbc :NUMTBL,X
bcs :DIV
sep %00110000
tya
dex
bmi :DIG
cmp #'0'
beq :MODE
sta NFL
:MODE bit NFL
bmi :DIG
lda JUST
bpl :NX
:DIG jsr Cout
:NX dex
bpl :LOAD0
plp
ply
plx
rts
:NUMTBL da 1,10,100,1000,10000
NL dfb 0
NH dfb 0
NFL dfb 0
JUST dfb 0
mx %00

```

```

**
Blank_Screen
lda Screen
asl
tax
lda Table5,x ;get total lines
inc
inc ;add 2 more
B_S tax ;store in X
clc
adc #35 ;Pascal gotoy
sep $20
sta BlankLine+2
rep $20
]lup phx
pea #*BlankLine
pea #BlankLine
_WriteCString
dec BlankLine+2
plx
dex
bne ]lup
rts

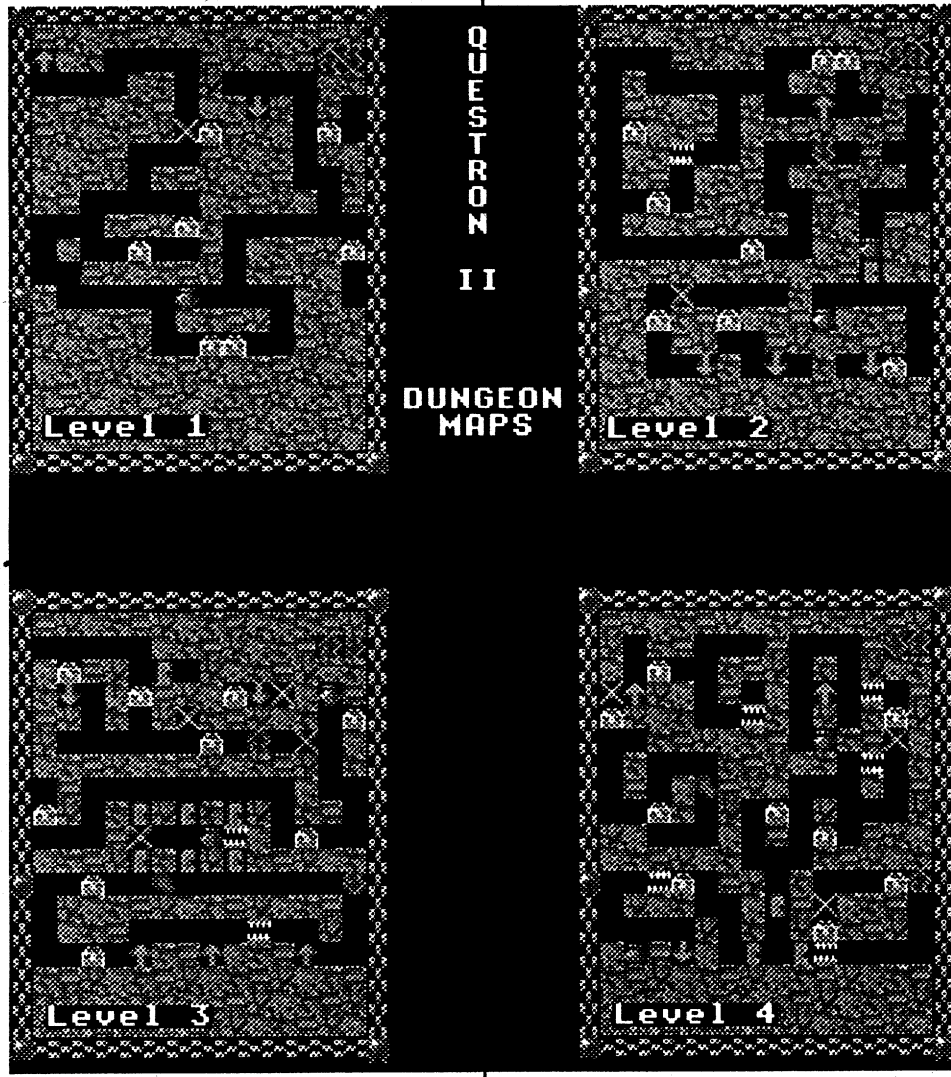
```

\* This will change the speed of the computer to fast

```

Fast_Speed
php

```



```

sep $30
lda #1
ror
ror
and #$80
sta Spd_fst
ldal $e0c036
and #$7F
ora Spd_fst
stal $e0c036
clc
plp
rts

```

\* This will copy the extra stuff into this program

```

put q2.cda.extra
typ cda
sav questron.cda.l
!#$%&'()*+,-./0123456789

```

### Data Tables used by Questron.CDA source

```

Char ds 2
Temp ds 2
Screen ds 2
Cursor ds 2

```

```

Spd_cur ds 2
Spd_fst ds 2
CurOffset ds 2
Address adrl $0
GK_Table
dw T0_C0 ;hp
dw T0_C1 ;gold
dw T0_C2 ;initialize status menu
dw T0_C3 ;initialize weapon menu
dw T0_C4 ;initialize armor menu
dw T0_C5 ;initialize spells menu
dw T0_C6 ;rope/hooks
dw T0_C7 ;about
GK_Table0
dw Stats
dw Weapons
dw Armor
dw Spells
Table adrl Table0
adrl Table1
adrl Table2
adrl Table3
adrl Table4
adrl Table5
Table0 dfb $1e,23+32,32+5 ;(10 lines)
ASC 'Change Hit Points',00
dfb $1e,23+32,32+6
ASC 'Change Gold',00

```

```

dfb $1e,23+32,32+7
ASC 'Alter Stats',00
dfb $1e,23+32,32+8
ASC 'Change Weapon',00
dfb $1e,23+32,32+9
ASC 'Change Armor',00
dfb $1e,23+32,32+10
ASC 'Change Spells',00
dfb $1e,23+32,32+11
ASC 'Toggle Rope & Hooks',00
dfb $1e,23+32,32+12
ASC 'About Info',00
dfb $1e,23+32,32+13
ASC 'Quit',00
BRK
BRK
Table1 dfb $1e,23+32,32+5 ;(6 lines)
ASC 'Charisma',00
dfb $1e,23+32,32+6
ASC 'Strength',00
dfb $1e,23+32,32+7
ASC 'Agility',00
dfb $1e,23+32,32+8
ASC 'Stamina',00
dfb $1e,23+32,32+9
ASC 'Intelligence',00
dfb $1e,23+32,32+10
ASC 'Quit',00

```

```

Table2 dfb $1e,23+32,32+4 ;(16 lines)
ASC 'None',00
dfb $1e,23+32,32+5
ASC 'Dagger',00
dfb $1e,23+32,32+6
ASC 'Hammers',00
dfb $1e,23+32,32+7
ASC 'Hatchets',00
dfb $1e,23+32,32+8
ASC 'Staves',00
dfb $1e,23+32,32+9
ASC 'Cudgles',00
dfb $1e,23+32,32+10
ASC 'Rapiers',00
dfb $1e,23+32,32+11
ASC 'Axes',00
dfb $1e,23+32,32+12
ASC 'Fauchards',00
dfb $1e,23+32,32+13
ASC 'Sabres',00
dfb $1e,23+32,32+14
ASC 'Weighted Spears',00
dfb $1e,23+32,32+15
ASC 'Pikes',00
dfb $1e,23+32,32+16
ASC 'Shortbows',00
dfb $1e,23+32,32+17
ASC 'Broadwords',00
dfb $1e,23+32,32+18
ASC 'Crossbows',00
dfb $1e,23+32,32+19
ASC 'Quit',00
BRK
BRK
Table3 dfb $1e,23+32,32+5 ;(9 lines)
ASC 'None',00
dfb $1e,23+32,32+6
ASC 'Rawhide',00
dfb $1e,23+32,32+7
ASC 'Studded Leather',00
dfb $1e,23+32,32+8
ASC 'Ring Mail',00
dfb $1e,23+32,32+9
ASC 'Bar Mail',00
dfb $1e,23+32,32+10
ASC 'Chain Mail',00
dfb $1e,23+32,32+11
ASC 'Plate Mail',00
dfb $1e,23+32,32+12
ASC 'Ribbed Mail',00
dfb $1e,23+32,32+13
ASC 'Quit',00
BRK
BRK
Table4 dfb $1e,23+32,32+5 ;(5 lines)
ASC 'Magic Missiles',00
dfb $1e,23+32,32+6
ASC 'Fireballs',00
dfb $1e,23+32,32+7
ASC 'Sonic Whines',00
dfb $1e,23+32,32+8
ASC 'Time Saps',00
dfb $1e,23+32,32+9
ASC 'Quit',00
BRK
BRK
Table5 da 8 ;total lines in each table
da 5
da 15
da 8
da 4
BlankLine
hex 1e ;Pascal GOTOTOX
dfb 21+32 ;X = column 20
dfb 0+32 ;Y = row
lup 38
asc ''
^
hex 0E ;inverse off
hex 18 ;mouse text off
brk
WaitLine
hex 1e ;Pascal GOTOTOX
dfb 23+32 ;X = column 20
dfb 7+32 ;Y = row

```

```

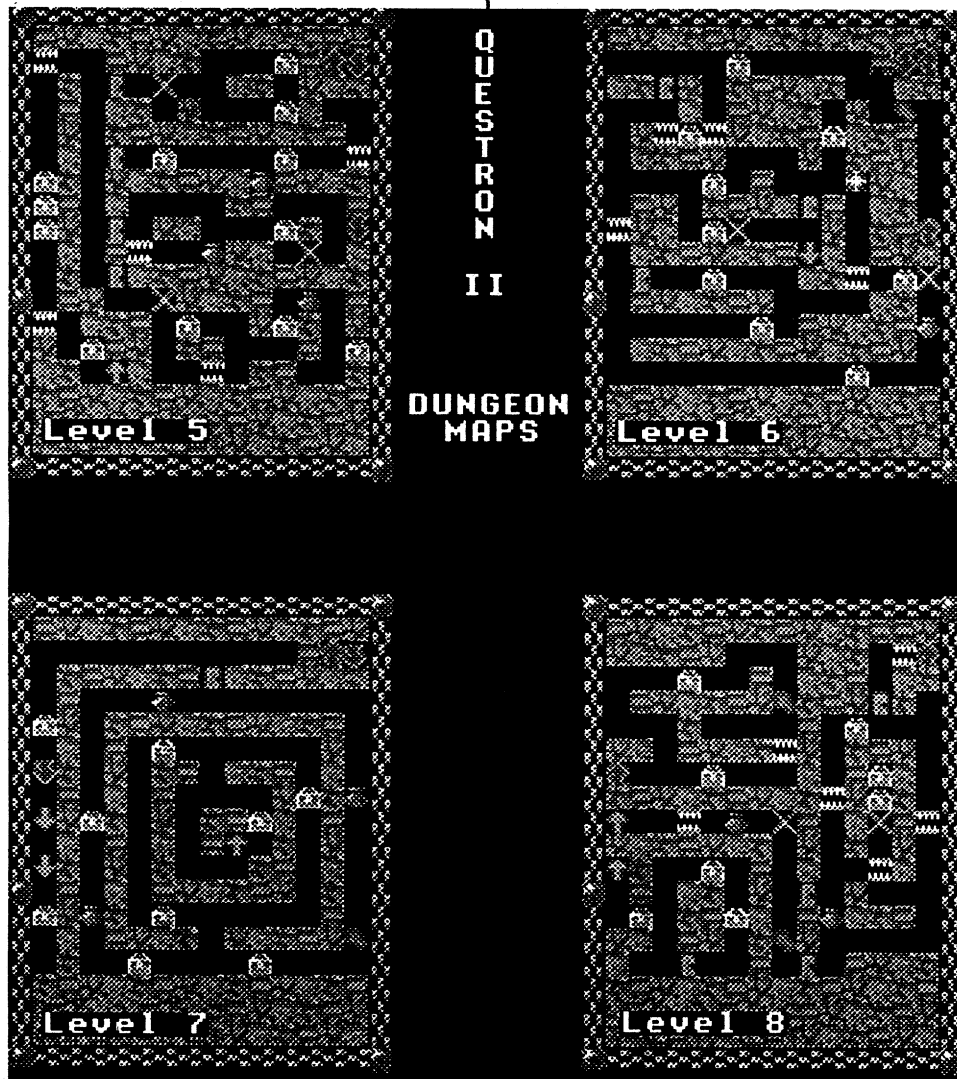
asc 'Scanning memory for the game
    data'
hex 0E ;inverse off
hex 18 ;mouse text off
brk
NotFoundLine
hex 1e ;Pascal GOTOXY
dfb 23+32 ;X = column 20
dfb 7+32 ;Y = row
asc ' Questron II is not in MEMORY '
hex 0E ;inverse off
hex 18 ;mouse text off
brk
Block0
Line1 hex 1e ;Pascal GOTOXY
dfb 21+32 ;X = column 21
dfb 0+32 ;Y = row 0
lup 38
asc ' :38 under-bars
    _^
Line2 hex 1e ;Pascal GOTOXY
dfb 20+32 ;X = column 20
dfb 1+32 ;Y = row 1
hex 0F ;inverse on
asc 'Z';draw |
hex 0E ;inverse off
asc ' ';space
hex 0F ;inverse on
asc 'A';draw open apple
hex 0E ;inverse off
asc ' Questron II CDA Cheat v2.0 '
hex 0F ;inverse on
lup 8
asc ' :8 spaces
    _^
Line3 asc ' ';draw |
hex 1e ;Pascal GOTOXY
dfb 20+32 ;X = column 20
dfb 2+32 ;Y = row 2
asc 'Z';draw |
lup 38
asc 'L';draw under-bars
    _^
asc ' ';draw |
Block0End
Block1
jX = 20+32 ;Column = 20
jY = 3+32-1 ;Starting position = row 3
Line4_21 lup 19
jY = jY+1 ;Increase row
hex 1e ;Pascal GOTOXY (+32 to
    XY)
dfb jX ;X = column 20
dfb jY ;Y = row (changing)
asc 'Z';Draw |
hex 1e ;Pascal GOTOXY
dfb jX+39 ;X = column + 39 = 59
dfb jY ;Y = row (changing)
asc ' ';Draw |
    _^
Block1End
Block2
Line22 hex 1e ;Pascal GOTOXY
dfb 20+32 ;X = column 21
dfb 22+32 ;Y = row 0
asc 'Z';draw |
hex 0e ;inverse off
asc 'Select:'
hex 0f ;inverse on
asc 'J';draw up-arrows
hex 0e ;inverse off
asc ' '
hex 0f ;inverse on
asc 'K';draw up-arrows
hex 0e ;inverse off
hex 1e ;Pascal GOTOXY
dfb 50+32 ;X = column 30
dfb 22+32 ;Y = row 22
asc 'Open:'
hex 0f ;inverse on
asc 'M';draw RETURN
hex 0e ;inverse off
asc ' '
hex 0f ;inverse on
asc ' ';draw |
Line23 hex 1e ;Pascal GOTOXY
dfb 21+32 ;X = column 21
dfb 23+32 ;Y = row 0

```

```

lup 38
asc 'L';draw under-bars
    _^
hex 0e ;inverse off
Block2End
About
hex 1e ;Pascal GOTOXY
dfb 28+32 ;X = column 20
dfb 4+32 ;Y = row
asc 'Written by Vince Andrews'
hex 1e ;Pascal GOTOXY
dfb 26+32 ;X = column 20
dfb 6+32 ;Y = row
asc 'Member of Computist Magazine'
hex 1e ;Pascal GOTOXY
dfb 23+32 ;X = column 20
dfb 7+32 ;Y = row
asc 'PO Box 242, Kapowsin WA 98344-
    0242'
hex 1e ;Pascal GOTOXY
dfb 22+32 ;X = column 20
dfb 10+32 ;Y = row
asc '12/16/91 v1.0 - Started this
    program'
hex 1e ;Pascal GOTOXY
dfb 22+32 ;X = column 20
dfb 12+32 ;Y = row
asc '12/21/91 v1.1 - Fixed the search
    bug'

```



```

hex 1e ;Pascal GOTOXY
dfb 22+32 ;X = column 20
dfb 14+32 ;Y = row
asc '12/23/91 v1.2 - Fixed the Weapons
    &'
hex 1e ;Pascal GOTOXY
dfb 38+32 ;X = column 20
dfb 15+32 ;Y = row
asc 'Armor routines'
hex 1e ;Pascal GOTOXY
dfb 22+32 ;X = column 20
dfb 17+32 ;Y = row
asc '12/30/91 v2.0 - Rewrote the
    search'
hex 1e ;Pascal GOTOXY
dfb 38+32 ;X = column 20
dfb 18+32 ;Y = row
asc 'routines (faster!)'
hex 0E ;inverse off
hex 18 ;mouse text off
About_End
Bytes2Scan
hex 68,fa,85,14,86,16,00

```

## Questron.CHT Source

```

* Questron CDA Installer v1.0
* By Vince C. Andrews, 19 Jan 92
* Merlin-16+ Assembler
* Use OA-6 to Assemble
*
REL
LST off ;Turn listing on/off here
XC ;65C02 mode
XC ;65816 mode
MX %00 ;Full gs mode
*
* Equates and Macros used:
KEY = $E0C000
STROBE = $E0C010
Tool MAC ;Macro
LDX #j1
JSL $E10000
FIN
<<<
dl MAC ;Macro
ADRL j1
<<<
WriteChr MAC ;Macro
Tool $180C
<<<
PullLong MAC ;Macro
PLA
STA j1

```

```

PLA
STA j1+2
<<<
PushLong MAC ;Macro
IF #=j1
PushWord #^j1
ELSE
PushWord j1+2
FIN
PushWord j1
<<<
PushWord MAC ;Macro
IF #=j1
PEA j1
ELSE
IF MX/2
LDA j1+1
PHA
FIN
LDA j1
PHA
FIN
<<<
PRINT MAC ;Macro
JSR SENDMSG
ASC j1,00
<<<

```

```

_SetOutputDevice MAC
Tool $100C
<<<
_InitTextDevice MAC
Tool $150C
<<<
*
* Start of the program:
*
brl Start
asc 'Vince Andrews 15 Jan 92'
Start PHK
PLB
rep $30
pea 1 ;Pascal device type
pea 0 ;in slot 3
pea 3
_SetOutputDevice
pea 1 ;initialize standard output
_InitTextDevice
pea $06 ;turn off cursor
WriteChr
PushWord #0 ;space
PushWord #0 ;space
PushLong #0 ;space
PushWord #0 ;space
PHA ;room for result
LDA #%0101_0000_00000000 ;new ID;
type = $05 DESK
ACCESSORY
PHA
Tool $2003 ;_GetNewID
;leave user ID on stack
PushLong #FILENAME ;point to
    filename
PushWord #0 ;special-memory flag
Tool $911 ;_InitialLoad
BCC :readok
BRL MLlerr ;bomb if loader error
:readok
PLA
STA ourID
PullLong ptr ;get pointer to DA
PLA ;dump direct page/stack addr
PLA ;dump direct page/stack size
PushLong #0 ;space
PushLong ptr ;give pointer to DA
Tool $1a02 ;_FindHandle - ptr is
    already on stack
PullLong handle ;and we end up with
    a handle!
LDA handle+2 ;check handle
ORA handle+4
BNE :okay
PRINT "ILLEGAL HANDLE! DA
    not installed."8a8d
BRL EXIT
:okay PushLong handle ;pass handle to da
:CDA Tool $F05 ;_InstallCDA (doesn't
    return any errors)
PRINT 8a8d"You may now launch
    the Q2.SYS16 file"
EXIT PRINT 8a8d8a8d"Press
    RETURN."
JLUP SEP $30
LDAL KEY ;Else wait for key
BPL JLUP
XBA
LDAL STROBE
XBA
CMP #$8D
BNE JLUP
PRINT 8D8D
REP #$30
LDA #0 ;No error
:1 JSL $e100a8
DA $29
ADRL QParms
BCS :1
QParms ADRL $0000
DA $00
RTL ;All done
*
* MLI Error routine:
*
MLlerr PHA
PRINT 8a8d"Questron.CDA was
    not found"

```

```

PLA ;remove error code
BRI EXIT
mx %00

*JSR Routines:

SENDMSG REP %00100001
LDA 1,S
INC
STA 1,S
SEP %00110000
LDY #0
LDA (1,S),Y
BEQ :back
JSR OUTPUT
BRA SENDMSG
:back REP #30 ;back to full 16 bit
RTS

OUTPUT PHA
PHX
PHY
PHP
REP %00110000
PHA
WriteChr
PLP
PLY
PLX
PLA

BACK RTS
PrFile MX %11
PHP
SEP #30
LDY #3
JLUP LDA FILENAME,Y
BEQ :rts
JSR OUTPUT ;Print filename
INY
BRA JLUP
:rts PLP
RTS

```

```

* Storage area:

handle dl 0
ptr dl 0
ourID dw 0
FILENAME str '1/QUESTRON.CDA'
SAV QUESTRON.CHT.L

```

**QUESTRON.CDA**

```

2000:D1 11 00 00 00 00 00 $39DB
2008:6C 0F 00 00 00 0A 04 02 $4EAA
2010:00 00 01 00 00 10 00 00 $4CB7
2018:00 00 00 00 00 00 00 00 $0C07
2020:00 00 01 00 00 00 00 00 $6EA2
2028:2C 00 40 00 51 55 45 53 $76EB
2030:54 52 4F 4E 2E 43 51 55 $EE76
2038:45 53 54 52 4F 4E 2E 43 $D69C
2040:F2 6C 0F 00 00 0B 51 75 $D889
2048:65 73 74 72 6F 6E 20 49 $61E8

2050:49 14 00 00 00 77 01 00 $0A47
2058:00 8B 4B AB D4 E0 D4 E2 $B163
2060:64 E2 F4 01 00 F4 00 00 $EC1C
2068:F4 03 00 A2 0C 10 22 00 $4E90
2070:00 E1 F4 01 00 A2 0C 15 $B36D
2078:22 00 00 E1 F4 92 00 A2 $AFA9
2080:0C 18 22 00 00 E1 F4 06 $A972
2088:00 A2 0C 18 22 00 00 E1 $239B
2090:20 AA 09 9C CA 09 20 78 $5507
2098:01 AD D4 09 F0 03 4C FB $93CC

20A0:00 F4 00 00 F4 8B 0C A2 $C008
20A8:0C 20 22 00 00 E1 A2 01 $F68A
20B0:00 A0 00 10 64 E0 86 E2 $F185
20B8:C8 F0 12 C8 F0 0F C0 FD $AC0C
20C0:BF F0 05 C0 FE 8F D0 23 $6634
20C8:E0 01 00 D0 1E E8 86 E2 $B935
20D0:E0 10 00 D0 E3 F4 00 00 $347A
20D8:F4 B2 0C A2 0C 20 22 00 $6158
20E0:00 E1 20 F0 02 9C D4 09 $2E9C
20E8:4C 6D 01 B7 E0 CD 65 0F $4A2C

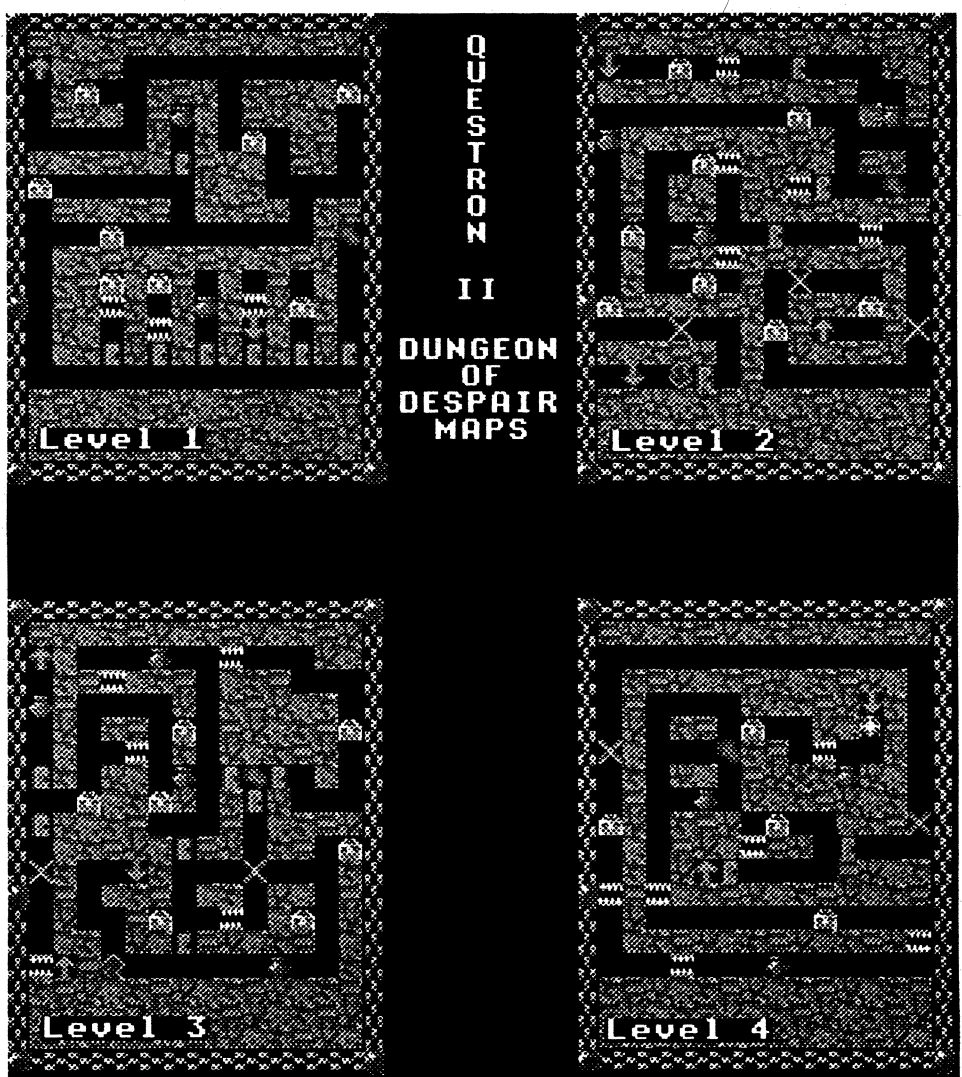
20F0:F0 06 CD 66 0F D0 C1 88 $F9B9
20F8:B7 E0 CD 65 0F D0 B9 C8 $52AB
2100:C8 B7 E0 CD 67 0F D0 B0 $6CA8
2108:C8 C8 B7 E0 CD 69 0F D0 $9BCB
2110:A7 C8 C8 5A DA B7 E0 48 $BA56

```

```

2118:A9 00 00 E2 30 68 FA C2 $DFFD
2120:30 FA 7A C9 AF 00 D0 15 $2901
2128:98 38 E9 0F 00 A8 B7 E0 $6DD0
2130:8D D4 09 C8 C8 C8 B7 E0 $B471
2138:8D D6 09 80 03 4C 73 00 $C676
2140:A9 00 00 E2 20 AF 61 C0 $AFEA
2148:E0 10 05 C2 20 20 29 07 $1C9C
2150:E2 20 A9 07 8D 61 0C C2 $D76C
2158:30 F4 00 00 F4 5F 0C A2 $0C6A
2160:0C 20 22 00 00 E1 9C C8 $5012
2168:09 9C CA 09 A9 00 00 AE $9E09
2170:CA 09 E0 00 00 D0 03 AD $C042
2178:C8 09 8D CC 09 9C D2 09 $D546
2180:20 C6 01 20 D7 01 20 C6 $BBEC
2188:01 20 03 02 20 46 02 20 $77F2
2190:C6 01 20 B7 02 90 EF AD $8706
2198:C6 09 C9 8D 00 F0 CD C9 $5FDA
21A0:00 00 F0 0E AD CA 09 0A $2BA9
21A8:AA BD 55 0C 8D CC 09 18 $23D4
21B0:80 D4 18 C2 30 AB 68 85 $89DA
21B8:E2 68 85 E0 6B F4 9B 00 $4A4B
21C0:A2 0C 18 22 00 00 E1 F4 $631B
21C8:00 00 F4 D9 0C F4 00 00 $AAE9
21D0:F4 84 00 A2 0C 1E 22 00 $81FF
21D8:00 E1 F4 00 00 F4 5D 0D $1360
21E0:F4 00 00 F4 98 00 A2 0C $A816
21E8:1E 22 00 00 E1 F4 00 00 $FF20
21F0:F4 F5 0D F4 00 00 F4 4F $69E0

```



```

21F8:00 A2 0C 1E 22 00 00 E1 $8596
2200:F4 98 00 A2 0C 18 22 00 $0651
2208:00 E1 60 AD CA 09 0A 0A $E164
2210:AA BD F0 09 85 E0 BD F2 $F4F5
2218:09 85 E2 60 C2 30 A5 E2 $2FE2
2220:48 A5 E0 18 6D D2 09 48 $525B
2228:A2 0C 20 22 00 00 E1 E2 $31B8
2230:30 AC D2 09 B1 E0 F0 03 $7624
2238:C8 80 F9 C8 B1 E0 F0 05 $C97A
2240:8C D2 09 80 D7 C2 30 60 $B606
2248:AD CC 09 AA A0 00 00 E2 $FE0C
2250:30 E0 00 F0 0B B1 E0 F0 $3A2B
2258:03 C8 80 F9 C8 CA 80 EF $D54C
2260:C2 30 8C D2 09 F4 0F 00 $35BD
2268:A2 0C 18 22 00 00 E1 A5 $7DA5
2270:E2 48 A5 E0 18 6D D2 09 $F1CA
2278:48 A2 0C 20 22 00 00 E1 $270F
2280:F4 0E 00 A2 0C 18 22 00 $0423
2288:00 E1 60 AD D4 09 85 E0 $3334
2290:AD D6 09 85 E2 AD CA 09 $2A0B
2298:C9 02 00 F0 06 C9 03 00 $438E
22A0:F0 27 60 E2 20 A0 0C 00 $A6F6
22A8:B7 E0 C9 FF D0 04 A9 00 $BE72
22B0:80 01 1A 18 69 04 8D AC $86C2
22B8:02 C2 20 F4 00 00 F4 AA $2B60
22C0:02 A2 0C 20 22 00 00 E1 $E805
22C8:60 E2 20 A0 0D 00 B7 E0 $4BF6

```

```

22D0:C9 FF D0 04 A9 00 80 01 $EF75
22D8:1A 18 69 05 8D AC 02 C2 $842E
22E0:20 F4 00 00 F4 AA 02 A2 $E996
22E8:0C 20 22 00 00 E1 60 1E $F539
22F0:4A 25 52 65 61 64 69 65 $22FD
22F8:64 00 00 00 20 F0 02 AD $A821
2300:C6 09 C9 8A 00 F0 49 C9 $F97E
2308:8B 00 F0 5E C9 8D 00 F0 $C3DD
2310:73 C9 9B 00 D0 E6 18 20 $B8C2
2318:2A 02 38 60 48 E2 20 B8 $9512
2320:9C B5 02 AF 61 C0 E0 10 $A8B6
2328:08 A9 60 8D B5 02 2C B5 $3D90
2330:02 C2 30 68 60 E2 20 AF $8B77
2338:00 C0 E0 10 F8 8F 10 C0 $82A6
2340:E0 C2 20 C9 E0 00 90 03 $44FA
2348:29 DF 00 8D C6 09 18 60 $FEE6
2350:20 2A 02 AD CA 09 0A AA $1926
2358:AD CC 09 DD 55 0C F0 05 $178B
2360:EE CC 09 18 60 9C CC 09 $6922
2368:18 60 20 2A 02 AD CC 09 $B8CF
2370:D0 0D AD CA 09 0A AA BD $3723
2378:55 0C 8D CC 09 80 03 CE $9022
2380:CC 09 18 60 AD CA 09 0A $B9C4
2388:AA AD CC 09 DD 55 0C D0 $C3D1
2390:12 AD CA 09 D0 05 9C C6 $3E94
2398:09 80 03 20 7E 09 9C CA $63B5
23A0:09 38 60 AD CA 09 F0 09 $8BE4
23A8:AD CA 09 3A 0A AA 7C E8 $7605

2488:C2 20 F4 00 00 F4 76 04 $1C78
2490:A2 0C 20 22 00 00 E1 80 $FD5A
2498:C2 E2 20 A9 01 A0 45 00 $D167
24A0:97 E0 C2 20 F4 00 00 F4 $6DEE
24A8:6E 04 A2 0C 20 22 00 00 $6F05
24B0:E1 80 A8 1E 4C 2B 59 65 $4D77
24B8:73 00 00 1E 4C 2B 4E 6F $6D5F
24C0:20 00 00 1E 4C 2B 20 20 $906F
24C8:20 00 A9 11 00 20 88 09 $EE1E
24D0:F4 00 00 F4 44 0E F4 00 $40B3
24D8:00 F4 21 01 A2 0C 1E 22 $C3BD
24E0:00 00 E1 20 F0 02 A9 11 $C66C
24E8:00 20 88 09 A9 8D 00 8D $4B57
24F0:C6 09 38 60 AD CC 09 0A $8F8E
24F8:AA 7C B7 04 C3 04 EB 04 $FDB7
2500:13 05 3B 05 63 05 8B 05 $B577
2508:F4 00 00 F4 DB 04 A2 0C $C3B6
2510:20 22 00 00 E1 A0 07 00 $9605
2518:A2 C8 00 20 16 08 18 60 $3C85
2520:1E 4C 25 43 68 72 3A 20 $8148
2528:20 20 20 20 1E 50 25 00 $2AFD
2530:F4 00 00 F4 03 05 A2 0C $75FE
2538:20 22 00 00 E1 A0 08 00 $D9C2
2540:A2 C8 00 20 16 08 18 60 $C372
2548:1E 4C 26 53 74 72 3A 20 $7B60
2550:20 20 20 20 1E 50 26 00 $F756
2558:F4 00 00 F4 2B 05 A2 0C $BC05
2560:20 22 00 00 E1 A0 09 00 $42C8
2568:A2 C8 00 20 16 08 18 60 $28D8
2570:1E 4C 27 41 67 6C 3A 20 $3DBC
2578:20 20 20 20 1E 50 27 00 $534B
2580:F4 00 00 F4 53 05 A2 0C $D468
2588:20 22 00 00 E1 A0 0A 00 $2D36
2590:A2 C8 00 20 16 08 18 60 $77C6
2598:1E 4C 28 53 74 61 3A 20 $7555
25A0:20 20 20 20 1E 50 28 00 $027D
25A8:F4 00 00 F4 7B 05 A2 0C $D19E
25B0:20 22 00 00 E1 A0 0B 00 $3AA1
25B8:A2 C8 00 20 16 08 18 60 $5061
25C0:1E 4C 29 49 6E 74 3A 20 $4676
25C8:20 20 20 20 1E 50 29 00 $43AF
25D0:9C CA 09 38 60 AD D4 09 $CD07
25D8:85 E0 AD D6 09 85 E2 AD $CCC1
25E0:CA 09 0A AA AD CC 09 DD $2AA0
25E8:55 0C F0 4D E2 20 A0 0C $D84F
25F0:00 B7 E0 C9 FF D0 04 A9 $BC4F
25F8:00 80 01 1A 18 69 04 8D $BA07
2600:FB 05 C2 20 F4 00 00 F4 $F1FE
2608:F9 05 A2 0C 20 22 00 00 $D9E6
2610:E1 E2 30 AD CC 09 D0 04 $EF54
2618:A9 FF 80 01 3A A0 0C 97 $7F9F
2620:E0 AD CC 09 F0 0C 18 69 $8819
2628:18 3A A8 B7 E0 D0 03 1A $FEA5
2630:97 E0 C2 30 20 C6 01 18 $F66C
2638:60 9C CA 09 38 60 1E 4A $9B9F
2640:25 20 20 20 20 20 20 20 $C9FE
2648:00 AD D4 09 85 E0 AD D6 $A721
2650:09 85 E2 AD CA 09 0A AA $65A5
2658:AD CC 09 DD 55 0C F0 4D $63DC
2660:E2 20 A0 0D 00 B7 E0 C9 $9F6B
2668:FF D0 04 A9 00 80 01 1A $67E6
2670:18 69 05 8D 6F 06 C2 20 $DDD7
2678:F4 00 00 F4 6D 06 A2 0C $A4CB
2680:20 22 00 00 E1 E2 30 AD $11AC
2688:CC 09 D0 04 A9 FF 80 01 $CAD4
2690:3A A0 0D 97 E0 AD CC 09 $B206
2698:F0 0C 18 69 27 3A A8 B7 $E733
26A0:E0 D0 03 1A 97 E0 C2 30 $E442
26A8:20 C6 01 18 60 9C CA 09 $76A0
26B0:38 60 1E 4A 25 20 20 20 $1D7A
26B8:20 20 20 20 00 AD CC 09 $EF7E
26C0:0A AA 7C 80 06 8A 06 B4 $38B2
26C8:06 DB 06 02 07 8B 05 F4 $AD0F
26D0:00 00 F4 A2 06 A2 0C 20 $EC1F
26D8:22 00 00 E1 A0 13 00 A2 $E7E9
26E0:C8 00 20 16 08 18 60 1E $4F7D
26E8:4A 25 20 20 20 4D 4D 3A $4861
26F0:20 20 20 20 20 1E 4F 25 $DCBF
26F8:00 F4 00 00 F4 CC 06 A2 $33AC
2700:0C 20 22 00 00 E1 A0 14 $E5E6
2708:00 A2 C8 00 20 16 08 18 $556C
2710:60 1E 4C 26 46 42 3A 20 $AB67
2718:20 20 20 20 1E 50 26 00 $A7E1
2720:F4 00 00 F4 F3 06 A2 0C $7D65
2728:20 22 00 00 E1 A0 15 00 $E534
2730:A2 C8 00 20 16 08 18 60 $3F64

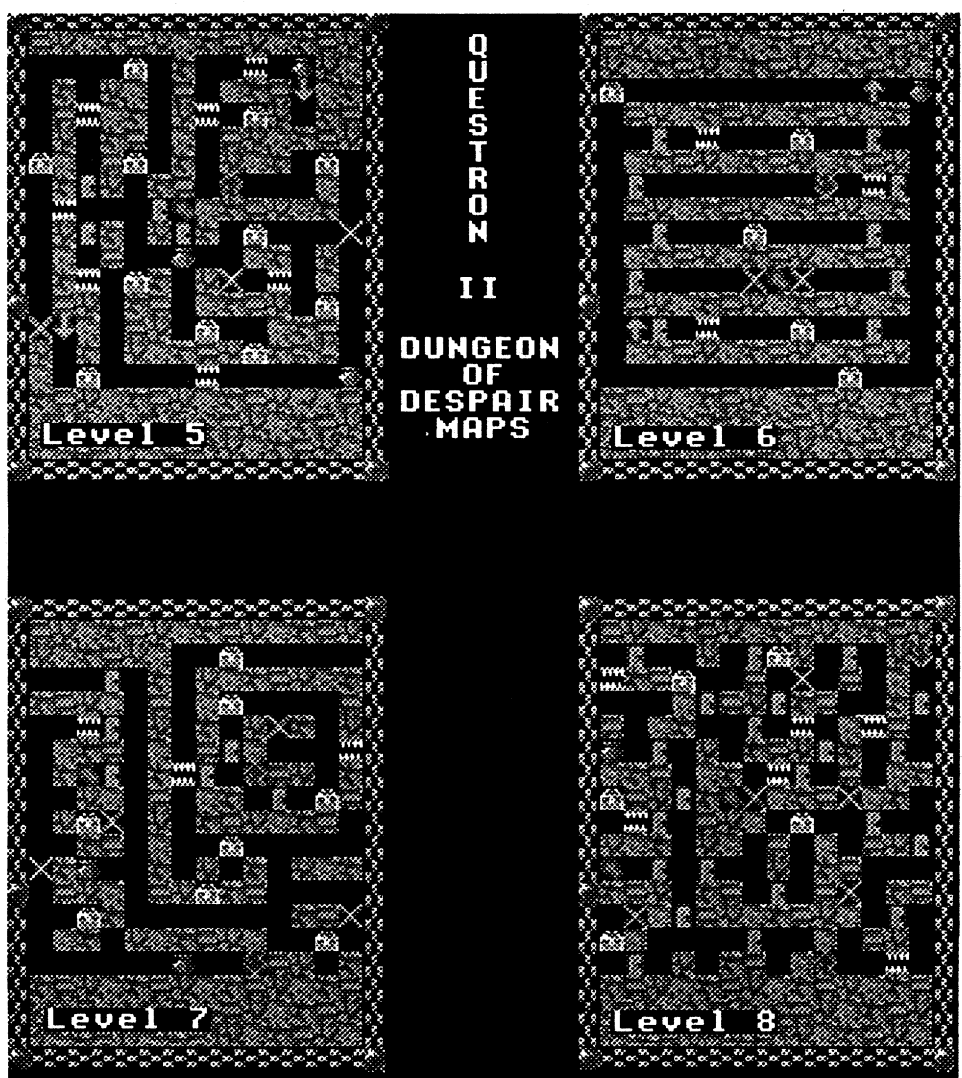
```

2738:1E 4C 27 53 57 3A 20 20 \$31A0  
 2740:20 20 20 1E 50 27 00 F4 \$FEB4  
 2748:00 00 F4 1A 07 A2 0C 20 \$FFBD  
 2750:22 00 00 E1 A0 16 00 A2 \$1163  
 2758:C8 00 20 16 08 18 60 1E \$59D7  
 2760:4C 28 54 53 3A 20 20 20 \$EB3A  
 2768:20 20 1E 50 28 00 08 5A \$AEF9  
 2770:DA F4 00 00 F4 49 07 A2 \$0BC3  
 2778:0C 20 22 00 00 E1 FA 7A \$2AD4  
 2780:AD D6 09 20 56 07 AD D4 \$0881  
 2788:09 20 56 07 28 60 1E 21 \$72CF  
 2790:21 41 64 64 72 65 73 73 \$5128  
 2798:3A 20 00 DA 5A 48 E2 30 \$1A97  
 27A0:7A FA 20 65 07 C2 30 7A \$8B4D  
 27A8:FA 60 8A 20 6A 07 98 48 \$7B8E  
 27B0:4A 4A 4A 4A 20 75 07 68 \$81B3  
 27B8:29 0F 09 B0 C9 BA 90 02 \$BCBF  
 27C0:69 06 48 DA 5A 08 C2 30 \$2A4A  
 27C8:48 A2 0C 18 22 00 00 E1 \$34A3  
 27D0:28 7A FA 68 60 AD D4 09 \$AB5C  
 27D8:85 E0 AD D6 09 85 E2 E2 \$85CD  
 27E0:20 AD CC 09 18 69 05 8D \$93DC  
 27E8:FB 07 8D 05 08 8D 09 08 \$7596  
 27F0:C2 30 DA 5A F4 00 00 F4 \$95F7  
 27F8:F9 07 A2 0C 20 22 00 00 \$FD6E  
 2800:E1 7A FA B7 E0 18 20 23 \$902A  
 2808:09 20 F0 02 AD C6 09 C9 \$2F99  
 2810:8A 00 F0 25 C9 8B 00 F0 \$22C5  
 2818:1B C9 8D 00 D0 EB DA 5A \$AC4F  
 2820:F4 00 00 F4 07 08 A2 0C \$0CE0  
 2828:20 22 00 00 E1 7A FA 20 \$A507  
 2830:C6 01 18 60 20 A6 08 80 \$072A  
 2838:B9 20 E4 08 80 B4 1E 50 \$023F  
 2840:25 20 20 20 20 20 20 20 \$80DE  
 2848:1E 50 25 00 1E 4C 25 20 \$A414  
 2850:20 20 20 20 20 20 20 20 \$64D4  
 2858:20 20 00 AD D4 09 85 E0 \$A251  
 2860:AD D6 09 85 E2 E2 20 AD \$53FF  
 2868:CC 09 18 69 05 8D 8B 08 \$89DB  
 2870:8D 95 08 8D 99 08 C2 20 \$E22F  
 2878:DA 5A F4 00 00 F4 89 08 \$6B89  
 2880:A2 0C 20 22 00 00 E1 7A \$4006  
 2888:B7 E0 48 A9 00 00 E2 30 \$3739  
 2890:68 FA C2 30 18 FA 20 23 \$9546  
 2898:09 20 F0 02 AD C6 09 C9 \$3AB5  
 28A0:8A 00 F0 25 C9 8B 00 F0 \$27A9  
 28A8:1B C9 8D 00 D0 EB DA 5A \$B963  
 28B0:F4 00 00 F4 97 08 A2 0C \$E1CC  
 28B8:20 22 00 00 E1 7A FA 20 \$D86B  
 28C0:C6 01 18 60 20 C3 08 80 \$7FFE  
 28C8:AF 20 01 09 80 AA 1E 50 \$1CA1  
 28D0:25 20 20 20 20 20 20 20 \$7E20  
 28D8:1E 50 25 00 1E 4C 25 20 \$3A8A  
 28E0:20 20 20 20 20 20 20 20 \$9A2A  
 28E8:20 20 00 20 D7 02 8E 21 \$878C  
 28F0:09 B7 E0 50 05 B8 18 69 \$9569  
 28F8:63 00 1A CD 21 09 90 05 \$70C5  
 2900:F0 03 AD 21 09 97 E0 60 \$A917  
 2908:20 D7 02 8E 21 09 B7 E0 \$AFC3  
 2910:50 05 B8 18 69 04 00 1A \$4455  
 2918:E2 20 CD 21 09 90 05 F0 \$DOA4  
 2920:03 AD 21 09 97 E0 C2 20 \$65F7  
 2928:60 20 D7 02 8E 21 09 B7 \$D159  
 2930:E0 50 05 B8 38 E9 63 00 \$3739  
 2938:3A CD 21 09 90 05 F0 03 \$129D  
 2940:A9 00 00 97 E0 60 20 D7 \$F2B8  
 2948:02 8E 21 09 B7 E0 50 05 \$AE19  
 2950:B8 38 E9 04 00 3A E2 20 \$FDD9  
 2958:CD 21 09 90 04 F0 02 A9 \$0A41  
 2960:00 97 E0 C2 20 60 00 00 \$396D  
 2968:DA 5A 08 AA EB E2 30 8D \$DEF7  
 2970:7B 09 8E 7A 09 A9 00 90 \$06EC  
 2978:02 A9 A0 8D 7D 09 A2 08 \$F949  
 2980:8E 7C 09 A0 AF C2 20 AD \$1E64  
 2988:7A 09 38 C8 8D 7A 09 FD \$2345  
 2990:70 09 B0 F7 E2 30 98 CA \$3221  
 2998:30 11 C9 B0 F0 03 8D 7C \$584A  
 29A0:09 2C 7C 09 30 05 AD 7D \$4376  
 29A8:09 10 03 20 7D 07 CA 10 \$F692  
 29B0:D2 28 7A FA 60 01 00 0A \$A7E2  
 29B8:00 64 00 E8 03 10 27 00 \$7FB6  
 29C0:00 00 00 AD CA 09 0A AA \$09D1  
 29C8:BD 55 0C 1A 1A AA 18 69 \$FE18  
 29D0:23 00 E2 20 8D 61 0C C2 \$3171  
 29D8:20 DA F4 00 00 F4 5F 0C \$6623  
 29E0:A2 0C 20 22 00 00 E1 CE \$6956

29E8:61 0C FA CA D0 EB 60 08 \$93FD  
 29F0:E2 30 A9 01 6A 6A 29 80 \$3EE0  
 29F8:8D D0 09 AF 36 C0 E0 29 \$1B7B  
 2A00:7F 0D D0 09 8F 36 C0 E0 \$BBE8  
 2A08:18 28 60 00 00 00 00 00 \$37DC  
 2A10:00 00 00 00 00 00 00 00 \$879C  
 2A18:00 00 00 00 00 80 03 A9 \$994B  
 2A20:03 D2 03 E0 03 EE 03 FC \$558F  
 2A28:03 0A 04 85 04 AF 04 90 \$EA88  
 2A30:05 04 06 78 06 08 0A 00 \$ED97  
 2A38:00 9C 0A 00 00 E4 0A 00 \$6993  
 2A40:00 9B 0B 00 00 0F 0C 00 \$FCB6  
 2A48:00 55 0C 00 00 1E 37 25 \$A3A4  
 2A50:43 68 61 6E 67 65 20 48 \$744E  
 2A58:69 74 20 50 6F 69 6E 74 \$2942  
 2A60:73 00 1E 37 26 43 68 61 \$0BE5  
 2A68:6E 67 65 20 47 6F 6C 64 \$9DDF  
 2A70:00 1E 37 27 41 6C 74 65 \$B8D9  
 2A78:72 20 53 74 61 74 73 00 \$49A8  
 2A80:1E 37 28 43 68 61 6E 67 \$F97B  
 2A88:65 20 57 65 61 70 6F 6E \$4C3E  
 2A90:00 1E 37 29 43 68 61 6E \$7991  
 2A98:67 65 20 41 72 6D 6F 72 \$1DDA  
 2AA0:00 1E 37 2A 43 68 61 6E \$4FA8  
 2AA8:67 65 20 53 70 65 6C 6C \$559E  
 2AB0:73 00 1E 37 2B 54 6F 67 \$97ED  
 2AB8:67 6C 65 20 52 6F 70 65 \$6978  
 2AC0:20 26 20 48 6F 6F 6B 73 \$4F49

2BA0:61 72 73 00 1E 37 2F 50 \$D271  
 2BA8:69 6B 65 73 00 1E 37 30 \$1D1F  
 2BB0:53 68 6F 72 74 62 6F 77 \$E7AD  
 2BB8:73 00 1E 37 31 42 72 6F \$8878  
 2BC0:61 64 73 77 6F 72 64 73 \$A22E  
 2BC8:00 1E 37 32 43 72 6F 73 \$3659  
 2BD0:73 62 6F 77 73 00 1E 37 \$5805  
 2BD8:33 51 75 69 74 00 00 00 \$F713  
 2BE0:1E 37 25 4E 6F 6E 65 00 \$C423  
 2BE8:1E 37 26 52 61 77 68 69 \$D34D  
 2BF0:64 65 00 1E 37 27 53 74 \$37C7  
 2BF8:75 64 64 65 64 20 4C 65 \$BF7C  
 2C00:61 74 68 65 72 00 1E 37 \$3380  
 2C08:28 52 69 6E 67 20 4D 61 \$552F  
 2C10:69 6C 00 1E 37 29 42 61 \$6132  
 2C18:72 20 4D 61 69 6C 00 1E \$4BA7  
 2C20:37 2A 43 68 61 69 6E 20 \$FD8D  
 2C28:4D 61 69 6C 00 1E 37 2B \$26BC  
 2C30:50 6C 61 74 65 20 4D 61 \$C715  
 2C38:69 6C 00 1E 37 2C 52 69 \$6634  
 2C40:62 62 65 64 20 4D 61 69 \$0D97  
 2C48:6C 00 1E 37 2D 51 75 69 \$6DE8  
 2C50:74 00 00 00 1E 37 25 4D \$ECE6  
 2C58:61 67 69 63 20 4D 69 73 \$BCCA  
 2C60:73 69 6C 65 73 00 1E 37 \$E2DB  
 2C68:26 46 69 72 65 62 61 6C \$617F  
 2C70:6C 73 00 1E 37 27 53 6F \$FA23  
 2C78:6E 69 63 20 57 68 69 6E \$78E5

2D58:6F 6E 20 49 49 20 43 44 \$A79C  
 2D60:41 20 43 68 65 61 74 20 \$A94D  
 2D68:76 32 2E 30 20 0F 20 20 \$B88B  
 2D70:20 20 20 20 20 20 5F 1E \$C71B  
 2D78:34 22 5A 4C 4C 4C 4C 4C \$606A  
 2D80:4C 4C 4C 4C 4C 4C 4C 4C \$F03A  
 2D88:4C 4C 4C 4C 4C 4C 4C 4C \$606A  
 2D90:4C 4C 4C 4C 4C 4C 4C 4C \$F03A  
 2D98:4C 4C 4C 4C 4C 4C 4C 4C \$606A  
 2DA0:4C 5F 1E 34 23 5A 1E 5B \$717C  
 2DA8:23 5F 1E 34 24 5A 1E 5B \$00C6  
 2DB0:24 5F 1E 34 25 5A 1E 5B \$5A08  
 2DB8:25 5F 1E 34 26 5A 1E 5B \$BDFA  
 2DC0:26 5F 1E 34 27 5A 1E 5B \$33FC  
 2DC8:27 5F 1E 34 28 5A 1E 5B \$3E76  
 2DD0:28 5F 1E 34 29 5A 1E 5B \$9C88  
 2DD8:29 5F 1E 34 2A 5A 1E 5B \$034A  
 2DE0:2A 5F 1E 34 2B 5A 1E 5B \$757C  
 2DE8:2B 5F 1E 34 2C 5A 1E 5B \$54E6  
 2DF0:2C 5F 1E 34 2D 5A 1E 5B \$5E08  
 2DF8:2D 5F 1E 34 2E 5A 1E 5B \$E9DA  
 2E00:2E 5F 1E 34 2F 5A 1E 5B \$37FC  
 2E08:2F 5F 1E 34 30 5A 1E 5B \$C216  
 2E10:30 5F 1E 34 31 5A 1E 5B \$9088  
 2E18:31 5F 1E 34 32 5A 1E 5B \$FF2A  
 2E20:32 5F 1E 34 33 5A 1E 5B \$797C  
 2E28:33 5F 1E 34 34 5A 1E 5B \$A886  
 2E30:34 5F 1E 34 35 5A 1E 5B \$5208  
 2E38:35 5F 1E 34 36 5A 0E 20 \$4697  
 2E40:53 65 6C 65 63 74 3A 20 \$9D4D  
 2E48:0F 4A 0E 20 0F 4B 0E 1E \$E688  
 2E50:52 36 4F 70 65 6E 3A 20 \$9954  
 2E58:0F 4D 0E 20 20 0F 5F 1E \$5AD6  
 2E60:35 37 4C 4C 4C 4C 4C 4C \$4798  
 2E68:4C 4C 4C 4C 4C 4C 4C 4C \$6768  
 2E70:4C 4C 4C 4C 4C 4C 4C 4C \$4718  
 2E78:4C 4C 4C 4C 4C 4C 4C 4C \$6768  
 2E80:4C 4C 4C 4C 4C 4C 4C 4C \$4718  
 2E88:0E 1E 3C 24 57 72 69 74 \$BFF2  
 2E90:74 65 6E 20 62 79 20 56 \$65F1  
 2E98:69 6E 63 65 20 41 6E 64 \$290C  
 2EA0:72 65 77 73 1E 3A 26 4D \$D62B  
 2EA8:65 6D 62 65 72 20 6F 66 \$E7AA  
 2EB0:20 43 6F 6D 70 75 74 69 \$BC14  
 2EB8:73 74 20 4D 61 67 61 7A \$714D  
 2EC0:69 6E 65 1E 37 27 50 4F \$0E8E  
 2EC8:20 42 6F 78 20 32 34 32 \$7BD8  
 2ED0:2C 20 4B 61 70 6F 77 73 \$DCA8  
 2ED8:69 6E 20 57 41 20 39 38 \$2591  
 2EE0:33 34 34 2D 30 32 34 32 \$4F68  
 2EE8:1E 36 2A 31 32 2F 31 36 \$FC52  
 2EF0:2F 39 31 20 76 31 2E 30 \$093C  
 2EF8:20 2D 20 53 74 61 72 74 \$B1F7  
 2F00:65 64 20 74 68 69 73 20 \$A973  
 2F08:70 72 6F 67 72 61 6D 1E \$4DEE  
 2F10:36 2C 31 32 2F 32 31 2F \$B056  
 2F18:39 31 20 76 31 2E 31 20 \$A9B5  
 2F20:2D 20 46 69 78 65 64 20 \$E282  
 2F28:74 68 65 20 73 65 61 72 \$985A  
 2F30:63 68 20 62 75 67 1E 36 \$4937  
 2F38:2E 31 32 2F 32 33 2F 39 \$AD98  
 2F40:31 20 76 31 2E 32 20 2D \$C5CC  
 2F48:20 46 69 78 65 64 20 74 \$8DD8  
 2F50:68 65 20 57 65 61 70 6F \$6BA4  
 2F58:6E 73 20 26 1E 46 2F 41 \$C8B8  
 2F60:72 6D 6F 72 20 72 6F 75 \$0392  
 2F68:74 69 6E 65 73 1E 36 31 \$7568  
 2F70:31 32 2F 33 30 2F 39 31 \$6566  
 2F78:20 76 32 2E 30 20 2D 20 \$F6C9  
 2F80:52 65 77 72 6F 74 65 20 \$9EFD  
 2F88:74 68 65 20 73 65 61 72 \$A415  
 2F90:63 68 1E 46 32 72 6F 75 \$2581  
 2F98:74 69 6E 65 73 20 20 28 \$F5F2  
 2FA0:66 61 73 74 65 72 21 29 \$E908  
 2FA8:0E 18 68 FA 85 F4 86 F6 \$AB8F  
 2FB0:00 F7 09 01 00 00 00 11 \$E88B  
 2FB8:4C 4F 52 55 5A 60 94 9E \$3255  
 2FC0:A1 A4 A9 AE B6 BF C8 EC \$D8E7  
 2FC8:F4 F9 1C 09 10 18 22 25 \$9805  
 2FD0:2B 33 36 39 3C 3F 42 45 \$912C  
 2FD8:48 4B 4E 53 60 65 68 86 \$8393  
 2FE0:99 AC C7 CD D2 E0 ED FC \$DA4A  
 2FE8:0F 04 1E 31 47 4C 51 72 \$88DC  
 2FF0:7A 98 A0 B8 BB D3 DC E7 \$40D1  
 2FF8:EA 28 07 0C 0F 14 17 1C \$B25E  
 3000:21 26 29 2E 33 36 3B 40 \$ACAB



```

3008:45 48 4D 52 57 5A 5F 64 $DAC3
3010:6A 6D 70 74 7E 84 94 AD $6E35
3018:BD D3 D9 DC E1 E7 EA EF $1831
3020:F5 F8 FD 19 03 06 0B 10 $7550
3028:17 2C 36 49 63 89 8F 9F $2646
3030:A5 AB B0 B5 B7 B9 BB BD $A0AC
3038:BF C1 C7 D7 EF FF 11 17 $1F15
3040:27 3F 4F 67 77 8C 91 96 $187C
3048:9B A0 A3 BB C3 CF DD F0 $0DOA
3050:F5 17 05 0A 0F 14 17 2F $85AD
3058:37 43 51 64 69 79 7E 80 $8BF5
3060:82 84 86 88 8E 9E B8 C8 $6662
3068:DF EF 17 06 16 30 3C 3F $B9A9
3070:42 45 5E 67 70 91 96 9D $22D2
3078:A3 A6 A9 B3 C2 C5 C8 DF $F9DF
3080:EB F0 F5 18 17 1C 23 29 $3D47
3088:2C 2F 39 52 55 58 6F 7B $5ED9
3090:80 85 A7 AA B7 BE C4 C7 $7E92
3098:D6 DD E5 E8 F5 20 02 05 $764E
30A0:14 2B 2E 37 3C 43 48 4B $FD32
30A8:5A 5D 62 67 7F 84 90 99 $2910
30B0:A3 B4 BD D8 DA DC DE E0 $765E
30B8:E2 E4 E6 E8 EA EC EE F5 $439C
30C0:03 00 04 0A 55 0C F5 03 $E3B7
30C8:00 00 0A 0F 0C F5 03 00 $51ED
30D0:FC 09 9B 0B F5 03 00 F8 $4153
30D8:09 E4 0A F5 03 00 F4 09 $67CD
30E0:9C 0A F5 03 00 F0 09 08 $68E1
30E8:0A F5 01 F0 96 09 5F 0C $9153
30F0:F5 01 F0 6C 08 97 08 F5 $7619
30F8:01 F0 36 08 89 08 F5 01 $D8A6
3100:F0 DC 07 07 08 F5 01 F0 $71C5
3108:B0 07 F9 07 F5 01 F0 2D $CFEF
3110:07 49 07 F5 01 F0 03 07 $E51C
3118:1A 07 F5 01 F0 DC 06 F3 $40CD
3120:06 F5 01 F0 B5 06 CC 06 $CABD
3128:F5 01 F0 8B 06 A2 06 F5 $6BD6
3130:01 F0 34 06 6D 06 F5 01 $C81B
3138:F0 C0 05 F9 05 F5 01 F0 $D8E2
3140:64 05 7B 05 F5 01 F0 3C $DE16
3148:05 53 05 F5 01 F0 14 05 $839C
3150:2B 05 F5 01 F0 EC 04 03 $6B4F
3158:05 F5 01 F0 C4 04 DB 04 $3F18
3160:F5 01 F0 8C 04 44 0E F5 $E46B
3168:01 F0 60 04 6E 04 F5 01 $CABF
3170:F0 46 04 76 04 F5 01 F0 $AB86
3178:29 04 7E 04 F5 01 F0 AA $49D3
3180:03 C1 03 F5 01 F0 81 03 $C220
3188:98 03 F5 01 F0 9D 02 AA $9019
3190:02 F5 01 F0 77 02 AA 02 $A64F
3198:F5 01 F0 A9 01 F5 0D F5 $BC52
31A0:01 F0 96 01 5D 0D F5 01 $0839
31A8:F0 83 01 D9 0C F5 01 F0 $1782
31B0:15 01 5F 0C F5 01 F0 91 $9B6B
31B8:00 B2 0C F5 01 F0 5D 00 $F024
31C0:8B 0C F5 03 00 10 00 77 $5DFC
31C8:01 F5 03 00 0C 00 14 00 $3386
31D0:00 $EOB3

```

### QUESTRON.CHT

```

2000:37 02 00 00 00 00 00 00 $6BA0
2008:C7 01 00 00 00 0A 04 02 $79C7
2010:00 00 01 00 00 10 00 00 $4BCA
2018:00 00 00 00 00 00 00 00 $3B6A
2020:00 00 01 00 00 00 00 00 $69DF
2028:2C 00 40 00 51 55 45 53 $4186
2030:54 52 4F 4E 2E 43 51 55 $E90B
2038:45 53 54 52 4F 4E 2E 43 $E1F1
2040:F2 C7 01 00 00 82 17 00 $4657
2048:56 69 6E 63 65 20 41 6E $62E1
2050:64 72 65 77 73 20 31 35 $69B0
2058:20 4A 61 6E 20 39 32 4B $3612
2060:AB C2 30 F4 01 00 F4 00 $25A1
2068:00 F4 03 00 A2 0C 10 22 $F98B
2070:00 00 E1 F4 01 00 A2 0C $5E1D
2078:15 22 00 00 E1 F4 06 00 $F38C
2080:A2 0C 18 22 00 00 E1 F4 $8AEC
2088:00 00 F4 00 00 F4 00 00 $200A
2090:F4 00 00 F4 00 00 48 A9 $D304
2098:00 50 48 A2 03 20 22 00 $0DDE
20A0:00 E1 F4 00 00 F4 B8 01 $6BF2
20A8:F4 00 00 A2 11 09 22 00 $13A6
20B0:00 E1 90 03 82 DB 00 68 $1DFE
20B8:8D B6 01 68 8D B2 01 68 $75E8
20C0:8D B4 01 68 68 F4 00 00 $FD40
20C8:F4 00 00 AD B4 01 48 AD $C06A

```

```

20D0:B2 01 48 A2 02 1A 22 00 $D636
20D8:00 E1 68 8D AE 01 68 8D $5DCE
20E0:B0 01 AD B0 01 0D B2 01 $6166
20E8:D0 2A 20 72 01 C9 CC CC $0437
20F0:C5 C7 C1 CC A0 C8 C1 CE $05FE
20F8:C4 CC C5 A1 A0 C4 C1 A0 $C0EA
2100:EE EF F4 A0 E9 EE F3 F4 $138C
2108:E1 EC EC E5 E4 AE 8A 8D $CAE2
2110:00 82 39 00 AD B0 01 48 $7856
2118:AD AE 01 48 A2 05 0F 22 $8F7F
2120:00 00 E1 20 72 01 8A 8D $D96C
2128:D9 EF F5 A0 ED E1 F9 A0 $4187
2130:EE EF F7 A0 EC E1 F5 EE $3BC1
2138:E3 E8 A0 F4 E8 E5 A0 D1 $3EB6
2140:B2 AE D3 D9 D3 B1 B6 A0 $8653
2148:E6 E9 EC E5 00 20 72 01 $4FA9
2150:8A 8D 8A 8D D0 F2 E5 F3 $3CA0
2158:F3 A0 D2 C5 D4 D5 D2 CE $C934
2160:AE 00 E2 30 AF 00 C0 E0 $9C43
2168:10 FA EB AF 10 C0 E0 EB $3B47
2170:C9 8D D0 F0 20 72 01 8D $3170
2178:8D 00 C2 30 A9 00 00 22 $288F
2180:A8 00 E1 29 00 46 01 00 $C9E8
2188:00 B0 F4 00 00 00 00 00 $4384
2190:00 6B 48 20 72 01 8A 8D $B2DD
2198:D1 F5 E5 F3 F4 F2 EF EE $2105
21A0:AE C3 C4 C1 A0 F7 E1 F3 $5CE7
21A8:A0 EE EF F4 A0 E6 EF F5 $4913
21B0:EE E4 00 68 82 96 FF C2 $7864
21B8:21 A3 01 1A 83 01 E2 30 $FC5F
21C0:A0 00 B3 01 F0 05 20 89 $864C
21C8:01 80 EC C2 30 60 48 DA $8451
21D0:5A 08 C2 30 48 A2 0C 18 $82EA
21D8:22 00 00 E1 28 7A FA 68 $9B55
21E0:60 08 E2 30 A0 03 B9 B8 $C1D9
21E8:01 F0 06 20 89 01 C8 80 $2FB9
21F0:F5 28 60 00 00 00 00 00 $758C
21F8:00 00 00 00 00 0E 31 2F $08F1
2200:51 55 45 53 54 52 4F 4E $2B69
2208:2E 43 44 41 F7 17 00 00 $08BE
2210:00 00 0D 61 74 78 7C 87 $EADC
2218:8B 97 9B 9E A1 A6 D0 D4 $D358
2220:DF 05 09 30 4F 82 A2 A7 $82E5
2228:F5 03 00 40 01 46 01 F5 $0EA6
2230:01 F0 5E 00 B8 01 00 $337E

```

Krakowicz NY

## The Basics of Kracking Part #15

### Kracking SSI's RDOS

It's really not fair when one publisher has a system that keeps their software from being conveniently backed up, especially when so many of the other "protection" schemes have fallen to the growing corps of talented Krackists. With that in mind, and because we all love a challenge, we will take a long look at the approach used by Strategic Simulations, Inc. (SSI) in providing copy protection for their series of war simulations and "rapid-fire" series, as well as some recent games which have a little redeeming social merit: Epidemic, Ring-side Seat, and Galactic Adventures.

As we've discussed in the basics of Kracking series, you can either protect a program by various means, or you can protect a disk full of programs with some sort of DOS modification. DOS modifications are usually not too successful, since some enterprising person out in pirate land will sooner or later figure a way to copy all the files onto a normal DOS disk, making all the disk protection worthless. SSI's answer to this problem was not only to write an extensively revised DOS, but to couple it with "enhancements" to Applesoft using the ampersand vector (more on this later). This way, even if you could

strip the files off the disk, you would need to write a DOS which was:

1. DOS 3.3 compatible
2. As short as RDOS (\$B100-\$BFFF), since the programs frequently use all of the free space
3. Capable of correctly interpreting the ampersand commands which are liberally sprinkled through all the Applesoft programs.

The amount of effort required to do this has kept Krackists at bay, at least until now.

First, how to approach this type of Kracking job? The seventh law of Krack-

not straightforward, "linear" code. You may recall that we described how to list an RDOS Applesoft file in Basics 103 (#3): reset, D6:00, C081, Ctrl-C, "LIST". You will be immediately struck by a whole new list of commands that mother Apple never told you about. These are ampersand (&) commands which have been added to implement the RDOS commands, and they work as follows: Whenever the "&" is encountered, Applesoft jumps to location \$3F5. Looking at that location will tell you where the ampersand evaluation routine is located; in this case, it contains "4C03B3" or JMP \$B303. Examination of the code

# "Krackowicz's seventh law of Kracking says: When you're totally lost, bootcode trace"

ing says: "when you're totally lost, bootcode trace" (I know, I promise that I'll write a complete column on boot-tracing soon. If you just can't wait, try to get hold of the Hardcore magazine update 3.1, pages 6-15. It has a lucid, well-exemplified discussion of the boot-tracing process). When you load T0, S0 into \$800, you will immediately see the familiar "Brody Loady" (named after that fun-loving bunch of Scandahoovians at Broderbund) which moves the entire page down to page 2 and jumps to \$20F to complete the boot. This is a fairly tricky boot which has been used for all types of protection schemes, but if you puzzle over it long enough, you'll see that the JMP (\$003E) at location \$343 doubles as a jump to the sector read routine, then as a jump to the program start when all the sectors are read in. The program start in this case is \$B300, which is a JMP \$B974 that drops you into a discouragingly complex series of JSR's and JMP's.

At this point, discretion is the better part of valor (remember the second law: There's always another way). What the boot-trace has told us is that the DOS code lives from \$B300 to \$BFFF, and is

there reveals that the accumulator is compared to a table of numbers in \$B320-\$B330, and the address of the routine to be executed is picked up from a table in \$B331-\$B352. You can easily see all this code by resetting any of the RDOS SSI games, and if you're really interested, you can contact your local pirate for a copy of the source code listings, in Big Mac format, for both the original and DOS 3.3 compatible versions of RDOS. The disk also contains object code for RDOS 3.3 and listings of the other programs used for secondary protection and initializing. If there's enough interest, the sysop might be persuaded to include them on his Apple Trek Kracking disk #2.

Now we're starting to make progress. Each time the "&" command is encountered, Applesoft obligingly jumps up to \$B303 with the hex value of the next basic token in the accumulator, and then decides what to do next. The tokens, with their value, subroutine address start, and function are shown in figure 1.

As you can see, not all have been chased down. Interested parties are invited to investigate and share the results with us all.

Figure 1

& command	hex	dec	Addr	Function in RDOS
C	43	67	B353	Catalog (&CAT)
LOAD	B6	182	B371	Load Applesoft file
RUN	AC	172	B446	Run Applesoft file
GOTO	AB	171	B44C	Exec (?)
SAVE	B7	183	B48D	Save Applesoft file
STORE	A8	168	B511	Bsave (with A,L)
RECALL	A7	167	B52B	Bload, A optional
DEF	B8	184	B544	?
PRINT	BA	186	B56C	Write to textfile
READ	87	135	B582	Read text file
END	80	128	B5A9	Close a file
DEL	85	133	B5AE	Delete a file
LEN	E3	227	B5CD	?
D	44	68	B620	Drive?
S	53	83	B62E	Slot?
NEW	BF	191	B64F	?
USR	D5	213	B670	Change return from &

But this, too, has only a limited value, since we run out of informative tokens at about \$B679 in a listing that continues up to \$BFFF. The only way to get there is to call up the infantry and slog our way through the code, brute force. Since it's a DOS, there must be read and write code of some sort, so looking around for disk accesses (\$C08C,X) is a good place to start. The first encounter is at \$BB6B, which is clearly a "write" section—\$C08F,X = output; \$C08E,X = sense write protect. It's followed by a read sector routine at \$BBFD-BC64, and read address routine at \$BC65-\$BCC0. On close examination, it can be seen that the

RWTS" (it goes into \$8000 as the default location). Next, BLOAD "RDOS WRITE",A\$7000, then move it to the normal RWTS locations with B700<7000.78FFM (this is necessary because you're using the RWTS routines to read in the files; writing on top of operating code can lead to very unpleasant results). Return to BASIC, delete line five, and type "RUN". answer the question "13 sector", enter the appropriate slots and drives, and you're off and running to create an RDOS 3.3 copy (when you're finished, you'll have a 16-sector diskette with only 13 sectors occupied per track, but you won't notice it in use).

0, 7, E, 6, D, 5, C, 4, B, 3, A, 2, 9, 1, 8, F.

The second catalog sector, then, appears to be sector 7. If you intend to do any amount of playing around with one of these disks, use the "RDOS WRITE" RWTS from the COPYB disk, or change bytes \$BE2A-\$BE2D to \$EA's with the Inspector. This omits the table lookup and makes the sector numbers follow the sequence as used by RDOS.

Next, copy the file called RDOS 3.3 from the COPYB disk (or track 0 of any of the recently unprotected SSI series) onto track zero, sectors 0-D. You would

entire track and obliterate data on any adjacent half-track. After reading in the data, the memory values are exclusive-or'ed (XOR) with the address (\$1000 contains \$00, \$1001 contains \$01, etc.), and if an error is found, it reboots the disk. Placing an RTS (\$60) at the entry point of \$A0F0 will avoid the entire issue and make the COPYA version run.

The final (I hope) hurdle to using RDOS 3.3 is the program which initializes a save game diskette in an RDOS compatible format. It is called SSI.INIT and loads into \$800-\$AFF (it is usually accessed via a "call 2800" from a basic program). Since it only writes address fields, and not data sectors (with no verify), it is a very fast init. All that's necessary to generate a disk compatible with RDOS 3.3 is to replace the "D4 AA B7" (or "D5 AA B5") address marker byte with D5 AA 96: change bytes \$8F5 to \$D5 and \$8FF to \$96.

Now for the bad news: while RDOS is fast, primarily because all files are stored in sequential blocks, RDOS 3.3 is slow because of the sector interleave used by SSI (the disk must make an almost an entire revolution for each sector that is read in). It is fairly easy to add an interleave lookup table to RDOS 3.3 (it's called RDOS 3.3a on the disk), but bad things happened during my attempts to incorporate it into the SSI.INIT program. The code from \$851 to \$86B needs more alteration than I had patience for (there's room for a patch in \$9D7-9FF), and would be worth the effort if some ambitious Krackist out there could find the time...

Finally, as a typical example of Murphy's Law ("if anything can go wrong, it will, and at the worst possible moment), that after doing all this and converting some 20 games, the very last one I tried was Germany 1985. This is a fairly recent publication of SSI which is completely written in machine language, does not use RDOS at all, and will require a totally different approach to unprotection.

Brian Coombs Australia

I need help with the following softkeys:

Risk v1.4 (issue #85, pg 19) by Alan Chaney. The softkey works fine but I am unable to transfer it to a 3.5" disk even though it is a ProDOS program.

Alternate Reality-the Dungeon (issue #52). I get an "Out of Data Error at 310" when I try to deprotect my backup copy of the game. When I softkey the original disk, I get a grating sound in the disk drive then the message "Disk Drive Error".

In addition, when I play this game, I find that when my character gets to a certain level, he keeps meeting this creature called the devourer—there are no encounters with other monsters. Even if I use the APT to create a super character he will be eventually worn down. Is there another way to deal with this creature or is it a bug in the program?

Champions of Krynn (issue #77) by Dr. Crack. In his softkey, Dr. Crack gives instructions to edit sectors 18 and 19 of a particular track. My sector editor only goes up to 16. Is this a misprint or does this disk have 19 sectors per track? If it does, what procedure do I use to make the edits?

My setup is: Apple IIGS, ROM version 1, 1.25 Mbytes RAM.

## "Krawowicz's second law of Kracking says: There is always another way."

routines have been lifted almost verbatim from DOS 3.2, with the address marker changed to "D4 AA B7" (in most cases). Aha! Maybe we can sneak in the appropriate routines from DOS 3.3 and make it "DO D5 AA 96's?" To make a lengthy story short, the answer is yes, but. DOS 3.2 uses, as you probably know, "6+2" nibblizing in storing data on the disk, while 3.3 uses "5+3". The end result is that the pre- and post-nibblizing routines must be transplanted from DOS 3.3, as well as both read and write byte translate tables. The address markers and the size of the nibble buffers must also be adjusted. When this is done (with much wailing and gnashing of teeth), the end result is a functional, DOS 3.3 compatible RDOS: RDOS 3.3.

(As a brief aside, the essential tools in this task are (of course), Beneath Apple DOS, and the DOS SOURCE commented listing of all the DOS code).

Now, we know from previous games like Crisis Mountain and Ming's Challenge that we can read the sectors into memory from a disk with modified RWTS routines by using its own RWTS and the Inspector, then swapping RWTS routines to standard DOS 3.3 and writing them out again on a formatted disk. The prospect of doing all the SSI games by hand boggles the mind, however, and requires an automated approach (they were supposed to work for ->us<-, remember?). The answer to this problem was the program now known as COPYB - a highly modified version of COPYA which does the RWTS swap for you, and even initializes disks as a bonus. The version of COPYB in general circulation includes RWTS routines which have been modified for reading and writing RDOS. Reasonable directions are included on the disk, so it should be possible to back up your own SSI disks, using the additional information provided below.

Armed with RDOS 3.3 and COPYB, it is now possible to begin attacking one of the SSI protected disks. Since RDOS is based on DOS 3.2, the disks are all 13-sector format, and since the DOS is all on track zero, you want to begin the track copying process with track one. To reiterate the COPYB instructions, run COPYB, then type Ctrl-C or reset when the prompt for source disk comes up. Get into the monitor and type 22E:1 to set the starting track to 1, then, if the address marker bytes were D4 AA B7, BLOAD the file called "RDOS READ

Some of the SSI games use the normal DOS 3.2 address marker bytes of "D5 AA B5." These should be read in using the "DOS 3.2 RWTS" file, but you still need to use the "RDOS WRITE" RWTS for the writing routine.

RDOS uses track 1 for the catalog, and identifies files via a 24-character alphanumeric name, a length in "blocks" as in Pascal, and the location of the starting block on the disk:

expect to have a working copy of the game at this point, but there are still a couple of surprises in store for you (I said it was a challenge!). There are several different secondary protection schemes used to defeat various copiers, usually going under the innocuous name of "QWERTY". The most common of these reads in an address field from track 0, delays a bit, and looks for an \$EE as the next byte on the track. If it finds it, a

TRACK	D1	SECTOR		0	SLOT		6	DRIVE		1
BUFFER 0800 DOS 16 2BCC										
=====										
	0	1	2	3	4	5	6	7	8	9 A B C D E F
00-	R	D	O	S	2	.	1	C	O	P Y R I G
10-	H	T	1	9	8	1	B	1A0010	001A0000	
20-	S	Y	S	T	E	M	B	O	O	T
30-								T	0100B1	00011A00
40-	R	E	G	1						
									B	0A6009 DC091B00
PROGRAM										FILE TYPE
										A, T, B
NAME										
										FIRST
										BLOCK
										NUMBER
										OF
										STARTING
										LOCATION
										BLOCKS

The starting block is equal to the track number multiplied by 13 plus the first block (\$1A00 is really \$001A, which is decimal 26, or track 2, sector 0). If you look through the catalog track with the Inspector, you find the beginning of the catalog as expected in T1,S0. Looking for the continuation in T1,S1, however, brings you to the next surprise held by RDOS: There is no sector interleaving in software; it is all done by the sector number sequencing during SSI's initialize routine. The importance of sector interleaving is discussed in "Bag of Tricks", and in a Softalk article about a year ago by Worth and Lechner. (DOS uses a lookup table at \$BFA8 to change the sector number read from the value read off the disk ("physical sector") to the number it thinks it should be ("logical sector"). SSI uses an "ascending 7" interleave scheme, which means that the sequence of sectors on the disk, as read by DOS 3.3 with its hex interleave table, is:

0 is stored in location 0, otherwise the disk spins forever. By changing bytes \$28-\$29 to "A9 00", this annoyance is removed. A similar routine, seen only once or twice, is called "QWERTY", looks for an \$AA following the address field on any track, and reboots if it's not found. The remedy here is to put "A9 00" in bytes \$20-\$21.

Recently, a much more sophisticated technique has been used (Galactic Gladiators, Road to Gettysburg), which does the SSI equivalent of "quarter-tracking" or "spiralling." This version of QWERTY reads in four pages of sequential bytes from each of the four adjacent half-tracks from \$20.5 to \$22.0, storing them at \$1000-\$1FFF. The three bytes following the four pages worth are used as the address marker for the data on the next half-track (as with all these protection techniques, the "sectors" are skewed so that there is never valid data overlapping on adjacent half-tracks). This approach effectively defeats copiers like NA II and Locksmith, which write an



Softkey for...

**Outpost****Sirius**

Just like almost every other Sirius disk, boot code tracing is the key to getting Outpost in memory. Once again, it is the standard Sirius boot program so often seen when breaking into Sirius disks. Seeing that I've documented this procedure in other softkeys I'll skip it this time.

Power up your Apple and let's begin!

CALL-151  
9600<C600.C6FFM  
96FA:98 N 9801:4C 59 FF  
9600G  
C0E8  
9800<800.8FFM  
9810:64 N 984C:68  
985C:01 N 9870:98  
987E:59 FF  
9600G  
C0E8  
8400<6400.67FFM  
6463:84  
6466:85  
6469:86  
646C:87  
9810:04 N 984C:08  
987D:A0 00 B9 00 64 99 00 04  
:B9 00 65 99 00 05 B9 00  
:67 99 00 07 C8 D0 EB 4C  
:46 04

The above procedure is standard for the other Sirius disks containing this same boot code. The only thing that usually varies in the boot code is what it does after the program is in memory. In this case it ends up at \$520 (which is currently at \$6520). There, it sets the stack values (which tell the RTS at \$558 where to go) and also plays around with the zero page.

At \$6523 is a JSR \$7D0 which should be changed to BIT \$7D0 because \$7D0 is a memory check routine that will cause a reboot.  
6523:2C

It normally exits at \$6558 so change the RTS to a JMP \$7D0 and we'll write a routine at \$7D0 to move the zero page to \$2700 and save the stack addresses.  
6558:4C D0 07  
67D0:A2 00 BD 00 00 9D 00 27  
:E8 D0 F7 68 8D 00 20 68  
:8D 01 20 4C 59 FF

Call up \$9600 to boot the disk and load the program.  
9600G

The program is in memory from \$800 to \$8FFF and hi-res page 1 is blank. Check locations \$2000 and \$2001 where we stored the entry point of the game program.  
2000.2001

The values should be \$FF and \$7F. The smallest typing error along the way could cause these values to be incorrect and if so, the program will probably not work correctly.

If everything is correct, move \$800-\$1FFF to \$2800.  
2800<800.1FFFF

At \$813D is a JMP \$8EC0 which is a disk check routine. Upon a successful check, it returns by RTS with no checksum. It can be easily disabled by putting an RTS at \$813D.  
813D:60

Insert a slave disk with no HELLO program and at least 105 available sec-

tors, then reboot DOS and enter the monitor.

C600G  
CALL-151

A memory move routine must be written to move \$2800 back to \$800 and \$2700 back to the zero page. Because the code begins at \$2700 this can be written right before it at \$26C0.

26C0:A2 00 BD 00 28 9D 00 08  
:E8 D0 F7 EE C4 26 EE C7  
:26 AD C4 26 C9 40 D0 EA  
:BD 00 27 9D 00 00 E8 D0  
:F7 4C 00 80

Finally, save the program to disk. We'll save \$26C0 through \$8DFF. Pages \$8E and \$8F are only for the disk check and do not need to be saved.  
BSAVE OUTPOST,A\$26C0,L\$6730

Softkey for...

**Twerps****Sirius**

If Sirius software were still in business, I'm sure they would have begun to despise me by now. Twerps is about the 15th Sirius game I have cracked and many of these cracking procedures I have submitted (or will be submitting) for publication in Computist. After a while, cracking Sirius disks became quite routine because most of them are protected the same way. However, even though many of them are alike, there is just enough variance from disk to disk to make one Softkey procedure obsolete on the next disk. Twerps is no exception.

Much like Epoch, Twerps occupies a great amount of memory and would take a lot of extra work to try to compact it into a standard DOS file. The program code runs from \$400 to \$BFFF and Hi-res page 2 is blank. The title picture loads from the disk separately and uses \$2000-\$3FFF. You could easily save all of the working game code, sacrificing the picture, and have a BRUNable file of Twerps or if you prefer to salvage the picture, you could scrunch it down to size with a picture packer and include it in the file. This is the way I originally did it, but because not all of us have the same picture packer, it would be a bit more difficult to document.

This game will be deprotected and saved much in the same manner as Epoch. It will involve boot code tracing the disk to get the program and writing the code directly to the copy disk with the RWTS. Also, because the title picture normally loads in from the disk between games and during the demo mode, we will use the RAM card to store the picture somewhat in the same way Gorgon was done.

We will begin by creating a blank copy disk with a modified boot 1. This is so we can boot this disk without DOS loading over part of the game program.  
FP

CALL-151  
B700:4C 59 FF  
INIT HELLO  
DELETE HELLO

Set this disk aside and insert the Twerps disk. It's time to boot code trace another typical Sirius disk. Although the below procedure is not documented step by step, try to use your own knowledge and what you've learned (if you understand machine language) to figure out what each step is doing. Believe me, it's by far the best way to learn and understand how protection works.  
9600<C600.C6FFM

96FA:98 N 9801:4C 59 FF  
9600G  
9800<800.8FFM  
985C:01 N 9870:98  
987D:A9 4C 8D 00 05 A9 59 8D  
:01 05 A9 FF 8D 02 05 4C  
:46 04  
9600G  
C0E8

All of the program code is in memory at this point but the title picture still hasn't been loaded in. This is done at \$BC03. A few modifications must be made to bypass a check to track \$17. Also, we want to load the picture into \$4000 instead of \$2000 because the code that runs at \$400-\$1FFF is at \$2000 right now.

BC03:20 90 BE  
BC22:4C 59 FF  
BC10:40  
BC17:4C  
BC1E:54  
300:A2 60 86 2B 4C 03 BC

The brief routine at \$300 restores the slot number in location \$2B before calling \$BC03. Turn on the drive, execute \$300, and turn off the drive.

C0E9  
300G  
C0E8

Everything we need to save is now in memory from \$2000-\$BFFF. We'll be using the RWTS (which occupies \$B800-\$BFFF) to save this game to disk. First, we have to move the \$B000 area memory elsewhere to make room for the RWTS.

1000<B000.BFFFFM

At \$BC00 (now \$1C00) is the routine that loads the title picture in from the disk. Since we'll be storing the picture in the RAM card, we'll write the routine at \$1C00 to move the picture from the RAM card to the Hi-res screen.

1C00:AD 83 C0 AD 83 C0 A9 00  
:85 D0 85 D2 A9 D0 85 D1  
:A2 20 86 D3 A0 00 B1 D0  
:91 D2 C8 D0 F9 E6 D1 E6  
:D3 CA D0 F2 AD 82 C0 60

At \$BCA6 is a routine to move pages \$20 to \$3B down to \$400. This is normally called right before jumping to \$400 to start the game but hasn't been called yet. To avoid writing an extra memory move routine, we will use this one before starting the game. Make one modification by patching an RTS at the end of the routine.  
1CC4:60

Insert the Twerps COPY disk, boot the disk, and turn off the disk drive.

C600G  
C0E8

Because we modified the boot program on the copy disk, only the RWTS is intact at this time. The rest of DOS did not load. The RWTS is all we will need to write the game to disk as well as read it in at boot. Begin by entering the necessary IOB information for writing the game to the disk.

B7EB:00 0C 0F  
B7F0:00 AF 00 00 02  
B7E1:A0  
B793G

Once the writing is finished, reboot DOS and run your sector editor. Then read track 00, sector 01. This is technically boot stage 1 (\$B700). You should see 4C 59 FF at the beginning of this sector. Beginning right at byte 00, enter the following program:

Trk	Sec	Byte	From	To
00	01	00	4C 59 FF	20 93 B7 A2 00

??	BD 00 B7 9D 00
	03 E8 D0 F7 4C
	11 03 BD 00 10
	9D 00 B0 E8 D0
	F7 EE 13 03 EE
	16 03 AD 16 03
	C9 C0 D0 EA 20
	A6 BC AD 81 C0
	AD 81 C0 AD 83
	C0 AD 83 C0 BD
	00 40 9D 00 D0
	E8 D0 F7 EE 38
	03 EE 3B 03 AD
	3B 03 C9 F0 D0
	EA AD 82 C0 A9
	4C 8D 00 03 A9
	00 8D 01 03 A9
	BC 8D 02 03 AD
	50 C0 4C 00 04
\$00 \$01 \$EB ??	00 0C 0F FB B7
	00 AF 00 00 01
	A0

That's the IOB data for the loading of the game at \$EB and the length (in pages) of the game program at \$E1.

Rewrite this sector and you're finished.

Softkey for...

**Snack Attack****Datamost**

Snack Attack is written to disk with a modified DOS 3.2. It can be deprotected one of two different ways. Either by copying the disk track by track to normal DOS 3.3, or boot code tracing the disk and saving the game as a BRUNable file. Due to the memory usage of this game, the hi-res picture would either have to be packed or sacrificed to save the game as a file. The easiest way to do it in this case is to copy the disk to a normal format. This can be done with a Super IOB swap controller. If you are interested in saving this game as a single file, you can use the boot code trace steps for County Fair in Issue #82. Take note that ONLY the boot code trace procedure will work for Snack Attack as the start address and memory use is not the same as County Fair. Once the game is in memory, the rest will be up to you if you wish to try it that way.

Otherwise, begin by formatting a disk for the copy.

INIT HELLO

Now we have to get the Snack Attack RWTS in memory. Boot the disk and after a few seconds, press RESET several times until the prompt appears. Then, enter the monitor and move the RWTS to \$1900 where Super IOB will use it.  
CALL-151

1900<B800.BFFFFM

Reboot DOS, load Super IOB and install the controller listed at the end of this article. It will copy tracks \$00 to \$11 from Snack Attack to the copy disk. Also, all tracks except for track 0 will be written with a reverse (Pascal) skew. This is to maximize loading speed. DOS 3.2 reads faster when the sectors are read in ascending order but DOS 3.3 is just the opposite. If we were to make the copy using the normal 3.3 skew, it would take 20-30 seconds just to load the game. It is possible to change the loader to read in reverse order but it is much easier to just write with a reverse skew.

When the Super IOB controller is installed, run the program and copy the disk. There will be a read error on track 0, and several on track \$11 but the program will ignore the errors and keep on copying. After the copy is finished, run

your sector editor. The original Snack Attack disk is no longer needed.

What we need to do now is copy a normal 3.3 RWTS to track \$00 but save sector \$0C which is the real boot 1 sector of Snack Attack. Boot 1 is normally found on track \$00, sector \$01 but Datamost used about 3 decoy boot 1 sectors to confuse boot code tracers.

This next step can be done one of two ways. You make the choice.

1. Read track \$00, sector \$0C and write it to track \$12, sector \$0C for safe keeping. Then take a copier that allows you to select tracks and copy track 0 from a normal (non-master) DOS disk. DON'T use the Apple master disk for this. Then, with your sector editor, read track \$12, sector \$0C and write it to track \$00, sector \$01.

2. Individually copy sectors \$00-\$09 (on track 0) one at a time from a normal disk to the copy disk. This is much more tedious than the first option. Then read sector \$0C from the Snack Attack copy and write it to sector \$01.

The final step of this Softkey is installing the sector skew into the RWTS on the disk. Remember, when you write with an abnormal skew, you must read with the same skew so the sectors are read in the correct order. The skew is found on track \$00, sector \$09 at byte \$B8.

Trk	Sct	Byte	From	To
00	09	B8	??	00 02 04 06 08 0A 0C 0E 01 03 05 07 09 0B 0D 0F

After the skew has been entered, re-write the sector and you're ready to play Snack Attack!

### CONTROLLER

```
1000 REM "SNACK ATTACK
CONTROLLER
1010 TK = 0:ST = 0:LT =
18:CD = WR
1020 T1 = TK: GOSUB 490:
GOSUB .360: ONERR GOTO 550
1030 GOSUB 430: GOSUB 100:ST
= ST + 1: IF ST < 13 THEN
1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1: IF
TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST =
0: GOSUB 360
1070 GOSUB 430: GOSUB 100:ST
= ST + 1: IF ST < 13 THEN
1070
1080 ST = 0:TK = TK + 1: IF
TK = 1 THEN GOSUB 2000
1085 IF BF = 0 AND TK < LT
THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE" :
END
2000 DATA 0,2,4,6,8,10,12,
14,1,3,5,7,9,11,13,15
2010 FOR I = 1 TO 16: READ
SS: POKE 49079 + I,SS:
NEXT : RETURN
```

### Checksums

1000-\$356B	1050-\$03B0	1090-\$2825
1010-\$3222	1060-\$4EFF	1100-\$F064
1020-\$16A9	1070-\$EBBA	2000-\$9E40
1030-\$E75E	1080-\$9128	2010-\$16F3
1040-\$0D7D	1085-\$971D	

Softkey for...

### Seven Cities of Gold

Electronic Arts

Here is another Most Wanted List disk sent to me by a Computist subscri-

er. Seven Cities of Gold is protected much like most of the EOA games I am familiar with. Most of them contain a normal or slightly altered DOS 3.3 format with the dreaded uncopyable track 6 which is used for a nibble count. Once finding the nibble count routine, I can be disabled. This disk appears to contain 3 nibble count routines in different places but I believe that only two of them are used. We'll begin by copying both sides of the disk.

You'll need a normal disk copier that bypasses read errors such as the Locksmith fast disk backup. The only read errors will be on Track 6 of side 1. After copying the disk, run your sector editor.

It didn't take long to find the first nibble count routine. It is on Track \$1, Sector \$F and resides in memory at \$A000. You would think that by just patching an RTS at the beginning of the routine, you could easily bypass it but EOA thought of that too. Somewhere, cleverly hidden in the program code is a memory check routine that will cause the program to crash if any changes are made. Through the process of trial and error, I concluded that it adds up all of the bytes in the routine and checks the result value. There is a way around this without having to find the memory check routine. Take the first three bytes of this routine which are \$4C \$69 \$A0. Add them up like this:

```
LDA #$4C
CLC
ADC #$69
ADC #$A0
```

The result is \$55. To bypass the nibble count, we want to put a CLC and an RTS (which would be an \$18 and a \$60) at the beginning of the routine. The trick is changing the third byte also where the sum of the three new bytes will also equal \$55. The third byte should be \$DD (\$18+\$60+\$DD=\$55).

Trk	Sct	Byte	From	To
\$01	\$0F	\$00	4C 69 A0	18 60 DD

If you were to boot the disk right now, you would make it just past the title page until it checks the disk again and crashes. This time, the nibble count routine is encoded on the disk. In a case like this, the best thing to do is halt the boot code after the program is loaded and search through memory to find the routine in its decoded form. I found the program start address at \$A946 in an encoded table of addresses. It uses this table for a series of indirect jumps obviously meant to confuse the cracker. By changing this address to \$FF59, I was able to enter the monitor and look at the program code once it was in memory and eventually, I found what I was looking for. The nibble count routine was at \$6100. Knowing this was encoded on the disk, I searched for a JSR \$6100 which I found at \$6875. Even though the actual nibble count routine was encoded, surprisingly the JMP was not. I found this on Track \$E, Sector \$7. Change the \$20 to a \$2C which will turn the JSR \$6100 to a BIT \$6100.

Trk	Sct	Byte	From	To
\$0E	\$07	\$75	20	2C

...and that takes care of the protection on Seven Cities of Gold!

Softkey for...

### Pensate

Penguin

This Most Wanted List member was sent to me by a subscriber who had read

my ad in the back of Computist. Pensate is on a standard type DOS 3.3 format but is a little trickier than some protection schemes. To begin, the address marks alternate between D5 AA 96 and D4 AA 96 on even and odd tracks respectively. Also, the DOS routines are encoded on the disk making them a bit more difficult to examine and modify. Let's begin and I'll try to explain as we go along.

Being that Pensate is on a mostly standard format, it can be copied track by track to a normal disk without too much of a problem. This can be done with a Super IOB controller. Begin by installing the controller into Super IOB and copying the disk. The only tracks containing data are \$00 to \$11. The controller pokes in the \$D5 address mark on even tracks and the \$D4 on odd tracks. This is done in line 1025 and line 1065 changes it to a normal \$D5 for writing. The rest of the controller is just standard procedure.

Now, run your sector editor. We need to change their DOS to read the normal format of the copy. Their DOS is capable of reading either a \$D4 or a \$D5 in the address field because of the way it was programmed. Here is an example:

```
1000- LDA $C08C,X
1003- BPL $1000
1005- LSR
1006- CMP #$6A
```

What this is doing is reading a byte from the disk, performing an LSR bit shift (which divides the byte by two leaving no decimal remainder) and comparing the byte to \$6A. Because the LSR

```
1025 DM = TK / 2 = INT (TK /
2): POKE 47445,212 + DM
1030 GOSUB 430: GOSUB 100:ST
= ST + 1: IF ST < DOS THEN
1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1: IF
TK < LT THEN 1025
1060 GOSUB 490:TK = T1:ST =
0
1065 POKE 47445,213
1070 GOSUB 430: GOSUB 100:ST
= ST + 1: IF ST < DOS THEN
1070
1080 ST = 0:TK = TK + 1: IF
BF = 0 AND TK < LT THEN
1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT
"DONE~WITH~COPY" : END
```

### Checksums

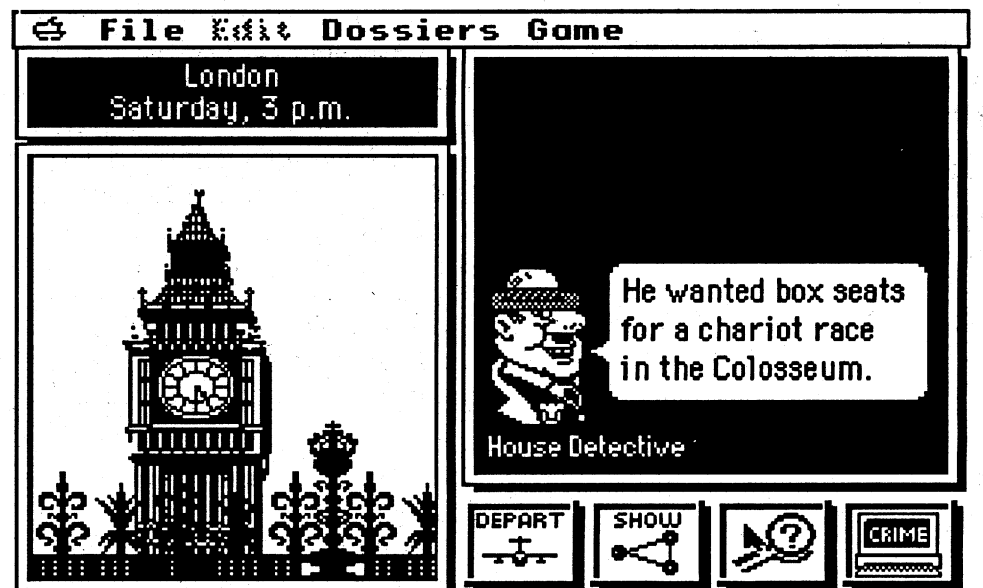
1000-\$356B	1040-\$57FB	1080-\$3D59
1010-\$3222	1050-\$5A21	1090-\$2807
1020-\$84E4	1060-\$E809	1100-\$D02C
1025-\$4755	1065-\$7991	
1030-\$5154	1070-\$7194	

Softkey for...

### Where in the World is Carmen Sandiego?

Broderbund

There have been two other Computist Softkeys for this game but this Softkey is for later releases of the game with a different copy protection. 'Where in the World...' appears to be the first Carmen Sandiego disk released by Broderbund and also may be the only disk of the series that was released prior to the birth



leaves no remainder, \$D4 and \$D5 will both result in a \$6A when bit-shifted right. Therefore, this part of the loader need not be modified.

The only other difference between Pensate's DOS and normal DOS is that the end marks have been changed from DE AA to DA AA. By changing their DOS to look for the DE AA, the copy will work. The problem is that the DOS is encoded on the disk. By making a quick modification to the boot code, I was able to look at the decoded DOS and also find out how it was encoded. With the needed information at my fingertips, I came up with the DOS modifications and the encoded values for these changes.

Trk	Sct	Byte	From	To
\$00	\$0E	\$A6	13	17
		\$A7	0A	0E
\$00	\$0D	\$0D	13	17
		\$0E	0A	0E

### CONTROLLER

```
1000 REM "PENSATE CONTROLLER
1010 TK = 0:ST = 0:LT =
18:CD = WR
1020 T1 = TK: GOSUB 490:
POKE 47426,24
```

of Broderbund's 18 sector per track format. The only other disk I've seen from this series is 'Where in the USA is Carmen Sandiego?' which is in the cursed 18 sector format. By pressing Reset on the game in such a way to freeze the text screen, you'll see a message at the bottom of the screen stating 'ROLAND WAS HERE, BUT HE LEFT...'. As far as I know, the wizard who devised the 18 sector format for Broderbund is named Roland but the message confuses me because I believe that his disk protection was not used until after the release of the first Carmen Sandiego disk. Well, enough about details, let's get to the Softkey!

This disk is protected much like Captain Goodnight which was Softkeyed in Computist #82. Side 1 is encoded in 4+4 and quarter tracked as well but fortunately, once the program is loaded in, it does not access side 1 again. To make things even easier, side 2 is unprotected. By copying side 2 and boot code tracing to get the game program from side 1, this disk can be deprotected.

To begin, label a 2 sided copy disk (or 2 single sided disks). Using a normal

DOS copier, copy side 2 of Carmen Sandiego. Then, insert side 1 of the copy disk. Even though the main program is quite large, it can still be saved without having to boot code trace more than once. This can be accomplished by causing the backup disk to halt the boot once the RWTS is in memory.

CALL-151  
B64A:4C 59 FF  
INIT HELLO  
DELETE HELLO

Once this is complete, insert 'Where in the World...' and begin boot code tracing the disk. This is almost the same identical boot code as Captain Goodnight.

9600<C600.C6FFM  
96FA:98 N 9801:AD E8 C0 4C 59 FF  
9600G  
9800<800.8FFM  
9837:10 N 9847:4C 59 FF  
9801G  
982D:60 10  
9831:EA EA  
9837:00  
9847:60  
106A:A9 64  
1105:94  
9500:AD E8 C0 4C 59 FF  
9600G  
106B:04  
65E4:2C  
65ED:00 4C A0 06  
9500:A2 00 BD 00 65 9D 00 05  
:BD 00 66 9D 00 06 E8 D0  
:F1 A2 60 4C 00 04

The boot code is set to jump to \$6A0 once the game is loaded in. At \$6A0 (right now at \$66A0) we'll write a routine to save the lower 6 pages of memory as well as the \$F000 RAM card area.

66A0:A2 00 BD 00 00 9D 00 20  
:E8 D0 F7 EE A4 06 EE A7  
:06 AD A4 06 C9 06 D0 EA  
:BD 00 F0 9D 00 30 E8 D0  
:F7 EE BD 06 EE BA 06 D0  
:EF AD 82 C0 4C 59 FF

Execute \$9600 to load in the game!  
9600G

Before rebooting, we must save the code occupying the RWTS area as well as page \$8.

2600<B600.BFFFM  
2300<800.8FFM

Insert the backup disk, reboot and turn off the disk drive and restore page \$8.

C600G  
C0E8  
800<2300.23FFM

Read enable the RAM card and move bank 1 \$D000 to \$5000.

C081 C081 N F800<F800.FFFFM  
C083  
5000<D000.DFFFM

We'll use the RWTS to write \$800 through \$B5FF to tracks \$01 through \$0C.

B7EC:0C 05 FB B7 00 B5 00 00 02  
B7E1:AE  
B793G

And write Bank 2 \$DO00-\$EFFF to track \$0D and \$0E.

C08B  
B7EC:0E 0F FB B7 00 EF  
B7E1:20  
B793G

Next, reboot DOS and run a sector editor. Read track 0, sector 0 and change it so it jumps to boot stage 2 instead of the monitor.

Trk	Sct	Byte	From	To
\$00	\$00	\$4B	59 FF	00 B7

Read track 0, sector 1 and at byte 00, enter a routine to read in the game program.

Trk	Sct	Byte	From	To
00	01	00	??	20 93 B7 AD 81
				C0 AD 81 C0 AD
				8B C0 AD 8B C0
				A9 0E 8D EC B7
				A9 0F 8D ED B7
				A9 EF 8D F1 B7
				A9 20 8D E1 B7
				20 93 B7 4C 00
				23

Move the cursor to byte \$EB and enter the IOB information.

Trk	Sct	Byte	From	To
00	01	EB	??	00 0C 05 FB B7
				00 B5 00 00 01

...and at byte \$E1, enter the number of pages to load.

Trk	Sct	Byte	From	To
00	01	E1	??	AE

Write this sector back to disk and read track 3, sector 3. This sector reads into memory at \$2300. This is where we will write a memory move routine to restore all of the code into its proper place. Begin at byte 00.

Trk	Sct	Byte	From	To
03	03	00	??	A2 00 BD 00 20
				9D 00 00 E8 D0
				F7 EE 04 23 EE
				07 23 AD 04 23
				C9 26 D0 EA A9
				00 85 00 85 02
				A9 26 85 01 A9
				B6 85 03 A2 0A
				20 47 23 A9 F0
				85 03 A2 10 20
				47 23 A9 50 85
				01 A9 D0 85 03
				A2 10 AD 83 C0
				20 47 23 4C 00
				60 A0 00 B1 00
				91 02 C8 D0 F9
				E6 01 E6 03 CA
				D0 F2 60

Save this sector to the disk, and if everything was followed correctly, you should now have an unprotected copy of 'Where in the World is Carmen Sandiego?'

Softkey for...

### Spectre

#### Datamost

This Softkey will convert Spectre into a copyable disk using Super IOB and a few sector edits...but first, a brief explanation of the disk. Tracks 0-2 are normal except for modified end marks of \$DF and \$AA. Tracks \$3-22 have the same modified end marks but the address and data headers are also altered:

	Normal	Spectre
Address	D5 AA 96	C5 AA B5
Data	D5 AA AD	C5 AA BD

The disk can be copied to a normal format quite easily by using a custom made controller for Super IOB. To begin, load Super IOB and install the Spectre controller listed at the end of the article. Then run the program, choose the proper slot and drives, and format the copy disk. THE COPY DISK MUST BE FORMATTED WITH A VOLUME OF 17 or it will not work. After formatting, copy the disk and then get out your sector editor.

The RWTS on Track 0 is still looking for the altered end marks of \$DF so it must be changed to look for \$DE.

Trk	Sct	Byte	From	To
\$00	\$03	\$35	DF	DE

\$91 DF DE

The altered address and data marks are not intact in the RWTS on the disk but are stored there by a routine at \$BED7. On Track 0, Sector 8, the JMP \$BED7 should be changed to JMP \$9D84 which is where it goes after the marks are changed.

Trk	Sct	Byte	From	To
\$00	\$08	\$CF	D7	84
		\$D0	BE	9D

Finally, we have to disable a disk check that is done before the title picture loads in. This can be found on Track 6, Sector \$D. Change the JMP \$35B to BIT \$35B so it will not jump to the 'crash' routine.

Trk	Sct	Byte	From	To
\$06	\$0D	\$27	20	2C

Note that if you make copies of this backup, the copies must have a volume of 17 (\$11 hex). Most normal disk copiers copy the volume, though, so there should be no problem.

### CONTROLLER

```
1000 REM "SPECTRE CONTROLLER
1010 TK = 0:ST = 0:LT =
    35:CD = WR
1020 T1 = TK:GOSUB 490:
    POKE 47426,24:ONERR GOTO
    550
1025 IF TK < 3 THEN 1030
1026 DATA
    197,170,181,197,170,189
1027 GOSUB 180:GOSUB 200:
    RESTORE
1030 GOSUB 430:GOSUB 100:ST
    = ST + 1:IF ST < DOS THEN
    1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1:IF
    TK < LT THEN 1025
1060 GOSUB 490:GOSUB 220:TK
    = T1:ST = 0
1070 GOSUB 430:GOSUB 100:ST
    = ST + 1:IF ST < DOS THEN
    1070
1080 ST = 0:TK = TK + 1:IF
    BF = 0 AND TK < LT THEN
    1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE." :
    END
```

### Checksums

1000-\$356B	1027-\$5BCB	1070-\$CA81
1010-\$3266	1030-\$4DCA	1080-\$28C8
1020-\$1CF0	1040-\$07BD	1090-\$D7CE
1025-\$5898	1050-\$0627	1100-\$1ABF
1026-\$4611	1060-\$C284	

Softkey for...

### Superkey

#### Bytes of Learning

Of all the disks I've had sent to me through my ad in the back of Computist, three of them were from a Canadian educational software publishing company called Bytes of Learning. Their disk protection was more difficult to remove than a lot of disks that I've seen. The disk 'SUPERKEY' was sent to me by two different readers from Canada. Take note that there may be different releases of this program that the Softkey might not work on.

Bytes of Learning seems to be quite consistent with their protection methods from disk to disk. They don't necessarily use the same exact protection but the methods are similar. Even though SuperKey can be read by COPYA if you ignore epilog and checksum errors, what it is reading is nothing close to the actual

disk data. The Bytes of Learning company uses a different type of disk encoding than normal DOS 3.3 does even though the disk is in a 3.3 format. After finding the address epilogs (DE AA) the read address routine reads two additional bytes that are used in the postnibble routine to decode the sector from the raw disk data. The read translate table is also abnormal. The best way to copy the disk is to do a swap copy. Super IOB can be used for this procedure.

To start with, we need to get our hands on SuperKey's DOS. It can be found on track 0, but most of it is encoded. Instead of walking through the decode routines one at a time, we'll just copy track 0 (the only normal track on the disk) and halt the boot process at a point where the DOS has been decoded. Format a backup disk and then, using any copier that can copy individual tracks, copy track 0.

Now, run your sector editor and edit the copy disk.

Trk	Sct	Byte	From	To
\$00	\$05	\$88	20 00 BE	4C 59 FF

Write this sector back and then boot the disk.

PR#6

The DOS is now in memory in decoded form. Make a few modifications so it will work with the swap copy and then move it to the location where it will be used by Super IOB.

BC09:B8  
BC0C:B9  
BC10:B9  
BC19:EA EA  
BC52:B9  
BC63:B8  
BFFD:4C 00 BE  
1900<B800.BFFFM

Insert a slave disk with no 'hello' program and reboot. Then enter the monitor.

C600G  
CALL-151

Because SuperKey's RWTS entry point is \$BE00 as opposed to the normal \$BD00, we'll have to work modify some part of the RWTS or the Super IOB program. In the RWTS, some of the code at the end of the \$BF00 page is not needed for RWTS use. We can put a JMP \$BD00 at \$BFFD in the normal RWTS and a JMP \$BE00 at \$BFFD in SuperKey's RWTS. Then change \$3D9 to JMP \$BFFD instead of \$B7B5.

BFFD:4C 00 BD  
20FD:4C 00 BE  
3DA:FD BF

Load Super IOB and install the SuperKey Controller. Then proceed to copy the SuperKey disk. When finished, run your sector editor.

Trk	Sct	Byte	From	To
\$00	\$05	\$88	4C 59 FF	20 00 BE
		\$0C	\$12	56 EB 50 E8
		\$0F	\$3E	16 72 6D
				66 D0 3B

There is only one thing left to do now. Replace the read translate table with a normal one. Because it is encoded, it must first be read into memory, decoded, replaced, re-encoded, and finally written back to the disk.

B7E1:01 read sector  
B7EB:00 00 0D  
B7F0:00 6D 00 00 01  
B793G  
6000:A2 6A 8A 5D 95 6D 49 00 decode!  
 encode routine  
:9D 95 6D CA D0 F4 60  
6000G decode  
6D96<BA96.BAFFM replace

6000G encode  
 B7E1:01 write sector  
 B7EC:00 0D  
 B7F0:00 6D 00 00 02  
 B793G

...and you're finished!

**CONTROLLER**

```

1000 REM "SUPERKEY
    CONTROLLER
1010 TK = 1:ST = 0:LT =
    34:CD = WR
1020 T1 = TK: GOSUB 490:
    GOSUB 360: ONERR GOTO 550
1030 GOSUB 430: GOSUB 100:ST
    = ST + 1: IF ST < DOS THEN
    1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1: IF
    TK = 17 THEN 1050
1055 IF TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST =
    0: GOSUB 360
1070 GOSUB 430: GOSUB 100:ST
    = ST + 1: IF ST < DOS THEN
    1070
1080 ST = 0:TK = TK + 1: IF
    TK = 17 THEN 1080
1085 IF BF = 0 AND TK < LT
    THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE." :
    END
  
```

**Checksums**

1000-\$356B 1050-\$36DB 1085-\$722E  
 1010-\$6145 1055-\$EFAF 1090-\$1DCC  
 1020-\$4124 1060-\$BC34 1100-\$D421  
 1030-\$5725 1070-\$B431  
 1040-\$EB16 1080-\$121D

James J. Harvey MI

Softkey for...

**TEAM Series**

- People and Places
- Science and Health
- Making Choices
- General Interest

Davidson Associates (1990)

**Requirements:**

disk copier that ignores errors  
 disk editor with search command

The "TEAM Series" is a four disk Reading Comprehension program that is appropriate for Junior High and High School students.

The total program contains over a thousand reading passages and comprehension questions.

The readings cover a wide range of subjects and may be used in any order. "People and Places" - These readings cover men and women who have shaped history plus important historical events and locations.

"Science and Health" - This disk covers questions that deal with science and health, demonstrates cause-and-effect relationships, sequencing of events, and technical vocabulary.

"Making Choices" - These selections deal with issues facing young people today such as methods for handling conflict, managing money, and reducing stress.

"General Interest" - These reading passages focus on humor, mystery, and adventure.

Each program is on a 3.5" disk and naturally each disk in the series contains copy protection. Here is the procedure to make an unprotected copy of each disk.

Boot Copy II Plus 9.1 (or any later version that will copy 3.5" disks).

Straight copy (disk with format) each of the four original disks. Ignore the read error on Block 0308.

For each of the four disks:

Scan for 20 C8 0A F0 D9  
 Change to 20 C8 0A EA EA

Bytes were found as follows but may be in a different location on other versions:

**People and Places**

Blk	Byte	From	To
031D	076	20 C8 0A F0 D9	20 C8 0A EA EA

**General Interest**

Blk	Byte	From	To
031E	076	20 C8 0A F0 D9	20 C8 0A EA EA

**Making Choices**

Blk	Byte	From	To
0032	076	20 C8 0A F0 D9	20 C8 0A EA EA

**Science and Health**

Blk	Byte	From	To
0032	076	20 C8 0A F0 D9	20 C8 0A EA EA

Write the changes back to each disk.

Softkey for...

**The Botanist's Apprentice**  
 National Geographic Society

Project Classify: Plants:

Disk I - Side I & II  
 "Desert, Forest, and Wetland"

Disk II - Side I & II  
 "Farmland, Grassland, and Town"

Softkey for...

**Journey to the**  
**Age of the Dinosaurs**

National Geographic Society

Project Classify: Dinosaurs:

Disk I - Side I & II  
 "North America"

Disk II - Side I & II  
 "Europe and Asia"

Softkey for...

**The Field Zoologist**

National Geographic Society

Project Classify: Mammals:

Disk I - Side I & II

Mammals I

Disk II - Side I & II

MammalsII

The following will work for all three of the above programs:

Disk I and Disk II are copy protected on Side I. Side II of each disk is not protected.

1. Format two blank disks (for each of the above programs that you wish to copy) in ProDOS using Copy II Plus and copy a fresh PRODOS file onto side I of each disk. Set these disks aside for now.

2. Boot a DOS 3.3 disk and use COPYA to copy the disk.

**POKE 47426,24** Ignore Errors  
**RUN COPYA**

3. Copy Side I of each original disk. Copy all of the files, except PRODOS, from your copy onto the ProDOS formatted disks that you made before.

4. Rename these disks the same as the original.

5. Straight copy side II of each original disk onto Side II of your ProDOS copies.

**MECC 5.25" disk softkeys**

Softkey for...

- Vacation Nation Travel (5.25")
- Grammar Gazette (5.25")
- Time Navigator Around the World (5.25")
- Estimation Activities (5.25")
- Woolly Bounce (5.25")
- Cleanwater Detectives (5.25")
- Paper Plane Pilot (5.25")
- The Living Cell (5.25")
- Blue Grass Bluff (5.25")
- Littletown Zoo (5.25")
- Problem Solving With Nim (5.25")
- MECC

All of these MECC programs are on 5.25" disks. Here are the instructions and the location of the bytes to change for deprotecting these programs.

1. Boot Copy II Plus and format a blank disk in ProDOS (One disk for each program):

2. Copy a fresh PRODOS file to each disk and set them aside:

3. Boot a DOS 3.3 disk and use COPYA to make copies.

**RUN COPYA**  
 ctrl-reset at the prompts  
 70

**POKE 47426,24**

**POKE 48584,71**

**RUN**

4. Make the sector edits.

5. After a disk is copied and sector edited; copy all of the files (except PRODOS) from this sector edited disk onto your formatted ProDOS disk that contains a fresh PRODOS file.

6. Rename each disk the same as the original.

7. For all 5.25" MECC disks, scan for 90 03 4C XX XX 60 and change to 18 EA EA EA 60.

8. Now make the correct edit for the disk you are copying.

**Vacation Nation Travel**

Trk	Sct	Byte	From	To
05	09	A6	90 03 4C DA 89 60	18 EA EA EA EA 60

**Grammar Gazette**

Trk	Sct	Byte	From	To
05	09	8D	90 03 4C 77 91 60	18 EA EA EA EA 60
07	0A	84	90 03 4C 52 03 60	18 EA EA EA EA 60

**Time Navigator Around the World**

Trk	Sct	Byte	From	To
05	09	8D	90 03 4C 4B 75 60	18 EA EA EA EA 60
07	0A	98	90 03 4C 52 03 60	18 EA EA EA EA 60

**Estimation Activities**

Trk	Sct	Byte	From	To
01	09	8D	90 03 4C 86 8D 60	18 EA EA EA EA 60
03	0A	9A	90 03 4C 52 03 60	18 EA EA EA EA 60

**Woolly Bounce**

Trk	Sct	Byte	From	To
05	07	66	90 03 4C A2 91 60	18 EA EA EA EA 60
17	02	84	90 03 4C 52 03 60	18 EA EA EA EA 60

**Cleanwater Detectives**

Trk	Sct	Byte	From	To
01	09	66	90 03 4C AE 89 60	18 EA EA EA EA 60
03	0B	28	90 03 4C 52 03 60	18 EA EA EA EA 60

**Paper Plane Pilot**

Trk	Sct	Byte	From	To
05	09	8D	90 03 4C D2 86 60	18 EA EA EA EA 60
07	0A	98	90 03 4C 52 03 60	18 EA EA EA EA 60

**The Living Cell**

Trk	Sct	Byte	From	To
01	09	66	90 03 4C 02 82 60	18 EA EA EA EA 60
07	0C	A4	90 03 4C 52 03 60	18 EA EA EA EA 60

**Problem Solving With Nim**

Trk	Sct	Byte	From	To
01	09	8D	90 03 4C 17 80 60	18 EA EA EA EA 60
03	0A	84	90 03 4C 52 03 60	18 EA EA EA EA 60

**Blue Grass Bluff**

Trk	Sct	Byte	From	To
01	09	66	90 03 4C A1 7F 60	18 EA EA EA EA 60
03	0A	A4	90 03 4C 52 03 60	18 EA EA EA EA 60

**Littletown Zoo**

Trk	Sct	Byte	From	To
05	09	A6	90 03 4C 71 8A 60	18 EA EA EA EA 60

**MECC 3.5" disk softkeys**

Softkey for...

- Vacation Nation Travel (3.5")
- Estimation Activities (3.5")
- The Living Cell (3.5")
- Problem Solving With Nim (3.5")
- Woolly Bounce (3.5")
- Time Navigator Around the World (3.5")
- Grammar Monsters (3.5")
- Cleanwater Detectives (3.5")
- Paper Plane Pilot (3.5")
- Littletown Zoo (3.5")
- Bluegrass Bluff (3.5")
- Grammar Gazette (3.5")
- MECC

A new series of MECC programs, that are ProDOS based and are on 3.5" and 5.25" disks, have been made available to me. These were released in the latter part of 1991. Here are the sector edits to bypass the protection for the 3.5" disks.

1. Copy each disk using Copy II Plus and ignore the error on Blk 0308.

2. Make the appropriate block edit. Don't forget to write the changes back to each disk as you sector edit them.

3. Rename each disk the same as the original.

**Vacation Nation Travel**

Blk	Byte	From	To
002D	0A6	90 03 4C DA 89 60	18 EA EA EA EA 60

**Estimation Activities**

Blk	Byte	From	To
000C	08D	90 03 4C 86 8D 60	18 EA EA EA EA 60
001B	19A	90 03 4C 52 03 60	18 EA EA EA EA 60

### The Living Cell

Blk	Byte	From	To
000C	066	90 03 4C 02	18 EA EA EA EA 60
003B	1A4	90 03 4C 52	18 EA EA EA EA 60

### The Living Cell

Blk	Byte	From	To
000C	08D	90 03 4C 17	18 EA EA EA EA 60
001B	184	90 03 4C 52	18 EA EA EA EA 60

### Woolly Bounce

Blk	Byte	From	To
002D	066	90 03 4C A2	18 EA EA EA EA 60
00BF	184	90 03 4C 52	18 EA EA EA EA 60

### Time Navigator Around the World

Blk	Byte	From	To
002C	08D	90 03 4C 4B	18 EA EA EA EA 60
003B	198	90 03 4C 52	18 EA EA EA EA 60

### Grammar Monsters

Blk	Byte	From	To
00B2	0A6	90 03 4C 7B	18 EA EA EA EA 60
00CA	184	90 03 4C 52	18 EA EA EA EA 60

### Cleanwater Detectives

Blk	Byte	From	To
002C	006	90 03 4C AE	18 EA EA EA EA 60
003B	028	90 03 4C 52	18 EA EA EA EA 60

### Paper Plane Pilot

Blk	Byte	From	To
002C	08D	90 03 4C D2	18 EA EA EA EA 60
003B	198	90 03 4C 52	18 EA EA EA EA 60

### Littletown Zoo

Blk	Byte	From	To
002C	0A6	90 03 4C 71	18 EA EA EA EA 60

### Bluegrass Bluff

Blk	Byte	From	To
000C	066	90 03 4C A1	18 EA EA EA EA 60
001B	1A4	90 03 4C 52	18 EA EA EA EA 60

### Grammar Gazette

Blk	Byte	From	To
002C	08D	90 03 4C 77	18 EA EA EA EA 60
003B	184	90 03 4C 52	18 EA EA EA EA 60

### Rodney W Peterson

☺ I deprotected Shanghai so That I could put it on my hard drive but it doesn't beep at the three second level when playing "Challenge". Try it and you will see what I mean. Is there a fix that corrects this?

Are there any deprotects for MECC programs? These are mainly educational programs.

One last thing, do you have an address for Brian A. Troha? I have written to him recently but have not received a reply.

It's for sure that you are not a regular subscriber. Softkeys for MECC software are a regular fixture at *Computist*. What program

are you trying to backup? There are too many titles for me to just send you everything. I haven't heard from Brian in some time. He's no doubt busy with his own affairs. Maybe he'll read this and drop us a line. *RDEXed*

Jack Nissel PA

### Bug in PSAT/SAT Analogies

### Bug in SAT Word Attack Skills

### Bug in PSAT Word Attack Skills

In issue 63 the controller for PSAT/SAT Analogies, SAT Word Attack Skills, and PSAT Word Attack Skills. Line 10010 should be:  
 10010 IF PEEK(6400) <> 162  
 THEN PRINT CHR\$(4) "BLOAD  
 RWTS.PSAT WORD ATTACK  
 SKILLS,A\$1900"

### Bug in Matchmaker World Geography Facts

In issue 62 the controller for Matchmaker World Geography Facts Line 10010 should be:  
 10010 IF PEEK(6400) <> 162  
 THEN PRINT CHR\$(4) "BLOAD  
 RWTS.MATCHMAKER WGF,  
 A\$1900"

### Bug in Algebra vol 3 & 4

In issue 63 the controller for Algebra Volume 3 & 4. Line 10010 should be:  
 10010 IF PEEK(6400) <> 162  
 THEN PRINT CHR\$(4) "BLOAD  
 RWTS.ALGEBRA,\$1900"

I'm sorry if these errors caused any-one problems.

Softkey for...

- Fact or Opinion:**  
**Smart Shopper Blue Level**  
**Fact or Opinion:**  
**Smart Shopper Red Level**  
**Cause & Effect:**  
**Mountain Climbing Blue Level**  
**Drawing Conclusions:**  
**Chief of Detectives Red Level**  
**Drawing Conclusions:**  
**Chief of Detectives Blue Level**  
**Reading for Detail:**  
**Race Track Blue Level**  
**Getting the Main Idea:**  
**Around the World Blue Level**  
*Learning Well*

#### Requirements:

The original disk(s)  
 A blank initialized disk for each title  
 A blank initialized slave disk  
 Super IOB v1.5  
 A way to reset into the monitor

These titles can be deprotected by using Super IOB with the Swap Controller. Use the RWTS of the protected disk to read the original disk then use a normal RWTS to write the information back to your blank disk.

1. INITialize your blank disk(s)  
 INIT HELLO  
 DELETE HELLO
2. Boot your original disk and at the Applesoft prompt reset into the monitor.

3. Move the RWTS to a safe place by entering:

1900<B800.BFFFFM

4. Put in your slave disk and boot it by entering:

C600G

5. After the disk boots and the Applesoft prompt appears insert your Super IOB disk and save the RWTS to it by entering:

BSAVE RWTS.LEARNING WELL,  
 A\$1900,L\$800

6. Install the controller into Super IOB, run it and copy your original disk to your blank disk. Answer NO when asked if you want to INITIALize the blank disk.

### CONTROLLER

```
1000 REM LEARNING WELL
1010 TK = 3:ST = 0:LT =
    35:CD = WR
1020 T1' = TK: GOSUB 490:
    GOSUB 360: ONERR GOTO 550
1030 GOSUB 430: GOSUB 100:ST
    = ST + 1: IF ST < DOS THEN
    1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1: IF
    TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST =
    0: GOSUB 360
1070 GOSUB 430: GOSUB 100:ST
    = ST + 1: IF ST < DOS THEN
    1070
1080 ST = 0:TK = TK + 1: IF
    BF = 0 AND TK < LT THEN
    1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE" :
    END
10010 IF PEEK(6400) < > 162
    THEN PRINT CHR$(4) "BLOAD
    RWTS.LEARNINGWELL,
    A$1900"
```

### Checksums

```
1000-$356B 1040-$6342 1080-$6CA2
1010-$3565 1050-$ABA3 1090-$9DCA
1020-$6170 1060-$20C0 1100-$C97D
1030-$7771 1070-$28C5 10010-$2E37
```

Softkey for...

- Reading for Detail:**  
**Race Track Red Level**  
**Sequence:**  
**What Comes First Red Level**  
**Sequence:**  
**What Comes First Blue Level**  
**Getting the Main Idea:**  
**Around the World Red Level**  
**Inference:**  
**School Days Red Level**  
**Inference:**  
**School Days Blue Level**  
**Following Directions:**  
**Behind the Wheel Red Level**  
**Context Clues:**  
**Hidden Treasure Red Level**  
*Learning Well*

#### Requirements:

The original disk(s)  
 A blank disk for each title  
 DOS 3.3 system disk  
 Any sector editor

All of these titles can be deprotected in the following manner.

1. Boot your DOS 3.3 system master and at the J prompt enter:  
 POKE 47426,24 ignore checksum and  
 epilogs  
 RUN COPYA
2. Copy your original disk to your blank disk.

3. Make the following sector edit to your disk.

Trk	Sct	Byte	From	To
00	03	42	38	18

4. Write the sector back to the disk.

### BBS #30

Gate, Space Fox, and HoverBlade are all available from LRO Computer Sales at 1-800-869-9152. The prices are \$45, \$49, and \$29 respectively. The best of the three by far is GATE - nice graphics, some good puzzles, and a number of surprises. The first two were produced by the Toolbox folks in France; they also produced Sensei GS, a karate game that is pretty cool (nice graphics, sound and animation), but it will reportedly not be released due to the weak GS software market. It is only to be found on "illicit" BBS's (whatever those are). Meanwhile, Secret of Bharas has just been released by Victory Software, the people who did 2088:Cryllan Mission I & 2. It reportedly has similar graphics to CM2, and is played similarly.

### BBS #181

☺ Does anyone have a crack for the 5.25 version of Children's Writing and Publishing Center for the Apple? I could really use some help on that one!

☺ Does anyone know if there is any way to get Apple IIGs programs, that apparently run on ROM 1 only, to run on a ROM 3 machine?

I would like to thank Jack Moravetz for all his help. It's great to know that there are those among us who are able and willing to help out. Thanks Jack!

Here's my address if you can help me with the above questions. (I don't get on this board too often.)

Ron Powers  
 1607 Skeels Ave.  
 Eau Claire, WI 54701  
 (715) 834-8676

### BBS #61

☺ I have Skate or Die for the IIGs, using the old ROM version 1. It doesn't work with the version 3 ROMs. When I called Electronic Arts and informed them of this problem, they said, "They couldn't help me, and that they are not going to release a new version to accommodate the new ROM version 3." Is there any way to rewrite the code so it could be used on the new ROM version 3?

### BBS #19

Softkey for...

### Superstar Ice Hockey

After someone telling me that the softkey I presented for Superstar Ice Hockey was, in not so many words, ridiculous, I looked for other ways of deprotection. Since I don't own the program, this was tricky, but a softkey originally appeared in issue 71. For those who don't have that issue but would still like to play the game without the manual checks, here is the softkey as it appeared. The credit for this one goes to Brian Troha and Stephen Lau.

Blk	Byte	From	To
\$9	\$5F	20 A3 1A	AD A3 1A
\$B	\$4	D0 13	EA EA
	\$C	D0 0B	EA EA

\$C \$11A D500 D400  
\$38 \$B7 D012 8038  
\$A6 \$B7 D012 8036

I would like to take this opportunity to thank Mr. Troha, Mr. Lau, and all the others who took the time to explain how and why they did certain things while deprotecting. It helped me (and I'm sure others) a lot.

---

Jack Moravetz

Softkey for...

### Skills Bank II

?

Skills Bank II is an educational package containing teacher disks, class record disks, student disks, and various levels of lesson disks. When the lesson disks are copied, the student disk accessing them will say that it is a copy and quit. The protection is on the student disk in a file called TUA.OVR. Using a block or sector editor (I use Glen Bredon's Block Warden) to follow the file TUA.OVR, search for the bytes \$20 41 77 and replace the \$20 with \$60 and write the block back to the disk. It is in the file at least twice. You need to place the \$60 at the first occurrence.

---

Tim Furry

### Notes on

#### Test Drive II cars disks

I just ran across this today. If you want to have only one data disk for your cars (supercars and musclecars), here's what to do: Copy all of your car files from one disk to the other (i.e. from muscle cars to super cars or vice versa). Boot up a source code editor (such as ORCA/M) and edit the file "CARS.DAT" so that it includes the following:

CAMA Camaro\_ZL-1  
DODG Dodge\_Charger  
GOAT Pontiac\_GTO  
GT50 Mustang\_GT500  
STNG Corvette\_Stingray  
That's all there is to it!

---

Unknown

IBM Softkey for...

### The Faery Tale

?

A really easy doc check, it can be bypassed in many ways. I put the alterations as deep as possible to avoid any possible complications. The protection is in FTC.EXE and FTE.EXE which are the CGA and EGA versions.

For Norton users, search FTC.EXE and FTE.EXE for the byte pattern 85 C0 74 23 9A and change the 74 23 to EB 2E.

DEBUG method. DEBUG is assumed to be in the current path or directory.

First the CGA version.  
REN FTC.EXE FTC.ZAP  
DEBUG FTC.ZAP  
E 352F EB 2E

W *to save it*  
Q *to quit to DOS*  
REN FTC.ZAP FTC.EXE

Now for the EGA version.  
REN FTE.EXE FTE.ZAP  
DEBUG FTE.ZAP  
E 3625 EB 2E

W *to save it*  
Q *to quit*  
REN FTE.ZAP FTE.EXE

Okay, that's it. Just hit return when asked for the words from the map.

IBM Softkey for...

### Gauntlet!

?

Use Norton to search the GPROG.EXE file for B4 1C (offset should be hex 7E51). Change B4 1C CD 21 8A 07 3C F8 75 to 90 90 E8 EF 01 31 C0 EB 04. Search for: 02 B8 (offset should be hex 8045). Change 02 B8 00 3F CD 21 72 07 A1 to 0E 1F BB EF FF B8 11 00 A3. Search for 03 06 (offset should be 8050) and change to 89 1E.

IBM Softkey for...

### Grave Yardage

#### Activision

Use PCtools or other program and edit GY.EXE. Go to sector 72, offset 373 and change 9A 34 11 to E9 90 00. That's it!

IBM Softkey for...

### Gun Boat

#### Accolade

Use Norton and search file GB.EXE (141,333 03-09-90 1:12p) for 75 03 E9 8A (should be at offset 27FFH). Change 75 03 to 90 90. Now, when you're asked to enter an answer, hit any number and the RETURN key.

IBM Softkey for...

### Astrilis

#### Shaman Games

Use debug to load astrilis.com  
DEBUG ASTRILIS.COM  
E 79D1  
XXXX:79D1 75.EB  
E 79EB  
XXXX:79EB 75.90 F2.90  
W

That's all there is to it. If you find that the address doesn't match then search for 3A E0 75 11 4F and change to 3A E0 EB 11 4F.

3AE0 CMP AH,AL  
7511 JNZ 79E4 ;changes JNZ to JMP  
4F DEC DI

Search for 3C 4D 75 F2 BF 00 01 and change to 3C 4D 90 90 BF 00 01.

3C4D CMP AL,AD  
75F2 JNZ 79DF ;changes JNZ to NOP  
BF0001 MOV DI,0100

IBM Softkey for...

### Ancient Art of War

#### Broderbund

1. Make a copy of the original distribution diskette using the DISKCOPY command. Put your original away, and work on the copy. Steps 2 through 6 assume that a disk containing DEBUG.COM is in drive A: and your working WAR disk is in drive B:

2. From the A> prompt, type:  
RENAME B:WAR.EXE WAR.SAV

Followed by, of course, hitting the Return key.

3. Now, start up DEBUG with:  
DEBUG B:WAR.SAV

4. From the '-' DEBUG prompt, enter:  
E 8BD8

xxxx:8BD8 CD-F8 13-90 73-EB

The 'xxxx' will vary for your system, but if everything from the colon (:) onward doesn't look the same, DON'T continue — either you have another version, or something is wrong.

DEBUG will display: xxxx:8BD8 CD-. Enter 'F8', and hit the space bar. DEBUG will display: 13- Enter '90', and hit the space bar. DEBUG will display: 73- Enter 'EB', and hit the space bar, and then Return.

5. Now, from the '-' DEBUG prompt, enter 'W', and hit the Return key. This will write the file back to disk.

6. Enter 'Q' to leave DEBUG, and from the A> prompt, enter:  
RENAME B:WAR.SAV WAR.EXE

Your working copy of WAR will now run, and can, itself, be backed up using the DOS DISKCOPY or COPY commands.

Note: if you use WAR on a fixed disk, you will not be able to save a game in progress, due to a small bug — WAR tests the default disk for the presence of a file named "M", instead of the selected data disk, the intent being to prevent saving a game on the same disk as that file. To disable this test, after completing step 4, patch the following location:  
E 3B1D  
xxxx:3B1D 0B-31

Replaces the '0B' with '31'. This will allow you to save a game on a floppy disk in the A: drive; Note, however, that this removes a small margin of safety (if you save a new campaign and a game to the same disk, the game reloader will get confused).

IBM Softkey for...

### Aardvark McGraw-Hill Financial Software

#### JRE

This information is provided to allow legal owners to liberate themselves from the dependence on a single floppy diskette.

Caution: It is highly recommended that you use this protection scheme on a backup copy of your software, in case you have a problem. Under no circumstance should you use this routine on an original floppy. I did the routine on a copy created by Quaid Software's "Copyright" (A program I highly suggest you purchase.)

Using the below patches, you will be able to make unlimited backups of the floppy, but more importantly, be able to boot the entire system off a hard disk.

The copy protection method used on this program expects an unusually formatted Track 7. The easiest way around this was to bypass the entire 'Verify Track 7' routine.

I assume that you are familiar with DEBUG (found on the 2nd DOS diskette). Everything displayed below in upper-case will be supplied by the computer; you must type everything that is in lower-case. Note that the '-' is DEBUG's prompt and should not be typed. Place the program diskette in Drive A, and Debug in Drive B.

A> ren atl.exe atl *Debug can't save an EXE file*

A> b:debug atl *Enter Debug*

-s 0 fff0 81 e0 ff 00 89 86 *Search for the routine to be patched.*

YYYY:21AF *DEBUG should return this number. If not, you have a different vers & should quit.*

-e 21af e9 79 00 *Eliminate Floppy Verify routine.*

-w *Save the program with the changes.*  
Writing 6800 bytes  
-q *Exit DEBUG*  
A> ren atl atl.exe *REName the file to it's original name*

YYYY:CF3D *If DEBUG does not return this number, add another 1000 and retry.*

-e yyyy:cf3d 90 90 90 90 90 *Tell program to skip copy prot. routine.*

-w *Save the program with the changes.*  
Writing 1E000 bytes *DEBUG displays this message.*

-q *Exit DEBUG*

That's It. Use 'COPY A:\*.\* B:' to move all files to a formatted diskette and you will have the whole system unprotected and copyable.

If you are like me, and you follow the above directions verbatim, you will never succeed in unprotecting the AARD-VARK software. In the step above where you are waiting for the computer to return CF3D, it does NOT have to be CF3D. On my software, it was something totally different. So...the proper step, in my case, was to take the value returned: YYYY:CF3D <or whatever value appeared>, and type:

-e yyyy:CF3D *(or whatever value appeared above) Put 90 90 90 90 90 in place of what the program tells you above.*

IBM Softkey for...

### ATI Training Software

#### JRE

This information is provided to allow legal owners of the above software to liberate themselves from the dependence of a single floppy diskette.

Using the below patches, you will be able to make unlimited backups of the floppy, but more importantly, be able to boot the entire system off a hard disk.

The copy protection method used on this program expects an unusually formatted Track 2. The easiest way around this was to patch the 'Verify Track' routine.

When you run the unprotected copy of Chart Master, you will get an error message. You can ignore this as just press <ENTER>. I'm sure that this can be bypassed also, but after running the program, I was disappointed and decided not to spend any more time debugging it. If you can eliminate this shortcoming, feel free to modify this routine. Please send changes to ComputistRDEX editor.

I assume that you are familiar with DEBUG (found on the 2nd DOS diskette). Everything displayed below in upper-case will be supplied by the computer; you must type everything that is in lower-case. Note that the '-' is DEBUG's prompt and should not be typed. Make a copy of the file 'ATI.EXE'; never work with you only copy. Place the program diskette in Drive A, and Debug in Drive B.

A> ren atl.exe atl *Debug can't save an EXE file*

A> b:debug atl *Enter Debug*

-s 0 fff0 81 e0 ff 00 89 86 *Search for the routine to be patched.*

YYYY:21AF *DEBUG should return this number. If not, you have a different vers & should quit.*

-e 21af e9 79 00 *Eliminate Floppy Verify routine.*

-w *Save the program with the changes.*  
Writing 6800 bytes

-q *Exit DEBUG*  
A> ren atl atl.exe *REName the file to it's original name*

That's It. Use 'COPY A:\*.\* B:' to move all files to a formatted diskette and you will have the whole system unprotected and copyable.

# unClassifieds

## How to place an UnClassified Ad

Send a typed sample copy with appropriate instructions. (If possible, send text on a 5.25" Apple format disk.) Use up to 40 characters per line, we will adjust word wrap.

**Special Graphics Instructions:** The first three words of the first line are printed in bold for free. If you want other words bolded, use 5 characters less per line. Use 10 characters less per line if you have a lot of uppercase bold letters. Bold letters are wider than normal. If the typed copy does not show bold, circle the words you want bolded and, on the side, write BOLD. If you want a line centered, write CENTER next to that line. There is no charge for centering any line.

You must check your ad for errors, the first time it runs. Errors on our part will be corrected, then, for free. Errors or changes on your part will be charged a \$5 processing fee.

### ★★★★ New Rates (per line) ★★★★★

Computist club member .....25¢  
All others .....35¢

The minimum order is \$5.

- Our liability for errors or omissions is limited to the cost of the ad.
- We reserve the right to refuse any ad.
- Washington state residents add 7.8% sales tax.
- Send a check or money order (funds drawn on US bank only) for the entire amount to:

**COMPUTIST unCLASSIFIEDS**  
33821 East Orville Road  
Eatonville, WA 98328

# WANTED

## "Most Wanted List" Software

### Need help to deprotect a disk

Softkey hobbyist is interested in acquiring copy protected software to deprotect. Good track record, many successful attempts. Original disk will be returned along with softkey for COMPUTIST. Especially interested in older software (pre-1988) but will give any disk a shot. I'm especially interested in:

Drol---Broderbund  
Serpentine---Broderbund  
Spare Change---Broderbund  
Wings of Fury---Broderbund  
Star Cruiser---Sirius  
Space Eggs---Sirius  
Falcons---Picadilly  
Microwave---Cavalier

System: Apple IIe, 128K. Send disk to:

**Rich Etarip**  
824 William Charles, Apt #2  
Green Bay, WI 54304

### Lode Runner Fun Club

1900 optional Lode Runner screens available for exchange with your original screens.

**Toshikazi Yamamoto**  
#706 Minami-Ikebukuro-Jutaku  
1-13-21 Minami-Ikebukuro, Toshima  
Tokyo 171 Japan

### Help! Looking for

a working copy of Eureka & The Graphics and Sounds Development System from the book "Eureka" by Timothy Knight published by Baen Computer Books. Also seeking a copy of Superworks (Appleworks for IBM).

**CK Bevan**  
PO Box 941072  
Atlanta GA 30341-0072

## Wanted! Information

on Cracking IBM protection. Also software like Copywrite, Locksmith PC, Central Point Software's PC Option board and PC Watch. Also send list of Apple II items for sale.

**Joe Torzewski**  
51625 Chestnut Road  
Granger IN 46530

## RDEX Contributors

Vince Andrews.....	7
BBS #181 .....	21
BBS #19 .....	21
BBS #30 .....	21
BBS #61 .....	21
Brian Coombs .....	16
Rich Etarip .....	17, 21
Tim Furry .....	22
James J. Harvey .....	20
Jeff Hurlburt .....	4
Krakowicz .....	15
Jack Moravetz.....	22
Unknown .....	22

## Apple Most Wanted

63 Alcon.....	Taito
74 Algebra Shop .....	Scholastic
63 Alien Mind.....	PBI Software
73 American History Explorer Series .....	Mindscape
75 Anchorman .....	Virginia Reel
74 Animals of the Past .....	Focus Media
72 Ankh .....	Datamost
73 Ant Farm .....	Sunburst
67 Aquatron .....	Sierra
63 Bad Street Brawler .....	Mindscape
73 Bank Street Beginner's Filer .....	Sunburst
73 Bank Street School Filer .....	Sunburst
63 Beyond Zork .....	Infocom
65 Bilestoad .....	Datamost
69 Blue Powder - Grey Smoke .....	Grade
74 Birds - Trees & Flowers .....	Focus Media
63 Border Zone .....	Infocom
67 Bouncing Kamungas .....	Penguin
66 Boxing.....	?
65 Bureaucracy .....	Infocom
67 C'est La Vie .....	Adventure International
69 Caverns of Callisto.....	Origin
69 Checker .....	Odesta
69 Chess 7.0.....	Odesta
81 Chessmaster 2100 IIe .....	Software Toolworks
75 Clue Master Detective .....	Leisure Genius
63 Cosmic Relief .....	Datasoft
65 Crime & Punishment .....	Imagic
81 Crosscountry USA School Edition .....	Didatech
69 Crossword Magic v4.0 .....	?
69 Cybernation .....	Nexa Corp.
74 Decimal Dungeon.....	Unicorn
74 Decisions Decisions: Colonization v1.0 .....	Tom Snyder Productions
69 Delta Squadron .....	Nexa Corp.
67 Desecration .....	Mind Games
66 Disk Optimizer System.....	Nibble Notch
65 Dondra .....	Spectrum Holobyte
69 Dragon Eye .....	Epyx
69 Dueling Digits .....	Broderbund
68 D&D-Master Assistant vol2 .....	SSI
62 DROL .....	Broderbund
74 Exploring Tables & Graphs Level 2 (SU) .....	Weekly Reader
67 Evolution .....	Sydney

67 Falcons .....	Picadilly
68 Factastics Trivia .....	Daystar
75 Final Frontier .....	Softsmith
73 Fisher's Cove .....	Tom Snyder Productions
69 Flt Wars .....	Sirius
74 Fraction Action .....	Unicorn
69 Gemstone Healer .....	SSI
73 Geometric Supposer (the).....	Sunburst
66 GEOS .....	Berkley Softworks
63 Gladiator .....	Taito
73 Goodell Diamond Caper .....	Tom Snyder Productions
66 GradeBuster 1 2 3 .....	Grade Buster
61 Gutenberg Sr. ....	Micromation LTD.
65 Halls of Montezuma.....	Electronic Arts
67 High Orbit .....	Softsmith
67 Horizon V.....	Softsmith
75 Hunt for Red October GS.....	Datasoft
69 Impossible Mission .....	Epyx
62 Indoor Sports .....	Mindscape
68 Infocomics .....	Infocom
66 Jane .....	?
63 Joker Poker .....	Mindscape
72 Kabul Spy .....	Sirius
68 Kingdom of Facts .....	Santa Barbara/Thunder Mountain
75 Kobayashi Alternative (The) .....	Simon & Schuster
72 Lane Mastodon .....	Infocom
67 Lancaster .....	SVS
72 Laser Force (Iigs) .....	Britannica
81 Last Ninja, The (Ile) .....	Activision
75 L.A. Land Monopoly.....	Softsmith
66 Legacy of the Ancients .....	Electronic Arts
65 Lost Tomb .....	Datasoft
81 M-ss-ing L-nks: Classics old & new .....	Sunburst
74 Mammals - Reptiles & Amphibians .....	Focus Media
65 Manhunter New York Iigs .....	Sierra On Line
65 Mavis Beacon Teaches Typing (gs) .....	Software Toolworks
73 McGraw-Hill Problem-Solving Lvl 5&6 .....	Tom Snyder Productions
67 Microwave .....	Cavalier
73 Mind Castle I.....	MCE Inc.
63 Modem MGR .....	MGR Software
68 Mr. Pixel's Cartoon Kit.....	Mindscape
73 Mystery of Hotel Victoria .....	Tom Snyder Productions
63 National Inspirer ...	Tom Snyder Productions
75 Neptune .....	Softsmith
66 Observatory (The) .....	Mindscape
74 Ocean Life .....	Focus Media
66 Odin .....	Odessta
63 Operation Wolf .....	Taito
69 Phantasie II.....	SSI
67 Phantoms 5 .....	Sirius
67 Pig Pen .....	Datamost
74 Plants & Animals of the Desert.....	Focus Media
75 Prince of Persia (5.25").....	Broderbund
67 Project: Space Station.....	Avantage
75 Promethean Prophecy (The) .....	Simon & Schuster
67 Pulsar II .....	Sirius
68 Pure Stat Basketball .....	?
62 Quadratic Equations II .....	Olympus Educational Software
81 Quarter Mile Ile .....	?
68 Rails West .....	SSI
63 Renegade .....	Taito
67 Rescue Raiders.....	Sir Tech
63 Rocket Ranger (Iigs) .....	Cinemaware
69 Roundabout .....	Datamost
75 Russki Duck.....	Softsmith
63 S.D.I. (Iigs) .....	Cinemaware

62 Sea Stalker .....	Broderbund
67 Serpentine .....	Broderbund
63 Sky Shark .....	Taito
63 Sound Song & Vision ...	Advanced Software
67 Space Ark .....	Datamost
62 Spare Change .....	Broderbund
62 Speedy Spides .....	Readers Digest
67 Star Cruiser .....	Sirius
63 StickyBear Math: Add & Subtract .....	Optimum Resources
68 Stickybear GS Versions 3.5 .....	Xerox
67 Succession .....	Picadilly
61 Superstar Indoor Sports .....	Mindscape
74 Surveys Unlimited .....	Mindscape
68 Talking Text Writer GS .....	Scholastic
68 Tangled Tales .....	Origin Systems
69 Tetris (Ile) .....	Spectrum Holobyte
72 Theatre Europe .....	PBI
74 The Other Side v2.0 .....	Tom Snyder Productions
81 Think Quick! v1.2 .....	Learning Company
65 Thunder Chopper .....	?
63 Ticket to Washington D.C. .....	Blue Lion Software
74 Time Explorers .....	Gameco
74 Time Liner v1.1 ....	Tom Snyder Productions
68 Tomahawk (Iigs) .....	Datasoft
69 Track Attack.....	Broderbund
68 Triad .....	Thunder Mountain
72 Triango (Iigs) .....	California Dreams
68 Trinity .....	Infocom
73 Unicorn 5.25" software .....	Unicorn
73 Vincent's Museum .....	Tom Snyder Productions
68 Volcanoes v1.8 .	Earthware Comp. Services
66 War in the Middle Earth .....	Melbourne
63 Wings of Fury .....	Broderbund
63 Wizardry:Return of Werda .....	Sir-Tech.
65 Works (the) .....	First Star Software
67 Zenith .....	Softsmith

## IBM Most Wanted

84 Ace of Aces .....	Accolade
84 Bar Games .....	Accolade
84 Colony .....	Mindscape
84 Don't Go Alone .....	Accolade
75 Empire .....	Intersil
84 Final Orbit .....	Innerprise
72 GBA Championship Football .....	Electronic Arts
68 Graphitti .....	George Best Phillips Academy
63 Heros of the Lance .....	SSI
84 Hardball II .....	Accolade
84 Harmony .....	Accolade
84 Hat Trick.....	Capcom
84 Ishido .....	Accolade
84 Jetfighter .....	Velocity
84 John Elway's Quarterback .....	Melbourne House
84 M1 Tank Platoon.....	Microprose
84 Monty Python's Flying Circus .....	Mastertronic
72 Operation Wolf .....	Taito
84 Outrun .....	Sega
84 Phantasm .....	Exocet
86 Pirates! .....	Microprose
84 Powerdrome .....	Electronic Arts
72 Radio Baseball .....	Electronic Arts
84 Sim City .....	Maxis
84 Stormovik .....	Electronic Arts
86 Sword of the Samurai .....	Microprose
84 Test Drive III: The Passion .....	Accolade
84 Third Courier .....	Accolade
84 Troika .....	Paragon
84 Wayne Gretzky Hockey 2 .....	Bethesda
84 World's Greatest Baseball Game .....	Epyx/Key Punch

#79• The Product Monitor• *Bitkeys*: Kabul Spy• *Softkeys*: ABM• Algebra 1-6 Cause and Effect• Chemistry: Series I• Computer Generated Mathematics Vol. 2• Cribbage• Designer Puzzles• Dungeon Master Assistant Vol. 2• Economics• Genesis• Gin King• Go• Graphmaster• Hard Hat Mack• Hi Res Computer Golf• Integer Arcade• Laser Bounce• Mammals Reptiles and Insects• Master Grades• Mickey's Crossword Puzzle Maker• Mind Benders• Missing Links• Non-Western Cultures• RoboCOP• Safari Search• SAT Score Improvement Series• Special Product and Algebraic Factors• Stickybear GS Talking series Talking Alphabet• Talking Opposites• Talking Shapes• Task Force• Teacher's Toolkit version 3.1• The Great Knowledge Race• The History of Europe• The Solar System• The Time Tunnel• Thief• TrianGO• US History• Wasteland• Water and Weather• Who Am I?• Word Problems for Algebra• Worksheet Generator• Writing Chemical Formulas• Your Body• Your Body: Series II• *Playing Tips*: Baneful Tales• Elite• *Mac Features*: Mac Hard Disk Ejection Fix• *Mac Softkeys and other Patches*: ABCBase• Animation Toolkit1• Aztec C 1.0• Aztec C version 1.00c• Championship Boxing• Chart• Checkminder• Cutthroats• Cutthroats alternate• Deja Vu• Desk Toppers• Dollars & Sense• Dollars & Sense alternate• Electric Checkbook• Excel• Excel alternate fix• Fact Finder 1.0• Factfinder• Fahrenheit 451• Feathers & Space• File• FileMaker• Filevision• Filevision alternate• Forecast• Frogger• FunPak• Gato• Grid Wars• Griffin Terminal• Haba-Comm• Haba-Comm alternate• HabaCheckMinder• Habadex 1.1• Harrier Strike Mission• Hayden Speller• Hayden Speller alternate• Hippo^C Level 1• Hitchhiker's alternate• Hitchhiker's Guide to the Galaxy• Home Accountant• Legacy• Lode Runner• Mac Fortran• Macattack• MacChkrs/Rvrsi• MacCommand• MacDraft 1.0• MacDraft 1.1• MacGammon/Cribbage• MacJack/Poker II• MacLabeller• MacMatch• MacPascal (version 1.0)• MacPoker• MacType• Master Type• Master Type alternate• Mouse Stampede• Multiplan alternate• Multip-

lan version 1.02• OverVue• PageMaker• PageMaker 1.0• Pensate• PFS• PFSFile/Report• PFS version A.03• Real Poker• Rogue• Sargon III• SkyFox• Smooth Talker• The Quest• Think Tank• ThinkTank 1.1• ThinkTank 128• ThinkTank 512• Transylvania• Triple Play 1.0• Trivia Arcade• Trivia Fever• Typing Intrigue• Ultima II• Ultima III• VideoWorks 1.0• WellTris• Winter Games• Xyphus• *Features, Notes & such*: COPYA-able Questron II• How to make Thief into a BRUNable file• How to run Task Force on your hard drive• Making Genesis into a single BRUNable file• Making Hard Hat Mack into a single BRUNable file• Making PLATO software run on the Enhanced //e• Multi-Column Print Utility (MCP)• Notes on Battle Chess• Notes on Silent Service GS• Notes on Wildcard II card• Object Module Format (OMF)• ORCA/Disassembler Scripts• ORCA/Disassembler utilities• Other Notes• Running Teacher's Toolkit v3.1 (3.5") on a Laser 128• Task Force on a hard drive and Wings by Vitesse• The Basics of Cracking (part 5): Deprotection of Modified DOS disks• The Basics of Cracking Part 6: Mating Zone & Nibblizing Mysteries• Update on the Silent Service GS v925.01 crack• Xternal Commands for BASIC: CWD (Change Working Directory)• ONLINE• #80• The Product Monitor• *Features, Notes & such*: Add Copy II Plus file handling to your BASIC program• Comments on the Beginner's Book• Formatting 720K disks as 1.44M HD• How to SAVE hexdumps as CDA's• Logging ProDOS Drives• The Basics of Cracking (part 7)• The Basics of Cracking (part 8)• *Bitkeys*: Black Magic• Guild of Thieves• Gunslinger• King's Quest Series• Leisure Suit Larry• Man Hunter: New York• Police Quest• Realms of Darkness• Saracen• Sierra Boot Disks• Silicon Dreams• Space Quest Series• Ultima V• Wizardry Series• Xyphus• *Softkeys*: Ancient Art of War• Battle Chess• Bridge 6.0• Captain Blood GS• Dinosaur Days v1.0• Empire• Fahrenheit 451• Fay's Word Rally• GATO v1.3• Greeting Card Maker• Hostage• Keef The Thief• Magic Spells v2.0• Magic Spells v2.1• Mickey's Crossword

Puzzle Maker• Monsters and Make Believe v1.1• Pipe Dream• Pipe Dreams• Rear Guard• Rendezvous with Rama• Same or Different• Teacher's Tool Kit• Teacher's Tool Kit (IIC)• War of the Lance• Where in the USA is Carmen Sandiego?• WindwalkerGS• Windwalker II• *APTs*: Space Rogue• Wizardry III• *Playing Tips*: Countdown• Space Rogue• *IBM Softkeys*: Serve and Volley• Welltris

#81• The Product Monitor• *Bitkeys*: Micro Typewriter• *Softkeys*: Backyard Birds• Balance of Power• Chemistry: Balancing Equations• Chemistry: The Periodic Table• Chuck Yeager's AFT• EquationMath• Estimation: Quick Solve I• Estimation: Quick Solve II• Five-Star Forecast• Fossil Hunter• Grammar Toy

Shop• Instant Survey• Micro Typewriter v4.0• Murphy's Minerals• Patterns• Picture Chompers• Probability Lab• Professor AI's Sequencing Lab• Stickybear Shapes (ProDOS 1.5)• Studymate (the grade booster)• Sun and Seasons• The Duel: Test Drive II• Time Navigator• Tomahawk• Windwalker• *APTs*: Where in Europe is Carmen Sandiego?• Where in the USA is Carmen Sandiego?• Where in the World is Carmen Sandiego?• Where in Time is Carmen Sandiego?• *Playing Tips*: Windwalker• *IBM Softkeys*: Crime Wave• Gauntlet II• Stunt Driver• Thexder II• Wing Commander• *IBM Reader Review*: Copyright and much more...

For a complete back issue list, send a 75¢ stamp to Computist.

## Free Software

(while they last—only two copies left)

These software packages are NEW (shrink-wrapped except for the one copy of Sound Master that I opened in order to find out what it was). They're software packages that someone ordered and then canceled and we were unable to return.

### SoundQuest CZ Master

*Sound Quest In*  
(For Commodore Amiga)

Send \$2.00 for postage and packaging

For use with the Casio CZ-101, CZ-1000, CZ-3000, CZ-5000 and other compatible synthesizers. Included are file management and bank editing features, patch mixing and random voice generation features. Compose and mix your own music using many of the package options available.

Send order to Computist at the address listed on the Back issue order form below.

Issue	Mag	Disk	Issue	Mag	Disk	Issue	Mag	Disk	Issue	Mag	Disk
Core1	<input type="checkbox"/>	<input type="checkbox"/>	22	<input type="checkbox"/>	<input type="checkbox"/>	46	<input type="checkbox"/>	<input type="checkbox"/>	70	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	23	<input type="checkbox"/>	<input type="checkbox"/>	47	<input type="checkbox"/>	<input type="checkbox"/>	71	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	24	<input type="checkbox"/>	<input type="checkbox"/>	48	<input type="checkbox"/>	<input type="checkbox"/>	72	<input type="checkbox"/>	<input type="checkbox"/>
Core2	<input type="checkbox"/>	<input type="checkbox"/>	25	<input type="checkbox"/>	<input type="checkbox"/>	49	<input type="checkbox"/>	<input type="checkbox"/>	73	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	26	<input type="checkbox"/>	<input type="checkbox"/>	50	<input type="checkbox"/>	<input type="checkbox"/>	74	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	27	<input type="checkbox"/>	<input type="checkbox"/>	51	<input type="checkbox"/>	<input type="checkbox"/>	75	<input type="checkbox"/>	<input type="checkbox"/>
Core3	<input type="checkbox"/>	<input type="checkbox"/>	28	<input type="checkbox"/>	<input type="checkbox"/>	52	<input type="checkbox"/>	<input type="checkbox"/>	76	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	29	<input type="checkbox"/>	<input type="checkbox"/>	53	<input type="checkbox"/>	<input type="checkbox"/>	77	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	30	<input type="checkbox"/>	<input type="checkbox"/>	54	<input type="checkbox"/>	<input type="checkbox"/>	78	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	31	<input type="checkbox"/>	<input type="checkbox"/>	55	<input type="checkbox"/>	<input type="checkbox"/>	79	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	32	<input type="checkbox"/>	<input type="checkbox"/>	56	<input type="checkbox"/>	<input type="checkbox"/>	80	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	<input type="checkbox"/>	33	<input type="checkbox"/>	<input type="checkbox"/>	57	<input type="checkbox"/>	<input type="checkbox"/>	81	<input type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input type="checkbox"/>	34	<input type="checkbox"/>	<input type="checkbox"/>	58	<input type="checkbox"/>	<input type="checkbox"/>	82	<input type="checkbox"/>	<input type="checkbox"/>
11	<input type="checkbox"/>	<input type="checkbox"/>	35	<input type="checkbox"/>	<input type="checkbox"/>	59	<input type="checkbox"/>	<input type="checkbox"/>	83	<input type="checkbox"/>	<input type="checkbox"/>
12	<input type="checkbox"/>	<input type="checkbox"/>	36	<input type="checkbox"/>	<input type="checkbox"/>	60	<input type="checkbox"/>	<input type="checkbox"/>	84	<input type="checkbox"/>	<input type="checkbox"/>
13	<input type="checkbox"/>	<input type="checkbox"/>	37	<input type="checkbox"/>	<input type="checkbox"/>	61	<input type="checkbox"/>	<input type="checkbox"/>	85	<input type="checkbox"/>	<input type="checkbox"/>
14	<input type="checkbox"/>	<input type="checkbox"/>	38	<input type="checkbox"/>	<input type="checkbox"/>	62	<input type="checkbox"/>	<input type="checkbox"/>	86	<input type="checkbox"/>	<input type="checkbox"/>
15	<input type="checkbox"/>	<input type="checkbox"/>	39	<input type="checkbox"/>	<input type="checkbox"/>	63	<input type="checkbox"/>	<input type="checkbox"/>			
16	<input type="checkbox"/>	<input type="checkbox"/>	40	<input type="checkbox"/>	<input type="checkbox"/>	64	<input type="checkbox"/>	<input type="checkbox"/>			
17	<input type="checkbox"/>	<input type="checkbox"/>	41	<input type="checkbox"/>	<input type="checkbox"/>	65	<input type="checkbox"/>	<input type="checkbox"/>			
18	<input type="checkbox"/>	<input type="checkbox"/>	42	<input type="checkbox"/>	<input type="checkbox"/>	66	<input type="checkbox"/>	<input type="checkbox"/>			
19	<input type="checkbox"/>	<input type="checkbox"/>	43	<input type="checkbox"/>	<input type="checkbox"/>	67	<input type="checkbox"/>	<input type="checkbox"/>			
20	<input type="checkbox"/>	<input type="checkbox"/>	44	<input type="checkbox"/>	<input type="checkbox"/>	68	<input type="checkbox"/>	<input type="checkbox"/>			
21	<input type="checkbox"/>	<input type="checkbox"/>	45	<input type="checkbox"/>	<input type="checkbox"/>	69	<input type="checkbox"/>	<input type="checkbox"/>			

Some disks apply to more than one issue and are shown as taller boxes.

- ☆ Limited supply — first-come-first-serve basis.
- Out-of-print — only "Xeroxed" copies for sale.
- \* Issue 66 is laser printed on 8 1/2 by 11 paper.

## Back Issue Order Form

### Back Issue and Library Disk Rates

	Quantity	US, Canada & Mexico	All others
Back issues	5 or less	\$4.75	\$8.75
	6 to 9	\$3.75	\$6.00
	10 or more	\$3.00	\$5.00
Zox back issues*	any qty.	\$4.75	\$8.75
Library disks	5 or less	\$5.50	\$7.50
	6 to 9	\$4.00	\$6.00
	10 or more	\$3.00	\$5.00

Note: Total back issue and library disk orders to get quantity discounts. (ie. ordering 5 back issues and 5 library disks means that you pay the the quantity 10 price of \$3 each for both.)

\*Due to the time and effort involved in making Zox copies, their price will remain at \$4.75 each for US, Canada & Mexico and at \$8.75 for all other Foreign.

Shipping is included in all the prices shown.

### What's a library disk?

A library disk is a 5 1/4 inch floppy diskette that contains programs that would normally have to be typed in by the user. Documentation for each library disk can be found in the corresponding issue.

Library disks are available for all issues of COMPUTIST.

For a complete back issue list, send a 75¢ stamp to Computist.

Number of back issues. \$ \_\_\_\_\_

Number of Zox back issues. \$ \_\_\_\_\_

Number of library Disks. \$ \_\_\_\_\_

Washington state residents add 7.8% tax \$ \_\_\_\_\_

Total enclosed \$ \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Country \_\_\_\_\_ Phone \_\_\_\_\_

VISA \_\_\_\_\_ MC \_\_\_\_\_ Exp. \_\_\_\_\_

Signature \_\_\_\_\_

• US funds drawn on US bank. • Most orders shipped within 5 working days, however please allow up to 4 weeks delivery for some orders. • Large orders are shipped UPS so please use a street address. • Offer good while supply lasts. • Call (206) 832-3055 to use a credit card or send check/money order to:

COMPUTIST 33821 E Orville Road Eatonville WA 98328