"Mapping the Atari" Returns!

Exclusive serialization of Ian Chadwick's reference classic

Customer List Database

New Atari Game Cartridges

11 Reviews
American Techna-Vision

For Orders Only: 1-800-551-9995
CA, Orders / Information 416-352-3787
- No surcharge for VISA/MasterCard
- Your card is not charged until we ship

SPINNAKER
EDUCATIONAL CARTRIDGES
Cartridge work with all Atari 8-bit Computers
- Kindercrisp
- Delta Drawing
- Fraction Fever
- Story Machine
- Alphabet Zoo
- All in the Color Caves $8.75 ea.

1050 MECHANISM
Factory fresh TANDON mechcs, make difficult repairs a snap. Units are complete with head, Slepper, Spindele motor, belt etc. Just plug in, no difficult alignments or adjust
ments required. $47.50

POWER PACKS
External power supply for your 8-bit Atari
Power Pack 5VDC, 2002/2020 $25.00

HOT DISK TITLES
- Video Title Shop
- ChessMaster 2000
- Alternate Reality the City
- Alternate Reality Dungeon
- Tomahawk (64k)
- Bramark
- 221 B Baker St. $12.50 ea.

VISICALC SPREADSHEET
Unleash the computing power of your 8-bit Atari with Visicalc. Compute everything from home finances to high powered financial projections. Hundreds of uses. $19.95

THE BOOKKEEPER AND CX58 KEYPAD
You get both your 8-bit bookkeeping system and the handy CX58 numeric keypad for one low price. Packages factory sealed. $19.95

WORD PROCESSORS
Bark Street Writer Disk... $17.50
Cut & Paste Disk... $17.50
AtariWriter Cartridge... $29.95

800 4 PIECE BOARD SET
Includes Main Board, Power Supply Assembly, CPU Module and 10K Ristikon B Operating System Module. All boards are new, tested and complete with all
components. $29.50

DATA BASE
Data Perfect by LJK is a high-powered general purpose data base for use in dozens of applications. Create your own mailing lists, business data base, etc. $89.95

800/400 MODULES NEW PARTS COMPLETE WITH IC'S
- CX85 16K RAM MODULE... $5.95
- 580 MAIN (MOTHER) BOARD... $9.50
- 800/400 CPU Module with GTIA... $8.50
- 800 10K "B" O.S. MODULE... $9.95
- 400 MAIN (MOTHER) BOARD... $5.95
- 400 POWER SUPPLY BOARD... $8.95
- 400 POWER SUPPLY BOARD $14.50

INTEGRATED CIRCUITS
- CD170B... $4.50
- CD200A... $4.50
- U7 - 6532... $6.50
- CD375... $4.50
- CD396... $4.50
- U9 - 6527... $4.50
- CD907... $4.50
- U10 - ROM... $19.50
- CD1298... $4.50
- CD2298... $4.50
- U21 - RAM... $19.50
- CD909B... $4.50
- CD1299... $4.50
- U22 - 1050 IC's... $4.50
- CD1499B... $4.50
- 1050 IC's... $3.75 each
- Specify Socket#: $4.50

REPAIR MANUALS
SAM'S Service Manuals for all Atari 8-bit Computers, parts lists, labelled photographs showing the location of checkpoints and more! A special section gives oscilloscope and logic probe readings allowing you to narrow the malfunction down to a specific chip or transistor! $8.50

MISCELLANEOUS
- 17 INK ROLLER $5.50
- 17 Pin Serial I/O Cable... $5.95
- ACE JoyStick... $7.95
- 800XL RF Modulator... $9.50
- 1050 Track 0 Sensor... $8.50
- 2793 1050 Controller IC... $19.50
- U.S. Dublars $29.50
- SPARADOS Tool-Kit... $32.50
- Paddle Controllers (Par)... $6.50
- 400 3 Piece Board Set... $19.50
- Fastchip for 800/400... $15.50
- RAMbo XL w/o RAM IC's... $19.95
- 1050 Print Cable... $19.45
- Printer Interface... $19.95
- I/O 13 Pin PC Cable... $24.95
- ST Monitor Cable connector... $5.95
- ST Drive Cable plug end... $6.50
- 5 1/4 Drive Cable... $32.95

COMPUTER BOOKS
Atari Programmer's Bible... $7.95
Handy Tools & Tricks... $6.95
Inside Atari Basic... $5.00
Atari Basic Ref. manual $5.00
How to Atari 8652 Program $5.00
Write Your Own Games... $5.00
Programmer's Ref. Guide... $12.95
Assembly Language Guide... $21.95
XE Users Handbook... $21.95
XL Users Handbook... $21.95
Atari Basic Faster & Better... $22.95
Your Atari... $17.95

SERIAL I/O CABLE
High quality, 12 pin, $5.95
Special 3 cables for $12.50

MAC-65 CARTRIDGE
6502 Machine language Macro Assembler, first glass tool for serious programmers... $59.95

ATARI 8580 BASE
PC Board with parts and crystal $7.50
Board & plug in IC's... $39.95

PR CONNECTION
Parallel Interface for connecting moderns and printers... $65.00

BASE CARTRIDGE
Exact replacement for 800/4000/1200XL... $15.00

ANIMATION STATION
Graphics Design Table... $74.95
Includes Design Lab software.

AMERICAN TECHNA-VISION
Mail Order: 15338 Inverness St., San Leandro, Ca. 94579
Sales Office: 2098 Pike Ave., San Leandro, Ca. 94577
Terms: NO MINIMUM ORDER. We accept money orders, personal checks or C.O.D.s. VISA, MasterCard okay. Credit cards restricted to orders over $19.50. No personal checks on C.O.D. Shipping: $4.00 on shipping and handling, orders under $150.00. Add $2.75 for C.O.D. orders. In Canada total $5.00 for shipping and handling. Foreign shipping extra, Calif. residents include 7% sales tax. All products new and guaranteed. All sales final.

CALL TOLL FREE
1-800-551-9995
IN CALIF. OR OUTSIDE U.S. CALL 415-352-3787

Prices subject to change without notice. Send SASE for free price list. Atari is a trademark of Atari Corp.
FEATURES

AUGUST 1989, VOL. 8, NO. 4

12 VRENAME by Doug White
   BASIC variable renaming tool.  Type-in Software 39

15 HI-RES PT AND GET by Brad Timmins
   Animation without Player/Missiles.  Type-in Software 38

20 SUPERHOP ACTION! by Douglas Skrecky
   Faster fractals grow in your Atari.  Type-in Software 37

22 MAPPING THE ATARI by Ian Chadwick
   Exclusive! Antic brings back the classic 8-bit reference book.

28 CARNIVAL OF CARTRIDGE reviewed by David Plotkin
   Food Fight, Karateka and lots more hits.

DEPARTMENTS

SUPER DISK BONUS

7 ATARI CHING by Patrick Harvey
   The Computer of Changes foretells your fate.

FEATURE APPLICATION

8 CUSTOMER LIST MANAGER by Thomas Andrews
   Much more than a phonebook.  Type-in Software 34

GAME OF THE MONTH

17 GOOD KING ZURP by Frank Martone
   The great pasta sauce chase... Don't ask!  Type-in Software 40

SOFTWARE LIBRARY

5 Easy-To-Type 8-Bit Listings

33 TYPO II, SPECIAL ATARI CHARACTERS

<table>
<thead>
<tr>
<th>5</th>
<th>I/O BOARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>TECH TIPS</td>
</tr>
<tr>
<td>44</td>
<td>CLASSIFIED ADS</td>
</tr>
<tr>
<td>45</td>
<td>ADVERTISERS LIST</td>
</tr>
</tbody>
</table>

Antic—The Atari Resource (ISSN 0745-2527) is published monthly by Antic Publishing. Editorial offices are located at 544 Second Street, San Francisco, CA 94107. Second Class Postage paid at San Francisco, California and additional mailing offices. POSTMASTER: Send address changes to Antic, The Atari Resource, P.O. Box 1569, Martinez, CA 94553. Subscriptions: One year (12 issues) $28. Canada and Mexico add $8, other foreign add $12. Disk Edition (12 issues with disks) $59.95, all foreign add $25. (California residents add 6.5% sales tax for disk subscriptions. Editorial submissions should include text and program listings on disk and paper. Submissions will be returned only if stamped, self-addressed mailer is supplied. Antic assumes no responsibility for unsolicited editorial material. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Antic is a registered trademark of Antic Publishing, Inc. An Information Technology Company Copyright ©1989 by Antic Publishing. All Rights Reserved. Printed in USA.)
Get Clicking With Your Atari ST!

Subscribe to START magazine with disk

PROGRAMMING
UTILITIES
GRAPHICS
GAMES

Tutorials on START Disk brings the expertise of top ST programmers right into your computer.

Manage the power of your ST, and make your computing more enjoyable with utilities on START disks.

Outstanding animation and graphics from the best ST artists around, from START Disk right on to your screen.

Stimulate your imagination and your reflexes with games on START Disk! How-to tutorials show you how to write your own pulse-pounding games.

Subscribe Today, call 1-800-234-7001

Use your Visa or MasterCard
BERSERK

I'm trying to use BSRk from the August 1988 Antic Disk, and it seems to have wrong and missing commands. The first problem was that the program would load only partially into the BSR Controller, then stop and lock up. Then the asterisks were not showing the correct days of the week in use. I seem to have corrected this problem by changing the (X - 1) to (X - 1) - .1 in line 6610.

Also, the program isn't sending the code to turn the appliance modules on and off, although the program seems to turn the lamps on and off all right.

Ed Sacks
Garden Grove, CA

As it says in the BSRk article, don't run the program from the monthly disk. Copy it to another disk, then follow the rest of the instructions on page 21 of that issue. Also, this program was tested with standard Atari BASIC, so don't use Turbo BASIC or BASIC XL/XE.—ANTIC ED

ANTIC REFRESHERS

I want to activate the unused 4K block of memory on my Atari 800 (for my own programs, only, of course), but I don't know if the ANTIC chip refreshes over 48K. Can you help me?

Ralph Munoz
Los Angeles, CA

There is no memory in that 4K block you can use—unless you have a memory upgrade that uses that area, in which case it will be refreshed.—ANTIC ED

OVER-PROTECTED INDUS

I have an Indus GT disk drive for my 130XE. The drive seems to be stuck in protect mode. Unless I can find a way to bypass or disable the protect mode, I am unable to save anything to disk. A letter to Indus Systems was returned unopened. Can you help me?

Vinton Henderson
APO NY

Future Systems Inc. does repair work on Indus drives. Their address is 21634 Lassen Street, Chatsworth, CA 91311. (818) 407-1647.—ANTIC ED

CHALKBOARD POWERPAD

In your October I/O board you had a reader who wanted to know where he could get the ChalkBoard PowerPad. We have several dozen units in stock priced at $30 each. We also have Music Maestro and Bear Jam PowerPad cartridge/overlay software available at $15 each. (The PowerPad is useless alone and requires a plastic overlay plus software.)

If any of your I/O Board readers could find the Micro Illustrator cartridge and overlay (ChalkBoard PowerPad version only!) they would have an excellent drawing pad due to the large drawing surface of the PowerPad.

Gail Maddox
American Techna-Vision
15338 Inverness Street
San Leandro, CA 94579
(415) 352-3787

Antic welcomes your feedback, but we regret that the large volume of mail makes it impossible for the Editors to reply to everyone. Although we do respond to as much reader correspondence as time permits, our highest priority must be to publish I/O answers to questions that are meaningful to a substantial number of readers.

Send letters to: Antic I/O Board, 544 Second Street, San Francisco, CA 94107.
NEW ARRIVALS

NEW CARTRIDGES!

1020 COLOR PRINTER PLOTTER
Complete with:
- 2 Pen Sets
- 1 Roll Paper
- Power Supply & Cable
EXTRA PEN SETS

$14.98

1200XL 64K COMPUTER
w/PAC-MAN

$49.

1027 PRINTER
- 80 Column
- Direct Connect
- Letter Quality

$79.

1050 $169.

ATTN. DEALERS
PAC-MAN
Case of 120

800 48K COMPUTER
w/PAC-MAN

$69.95

800 PRINTERS
- 80 Column
- Dot Matrix
- Electromagnetic
- Direct Connect

$79.98

CARTRIDGES FOR 800, XL, XE

BASIC CARTRIDGE
$4.95

BASIC TUTOR (4 BOOKS)
$4.95

PAC-MAN (no box)
$4.95

DONKEY KONG (no box)
$4.95

DEMON ATTACK (400,800)
$4.95

DELFX INVASION
$4.95

JOURNEY TO THE PLANET
$4.95

STAR RAIDERS
$4.95

MISSILE COMMAND
$4.95

ASTEROIDS
$4.95

GALAXIAN
$4.95

E.T.
$4.95

FACEMAKER
$4.95

MATH ENCOUNTER
$7.98

DANCE FANTASY
$8.98

LOGIC LEVELS
$8.98

MEMORY MANOR
$8.98

LINGLONG LOGIC
$8.98

CHICKEN
$8.98

CLAM JUMPER
$8.98

DELTAV DRAWING
$8.98

HEY DIDDLE DIDDLE
$8.98

SLIME (400/800)
$8.98

ALPHABET ZOO
$8.98

ALF
$8.98

ADVENTURE CREATOR
$8.98

DIG DUG
$14.95

SKY WRITER
$14.95

FOOTBALL
$14.95

DEFENDER
$14.95

TENNIS
$19.98

FINAL LEGACY
$19.98

MARIO BROSS
$19.98

DONKEY KONG JR.
$19.98

JUNGLE HUNT
$19.98

MOON PATROL
$19.98

BATTLEZONE
$19.98

FOOD FIGHT
$19.98

HARDBALL
$19.98

FIGHT NIGHT
$19.98

ONE ON ONE BASKETBALL
$19.98

DEER FALCON
$19.98

NECROMANCER
$19.98

RESCUE ON FRAGTUS
$19.98

BALLBLAZER
$19.98

BLUE MAX
$19.98

STAR RAIDERS II
$19.98

DAVID'S MIDNIGHT MAGIC
$19.98

ARCHON
$19.98

KARATEKA
$19.98

CHOPPER
$24.98

CAYO
$24.98

ACE OF ACES
$24.98

LODE RUNNER
$24.98

AIRPLANE
$29.98

SUMMER GAMES
$29.98

CROSSBOW
$29.98

CRIME DUSTERS
$29.98

DISK SOFTWARE FOR 800, XL, XE

DAVID'S MIDNIGHT MAGIC
$4.98

ZORRO
$4.98

BANDITS (48K/400,800)
$4.98

PROTECTOR II
$4.98

CLAIM JUMPER
$4.98

SYNTEX
$4.98

CROSSCHECK
$4.98

MOLECULE MAN
$4.98

CRYSTAL RAIDER
$4.98

DEEP STRIKE RIDER
$4.98

MISSION ASTEROID
$4.98

SPIDERMAN
$4.98

HULK
$4.98

COMBAT
$4.98

PREPPEE I
$4.98

PREPPEE II
$4.98

THE COUL
$4.98

DISK 50 500 GAMES
$4.98

FREAKY FACTORY
$4.98

LASER HAWK
$4.98

ROCKET REPAIRMAN
$4.98

ADVENTURELAND
$4.98

PIRATE ADVENTURE
$4.98

SECRET MISSION
$4.98

VOODOO CASTLE
$4.98

TECHNICOLOR DREAM
$4.98

STRANGE ODYSSEY
$4.98

VISCALC
$4.98

BOOKKEEPER
$29.98

GET RICH
$29.98

Light Gun

For use with pistol games on 8-Bit systems

$34.98

Light Gun Package

Includes:
- Light Gun
- Laser Blaster
- Crossbow
- Barnyard Blaster

$109.
Atar-I Ching

The Computer of Changes foretells your fate. By Patrick Harvey

Let your faithful Atari "tell your fortune" and answer your questions about the future with this month's Super Disk Bonus, Atar-I Ching.

The I Ching is a most ancient Chinese fortune-telling system. In China its users have ranged from carnival fakes to generals and emperors. Its advice was used in everything from deciding what crop to plant to deciding military strategy.

Like many other traditional fortune-telling devices such as the Tarot cards, the I Ching is based on using "organized randomness" to try tapping into the underlying trends in the universe. Like most such devices, the key is in how you interpret the results, rather than the actual results themselves.

In one form of the I Ching, the user shakes numbered sticks in a container until one falls to the ground. Atar-I Ching simulates today's most widely used method—tossing three coins to create a set of lines. Depending on the number of coins that land with heads or tails up (ancient Chinese coins were inscribed on only one side and blank on the other) the line will be Yin, Yang, moving Yin or moving Yang. This is done six times to produce a "hexagram."

Yang lines are solid and Yin lines are broken. If there are any moving lines (all three coins are the same) a second hexagram will be drawn by changing the moving lines.

ASKING QUESTIONS

Atar-I Ching will wait for you to mentally form your question and achieve the properly blank, receptive state of mind. The BASIC file is on this month's Antic Disk as ICHING.BAS and can be started by selecting it from the main menu. Then "toss the coins" by pressing the [SPACEBAR] three times.

Using the Atari's random number generator, the coins are tossed to determine a hexagram. Like binary code, six lines of either yin or yang provide 64 distinct hexagrams.

Based on the text associated with each of the hexagrams and component trigrams, plus your own intuitive understanding of the question, you can develop a forecast of future events. As the I Ching texts tend to be very vague, this can require a lot of imagination.

Atar-I Ching is a colorful, atmospheric program written by Patrick Harvey of Mountain View, California. This is his first Antic publication and he's a computer engineer doing customer training for National Semiconductor's 32-bit microprocessors.

Admittedly, Atar-I Ching is not quite as long as most of the other Super Disk Bonus programs that appear in Antic. However, the listing does include many lines crammed with interpretive text, which would be particularly time-consuming to type.

Your August 1989 Antic Disk—featuring Atar-I Ching as well as every type-in program from this issue—will be shipped to you within 24 hours after your order is received. Just phone Toll-Free to the Antic Disk Desk at (800) 234-7001. The monthly disk is only $5.95 (plus $2 for shipping and handling) on your Visa or MasterCard. Or mail a $5.95 check (plus $2 shipping and handling) to Antic Disk Desk, 544 Second Street, San Francisco, CA 94107.
Customer List Manager

Much more than a phonebook

By Thomas Andrews

Keep your important lists up-to-date with this powerful but user-friendly BASIC telephone/mailing list database. Works on 8-bit Atari computers with at least 48K memory, disk drive and a printer.

My family operates a retail vegetable business on our farm in central New York. We have a series of lists of customers who make quantity purchases each year and wish to be informed when certain seasonal products become available.

When we call these customers we want to have some information at hand concerning each one's preferences and a history of recent orders to aid them in making this year's purchases. Customer List Manager lets us find this information with ease. It's much better than the method we used before our Atari 800 arrived—a spiral notebook.

Customer List Manager is a telephone/mailing list/database program. Each record contains a customer's name, phone number and address, along with the type of product they want, some notes on personal preferences and a five year history of past orders. Each list has a title and year attached to identify it.

The list can be displayed on the screen, printed on most 80-column printers, or output to a word processor file. Entries can be edited or deleted, and histories updated, in one easy operation, thanks to the program's sorting and selecting routines.

The records in each list (referred to as "items" within the program) can be sorted on one or two levels. Subgroups of list records can be selected for processing under a search routine which allows use of a wildcard character in the search sequence. In case of disaster, the list can be reconstructed from the last editing operation or from the word processor file. All this—and it can be customized to better suit your own needs as well.

GETTING STARTED

Type in Listing 1, MANAGER.BAS, and check it with TYPO II. Be sure to SAVE a copy before you RUN it. If you have trouble typing the special Atari characters in lines 50, 60, 65, 85, and 86, don't type them. Listing 2 will create these lines for you. Type in Listing 2, check it with TYPO II, and SAVE a copy.

When you RUN Listing 2 it will create a file containing these hard-to-type lines called LINES.LST. Merge this file into Listing 1 by typing LOAD "D:MANAGER.BAS" and then ENTER "D:LINES.LST". Be sure to SAVE the final version of the file.

If you have the Antic Monthly Disk, you should copy MANAGER.BAS to another disk before you try to use it,
since the program needs to be able to write files to disk. You can either copy the file from DOS using command C or O, or LOAD and SAVE the file from BASIC. Make sure the disk to which you copy MANAGER.BAS also contains DOS.

The program is menu driven and most operations are self-explanatory. When you first RUN the program the main menu will appear. If no list has been loaded, only options A (Start New List), C (Load List From Disk), and H (Change List Title) will operate.

Note that the [BREAK] key is NOT disabled. To return to the main menu after an accidental BREAK, type GOTO 100. Records in memory should remain intact!

CUSTOMIZING LISTS

Customer List Manager uses a variable length field and record system to lower wasted memory space. Provisions have been made to limit the total list size to 19,000 characters and/or 100 records. This size allows the program and storage area to fit in a 48K Atari with standard DOS 2.5 configuration with about 1K bytes left over.

To change the maximum list length, change the value assigned to MAX in line 40. To change the maximum number of records, change the value of HUN in the same line. Remember that each record allowed uses seven bytes of bookkeeping overhead.

Each record has ten fields. The first five are labeled NAME, PHONE, ADDR., TYPE, and NOTES. These labels can be changed in line 90 by changing FNAMES$. Be sure to use no more than five characters, padded with spaces if shorter, and separate each label with a colon.

The last five fields contain the history of the last five years and the labels for these fields are determined by the year of the list. The maximum length of any field is 60 and the minimum is 1. Anything outside these limits will be either cut off or padded with a space.

Both upper and lower case characters are allowed in the list. Inverse characters are not allowed and will be changed to normal if they are used.

Menu selections D, E, and G (editing, saving, and printing) allow the operation to be performed on only part of the items on the list. Items are flagged for processing through the use of a search routine that uses the question mark (?) as a wildcard character.

As an example, a search of phone numbers using the sequence 575 will flag all items with two fives separated by any other character in the phone number field. Use all capital letters in the search sequence, and Customer List Manager will search for both upper and lower case.

To change the wildcard character, change the question mark in the machine language routine SRCH in line 60 to any character you wish. Keep in mind that the character used as a wildcard cannot be used as a regular character. DON'T change anything else in line 60 as this could cause your computer to lock up. Also change the screen prompt in line 15060 as a reminder that a different wildcard is in effect.

EDITING

The editing option is quite simple. There are three editing modes—edit and update, edit only, and update only. Update means that the list year is incremented by 1, the oldest year of history is dropped, the remaining history is shifted, and a new year is added to the end of each record.

The update-only mode is completely automatic and all list items will be updated. A space is inserted for the new history year field as the new list is written to a special edit file on the disk. After the entire list has been updated, the old list will be cleared from memory and the new list is input from the edit file.

The Edit and Update mode allows you to enter information for the newly-added year. You can also change the contents of any of the fields, or you can delete a whole item (or record) from the list. You must go through all of the items on the list in this mode. To edit, move the cursor to the desired line and change it to read as you like.

Because of the way the List Manager reads the list item from the screen, the INSERT and DELETE functions are disabled during editing, as is the clear screen command. When you have finished editing an item press [CONTROL] [A]. The item is read from the screen and placed in the edit file while Customer List Manager goes on to the next item.

To delete an item press [CONTROL] [D]. The item will not be sent to the edit file. (A last-chance "Are you sure?" question is asked before deletion is done.) If an item gets so messed up during editing that it would be easier to start over, press [ESC].

The edit-only mode allows the selection of part of the entries for processing. A menu comes up, letting you choose how much of your file you want to edit. If you decide to just edit some of the items (option B), you will be asked which field you want to use for the selection. Enter one, and you will then be prompted for the search sequence to use for a match. Remember, the ? character is the wildcard that matches any other character.

If you want to leave edit-only mode without editing all the items flagged for processing, press [CONTROL] [M]. The rest of the flagged items will be moved to the edit file on disk, with no changes. The program will load the new list from the disk, ready for any further operations.

In all modes, once the editing operation is done, the edit file is left on the disk as a backup file that can be recovered using main menu option 1. Each time a list is edited the old edit file is replaced with a new one.

List sorting can be done on any two fields. The primary field is selected first, followed by the secondary. To sort on only one field, use the same one for both primary and secondary.
The sort routine is completely automatic once started.

PRINTING

The print option assumes a generic, no frills, 80-column printer using continuous paper. Therefore, it ought to work on just about any printer you would connect to your Atari. It is not necessary to print the entire list if you only need part of the pages. Customer List Manager will print any individual page or group of pages, and as with editing, you may select to print only those records which contain a specified string in the chosen field.

If your printer has capabilities beyond the minimum supported here (and most do) you might wish to write a new printer routine to take advantage of them. But if you don’t want to do this, the program will print a list to a disk file that most word processors should be able to use. You can then use your word processor to add any special touches you want before you print out.

Customer List Manager has the ability to reconstruct a list from the word processor file (without the formatting commands or any other changes), providing a second emergency backup system. Use menu option I.

You can’t merge two lists with Customer List Manager, but you can merge files using your word processor. Load the two list files into the word processor one after the other. (Do not mix Customer List Manager word processor files with files saved with menu option E. They use a different format.)

Remove the list title and year from the second list and save the completed list on your disk. Load the combined list into Customer List Manager using menu option I. If the combined list is longer than the maximum allowed, the program will load as many records as will fit, then stop.

It has been my experience that most practical application programs cannot precisely fit the needs of all users, and databases are no exception.

That is why there are so many database programs out there—each user has different requirements.

Beyond changing field names and memory limits as discussed above, further changes would require more extensive rewriting of the program code. I have tried to program in a fairly modular, structured form to make this easier to do. If Customer List Manager does not suit your needs, study the listing and the instructions below, so you can adapt the database the way you want it.

ABOUT THE PROGRAM


As presented, FIND required that the strings be stored sequentially, unpadded with spaces, with the first character of each changed to inverse as a delimiter. The routine also required that an inverse character be appended to the end of the last field. Each string could be no longer than 255 characters.

FIND started at the beginning of a string and counted inverse characters until the desired field was encountered. FIND then determined the string position of the start and the length of the field.

This sequential access would be fine for most text adventures since FIND is quite fast. However, in the case of a search or sort operation where 100 or more calls to this routine are possible, or even likely, processing time mounts up quickly.

I added an index to the main storage area to provide pseudo-random access. The string position of the first character of each record is placed in a numerical array called LI. Now, if I want to examine the fifth field of record number 42, I use the contents of LI(42) as a starting point and go five fields from there. This provides very fast access—nearly as fast as if I had used fixed-length fields and calculated the position—and is much less wasteful of RAM.

Since the FIND routine uses inverse characters as delimiters, it follows that inverse characters cannot be allowed in the middle of a field. This necessitates a routine that does a character by character check of the input field, changing the first character to inverse and the rest of the characters to normal.

Since a BASIC routine to do this would be quite slow, I wrote the machine language routine CHNGR to do it more quickly. The call for CHNGR is:

\[ \text{DMY} = \text{USR(CHNGR,ADR(FIELD$),LEN(FIELD$))} \]

CHNGR is the address of the routine in memory that operates on the string FIELD$.

The USR routine SRCH will examine any block of memory for an occurrence of a specified sequence of characters. The sequence may contain any combination of letters, numbers, and punctuation. Capital letters (normal video) are checked as both upper and lower case. All other characters are examined only in the case and video type actually specified. The call for SRCH is:

\[ X = \text{USR(SRCH,ADR(SEQ$),LEN(SEQ$),ADR(AREA$),LEN(AREA$))} \]

SRCH is the address of the routine that attempts to find SEQ$ in string AREA$. A one is returned in X if the sequence is found anywhere in the search area, and a zero if the search was unsuccessful. The length of the sequence may vary from 1 to 255 characters. The search area can be any length from 1K to 64K.

SORT ROUTINE

The sort routine uses an insertion sort on two levels. For those unfamiliar with the insertion sort, consider a file drawer that needs sorting. You start on either end and thumb through the folders one at a time un-
til you find one that is out of the proper sequence. You then check back through the folders you have just done until you find the proper position for the misplaced one.

Then, you shift the folders to make room and insert the one in question in the right spot and go back to where you left off. When you reach the end, the sort is done. This type of sort, while certainly not the fastest type available, has the advantage of being very fast if most of the items are already in order, as is usually the case with Customer List Manager. It is also nearly as easy to program in BASIC as the bubble sort and is usually faster.

It is important to note that I do not actually move the text of the list around during the sort—I change the order of the pointers to that text. The reason for this is simple: When I move a pointer, I only have to move six bytes. Since the size of each record (item) in a list can vary anywhere from 10 to 600 bytes, not only would I have to move more bytes, but I would have to determine the record I have to move to each move at all. It is easy to see why this is faster.

The usual method used during an insertion sort with a numerical array is to shift the array elements one at a time as you are checking for proper position. In this sort, I wait until I have found the proper position andshift the elements in a block using a machine language block memory mover, SHIFT.

This routine is a little different than the usual block mover in that it is bilateral. Most routines, including the string functions of Atari BASIC, will not work for moving in both directions when the source and destination blocks overlap. SHIFT checks for the direction of the move and starts at the correct end of the block to properly shift it in that direction. The call for SHIFT is as follows:

DMY = USR(SMTH, HERE, THERE, LENGTH)

SHIFT is the address of the routine, HERE is the address of the first byte of the source, THERE is the address of the first byte of the destination and LENGTH is the number of bytes to move.

To determine the address of the numerical array, I DIMensioned a string just before it, determined the address of that string with the ADR function, and added the DIMensioned length of the string to it. Each element of the array takes six bytes. (Remember that the first element of the array has a subscript of 0.) Since I do not use LI(O) in CLM, I added six more bytes to the base address for LI.

Thomas Andrews has a degree in electrical engineering. He found his Atari 800, some software and 12 issues of Antic at a garage sale in 1985.

Listing on page 34

NEW HACK BACK Special OFFER

NEW PARROT II

An All New Parrot sound digital for your Atari. Parrot II is a revolutionized new hardware design that plugs into your joystick port. Parrot II has two inputs, one for a microphone and one for a powered source such as a talk phone, radio or Compact Disk. Parrot II will let you record sounds into your computer and play them back on any Atari. Parrot II lets you modify the sounds on a graphic display to create brand new sounds and special effects. Read the manual and you will be on your way to producing your very own music. Only $59.95

NEW POP-N-ROCKER

Simply hook up this unit to any analog sound source (e.g. CD player, tape deck, etc.) and it becomes an instant music machine. Works great with sampled audio files or for live performances. Only $24.95

COMPUTERYES & MAGNIPRINT II+

Save money, print faster, do more.

COMPUTERYES camera system Comes complete with everything above, plus a black and white video camera and connecting cable. $239.95

Graphics 8 Software - Add a new dimension to your COMPUTERYES pictures - captures in 16 shades of grey $12.00

Magniprint II+ Easily the most powerful print program available today. Print graphics from almost any format in hundreds of styles and sizes and shades. Supports color printer and lets you create giant posters. Magniprint II+ lets you stretch and squeeze, invert, add text, adjust shading and much more. Works with EPSON, NEC, HCL, Compaq, Panasonic, Gemini, IBM XM360, and compatible printers. (950 interface or equivalent required).

$24.95

Graphics Transformer Now you can combine the most powerful features of all your graphics programs. Create print shop logos from Arial typeface pictures, from a photo to a computerized ComputerYES, or from any picture. Graphics Transformer lets you Shrink, Enlarge and Merge pictures for unparalleled flexibility. $22.95

YOUR ATARI COMES ALIVE

SAVE MONEY. Finally an alternative to buying expensive computer add-ons. Your Atari Comes Alive shows you how to build them yourself. Highly detailed text and diagrams gives you complete step by step instructions and programs needed to build and control these exciting devices and MORE:

Lighting & Motor Controllers

Robot Controllers

Voice Recognition & Environmental Sensors

Data Decoder

More than 150 pages. Your Atari Comes Alive $24.95
VRename

BASIC variable renaming tool.

By Doug White

Clear up old BASIC programs and make your own programs more readable by renaming the variables with this handy BASIC utility. This BASIC program works on all 8-bit Atari computers of any memory size, with disk drive.

Have you ever gotten a BASIC program out of an old magazine or from a friend that was difficult to read because it had cryptic names for all of the variables? While Atari BASIC allows variable names up to 120 characters long, some of the more primitive BASIC interpreters on some computers allow a maximum of two characters for each variable name. If you didn't write the program how can you tell what names like R9, D8, L2 and X$ mean?

Or have you used a subroutine from an old program in a new BASIC program that you were writing? There'd be a good chance that some of the variables were not compatible between that subroutine and the rest of the program. You had two choices—either change every occurrence of each problem variable name throughout the entire program, or else add program lines to assign needed data from the variable names that the program uses to the corresponding variable names that the subroutine uses.

The first solution is time-consuming. The second creates confusing “spaghetti code;” since the same piece of data will have a different name depending upon where it is in the program.

To help you (and me) write understandable programs and decipher programs already owned, I have constructed a programming utility that will let you rename any or all of the variables in a BASIC program.

This programming tool is simple in theory but powerful in practice. Vrename can be used on completed programs stored on disk. But it is primarily designed to let you modify the variable names of a program that you are currently writing. For this reason the utility does not change the margins, colors, or any other feature of the display that you may have customized for your own convenience.

GETTING STARTED

Type in Listing 1, VRENAME.LST, check it with TYPO II. This utility will not work with SAVE and LOAD—so LIST a copy of it to disk before you use it the first time. The utility deletes itself from memory after it is finished, so once you RUN it
you'll have to retype the whole program if you don't have a copy safe on disk.

Rather than making up a useless test program for demonstration purposes, I've included a program that contains a Shellsort routine in lines 500-680 for sorting numbers stored in an array named A() into descending order. Type in Listing 2, SHELLSORT.BAS, check it with TYPO II, and SAVE a copy to disk.

The Shellsort is much faster than the commonly used bubble sort. If you want to use it in your programs, just rename A() to whatever name your program uses. If you need the numbers sorted in ascending order, change the < character in line 620 to a > character.

USING VRENAME.LST

Now you can RUN the Shellsort program to try it out. Then, with the sorting program still in memory, ENTER "D:VRENAME.LST" and RUN it.

The first variable used in the sort program, A(), will be shown on the screen. To rename it, type [Y] and press [RETURN]. Now type in the new name and press [RETURN]. Since the variable A() is an array, your new array name must also end with a ( character. Type [Y] and press [RETURN] if the name is spelled correctly.

The next variable name will now be shown on the screen. If you don't want to rename it, just press [RETURN] and the following variable name will appear. This process will continue until the last variable name of the sort program has been displayed.

Now the program will prompt you for a name for the disk file in which you want to keep the modified program. For example, type in D:SHELL2, or if you don't need a permanent copy you could use D8:SHELL2 to store it to a RAMdisk.

After the utility finishes, it will delete itself from memory. Press [RETURN] one more time to ENTER the modified program back into memory. Now if you LIST the program to the screen it contains the renamed variables. RUN the program and it will execute just as it did before.

CHANGING YOUR OWN

V Rename will work with almost any BASIC program, but your program must have line numbers greater than 0 and less than 31500. LOAD your program, then ENTER "D:VRENAME.LST" from disk. Then RUN the program, and rename variables as described above.

Be careful that you don't use reserved BASIC words (like LOAD, RUN, NEXT) as variable names. If you have a lot of variables, be sure you don't give different variables the same name by mistake. As always when modifying a program, it's a good idea to keep a backup copy on disk.

HELPFUL HINTS

While you are typing in a program, you can use variable names that are short and easy to type. After you finish you can substitute longer, more descriptive names. Simpler variable names mean less typing, and will not only save you time but will probably reduce the number of typing errors that you make.

When your program is running properly, you can use my variable renaming utility to add descriptive names that use these characters without causing any problems.

BASIC TOKENS

When you type in a program in Atari BASIC, the BASIC interpreter translates each program line into a line number and a series of string constants, floating-point constants, and tokens. Tokens are numbers that stand for each of the variables and BASIC commands in the program. The tokens are stored in a table called the statement table, which contains a "shorthand" version of the program. The tokenized version of the program in the statement table takes up less space and executes much faster than the program would if it were still in the original ASCII form.

If the value of a token is less than 128, it represents one of the BASIC commands or BASIC operators. If a token value is greater than 128 it stands for one of the variables in the program. For example, if the program contains a SETCOLOR command, the statement table will contain a 48 which is the token value for this command. In similar fashion a variable name will be replaced by a one-byte token each time it appears in the statement table.

The characteristics of each of the variables are stored in three other system tables—the variable name table (VNT) which contains the variable names that correspond to each of the variable tokens; the variable value table (VVT) which indicates whether a variable is a string, an array, or a simple variable called a scalar; or the string and array table (START) which holds the contents of every string variable and the value of each element in every array variable in the program.

In the variable renaming utility we are primarily concerned with the VNT. After a program is stored in memory, BASIC works with the tokens, the VVT and the START and ignores the variable names for the most part. Only when you are manually entering a program, LISTing it, or ENTERing it from disk or cassette are you directly working with the individual variable names. The SAVE and LOAD commands access the system tables only as a continuous block of data, and are of no use in renaming the variables.

All of the BASIC system tables that I've mentioned change size and move around in memory as different BASIC processes occur. The BASIC interpreter keeps track of the location of each of the tables using pointers.

Pointers are memory locations that contain the addresses of tables and other important data that move
around in memory. While the address stored in a pointer may change, the pointer itself stays in the same place. The variable renaming utility uses two pointers called VNTP, which points to the beginning of the VNT, and VNTD, which points to the byte immediately following the end of the VNT.

HOW IT WORKS

When you ENTER the utility from disk, it merges itself around your program. Your program must therefore have line numbers greater than 0 and less than 31500. The utility jumps from line 0 directly to line 31500 and does not execute the original program. The utility then reads each variable name out of the VNT and asks if you want to change that name. The old variable name or the new variable name (if you just changed it) is then stored in VARNAMES$.

The utility goes through all of the names in the VNT until it reaches VARNAMES$, the first variable of the utility itself. (If no names were changed, the utility terminates and both the program and the utility remain in memory.)

Now there is a complete updated list of all of the variable names stored in VARNAMES$. The utility puts the address of VARNAMES$ into VNTP to make BASIC look for variable names in VARNAMES$ instead of looking in the old table.

After you type in the name of the disk file where you want to store the modified version of your program, the utility LISTs lines 1-31499 (the original program) out to that disk file. As the program is being converted from tokens back into ASCII, the address in VNTP tells BASIC to read the updated names stored in VARNAMES$.

Finally the utility clears memory with a NEW command, which terminates the utility. Press the [RETURN] key and the modified program will be reentered into memory from the ASCII file now on disk.

An added benefit of LISTing and ENTERing your program to and from disk is that it cleans up your program. Old unused variable names will be discarded. If your 800XL has Revision B BASIC ROM, 16 bytes of garbage are added to the end of your program each time you SAVE it to disk. Periodically LISTing the program to disk and ENTERing it removes this garbage. You have Revision B ROM if you PEEK$(43234) and the result is 96.

PROGRAM TAKE-APART

31514 VNT is assigned the address of the variable name table. VNT is incremented to keep track of the current character being read from the table.

31518-31530 Read each variable name.

31531-31598 Change a variable name and put the new name in VARNAMES$. If unchanged the old name is put in VARNAMES$.

31606-31614 Store the old variable name table pointers.

31618-31648 Make the variable name table pointers point to VARNAMES$.

31652-31688 LIST the modified program out to a disk file.

31690 NEW clears out user memory and resets the system table pointers.

31694-31706 TRAP routine. If an error occurs while LISTing the modified program to disk, the old VNT pointers are restored.

SPECIAL CASES

As I mentioned earlier, this utility is designed to merge itself around a program that is currently in memory. With a dimensioned size of 2000 characters, VARNAMES$ will hold 128 names that average 15 characters in length, or 64 names that average 31 characters, or 32 names that average 63 characters, etc. If you have a particularly large program, both the program and the utility may not fit in available memory at the same time. If this occurs feel free to decrease the dimension of VARNAMES$.

Another problem might arise with programs that use a lot of variable names. Atari BASIC limits you to 128 variables in one program. The renaming utility itself uses three string variables and 13 scalar variables.

$128 - (3 + 13) = 112 VARIABLE NAMES$

If your program uses more than 112 variables, ENTERing the utility may cause an error since the total number of variables exceeds 128.

First, try to localize the section of the program that contains the variables that you wish to rename. Rename the variables in that section. Then recombine the program section with the rest of the program.

If this process does not solve the problem, you will have to create a special version of the renaming utility which uses the same variable names as your program. Use the following procedure:

1. LIST your program to disk and type NEW to clear memory.
2. ENTER the utility into memory.
3. Delete line 31534.
4. Modify line 31680 to read: 31680 LIST NAME$
5. Type RUN to run the utility on itself.
6. As each variable name is displayed, rename it to one of the variable names in your program.
7. When the utility is finished, ENTER the modified copy of the utility into memory.
8. Modify line 31680 to read: 31680 LIST NAME$,1,31499
9. LIST the modified utility back out to a disk file.

When you ENTER this specially tailored copy of the utility into memory with your program, BASIC will not have to add any new variables, since they already exist in your program.

Doug White of Arlington, Texas uses his 1200XL as an aid in designing and testing loudspeakers. His article Equivalence appeared in the February 1989 Antic.

Listing on page 39

ANTIC, THE ATARI RESOURCE
Hi-Res PUT and GET

Animation without Player/Missiles.

By Brad Timmins

Here's a different way to create colorful objects that move smoothly around your computer screen. This BASIC demo program includes the handy PUT-GET machine language routine for use in your own programs. It works on all 8-bit Atari computers of any memory size, with disk or cassette.

Creating colorful objects that move smoothly around your computer screen can be one of the most satisfying and eye-popping effects your Atari has to offer. Player/Missile graphics were specifically designed for this purpose, but P/M graphics have their limitations. First, you can only have a maximum of five players (objects) on the screen at once. Each player can only be eight resolution-lines wide. Also, you are limited to one color per player.

Hi-Res PUT and GET overcomes these limitations by allowing you to take a "snapshot" of any rectangular section of the screen and store it in memory. Then you retrieve the image, and reposition it on screen.

For example, if you wanted to move a box smoothly across the screen, you would start by drawing the box using PLOT and DRAWTO commands. Then, using the PUT and GET routine, save to memory the portion of the screen where you drew the box. And finally, put the image back on the screen shifted over by one pixel from the previous image.

Cycling through this procedure will create the illusion of smooth motion.

You can also cycle through changing images to give the illusion of animation.

GETTING STARTED

To see the routine in action, type in Listing 1. PUTGET.BAS, and check it with TYPO II. Be sure to SAVE a copy before you RUN it. This demonstration program will draw an object and move it smoothly around the screen.

PUT and GET works with all high-resolution graphics modes (3-11). It also supports ANTIC modes E and D. ANTIC E is the so-called 7 1/2 graphics mode. This mode offers fine resolution and four colors to work with. ANTIC E is available on the XL/XE computers as Graphics 15.

ANTIC D has greater resolution than ANTIC E but only allows one color. This mode is available on XL/XE computers as Graphics 14. Atari 400/800 users must set up a special Display List to access these modes. PUT and GET does not support any text modes.

USING THE ROUTINE

PUT and GET is written in machine language for maximum speed. It is called by the USR statement:

X = USR(1536, XSTART, YSTART, XWIDTH, YLENGTH, STORAGE, GRAPHICS MODE, COMMAND)

XSTART and YSTART are the upper left corner of the rectangular section of screen you want to work with.
XWIDTH and YLENGTH define the size of the rectangle. XWIDTH defines the width of the rectangle. YLENGTH defines the length of the rectangle. Both can range from zero to the maximum coordinate value for the graphics mode you’re in. For speed purposes, PUT and GET does not check for boundary errors, so be careful.

STORAGE is the area of memory you want to store your image in. To determine how much memory you’ll need, multiply XWIDTH by YLENGTH, then divide the result by the appropriate number listed on the table below:

<table>
<thead>
<tr>
<th>Graphics mode</th>
<th>Divide by</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

9-11 2
14 (ANTIC mode D) 8
15 (ANTIC mode E) 4

Add one to the result of the division if you get a remainder other than zero. If you are going to store your image in a string, add one to XWIDTH and YLENGTH before you multiply.

GRAPHICS MODE tells PUT and GET what graphics mode you’ll be using. Values range from 3 to 13. Values 3 through 11 are the standard high-resolution graphics modes. Use a value of 12 for Graphics 14 (ANTIC D), and 13 for Graphics 15 (ANTIC E).

COMMAND tells PUT and GET what action you want done. If COMMAND equals zero, PUT and GET will copy a section of the screen to memory. If COMMAND equals one, an area of memory will be copied to the screen.

PUT and GET takes up most of Page six and all of the cassette buffer. So if you do any cassette operations, you’ll have to reinitialize PUT and GET.

Machine language programmers can access PUT and GET with a JSR to $609, and by substituting these Zero Page locations for the USR parameters:

| XSTART | 224 (two bytes) |
| YSTART | 222 |
| XWIDTH | 220 (two bytes) |
| YLENGTH | 218 |
| STORAGE | 216 |
| GRAPHICS MODE | 214 |
| COMMAND | 212 |

Note that XSTART and XWIDTH should both be set up as two-byte values, even if the coordinate values are less than 255.

Brad Timmins is a freelance programmer living in Draper, Utah. His program Macro RESET appeared in the January, 1989 Antic.

Listing on page 38

---

**Bought An ST Computer?**

Then convert from Antic to START now!

As you upgrade your computer you'll want to stay with us. START magazine for the Atari ST guides you every step of the way.

You can have the remaining value of your Antic subscription converted to an equal value of START magazine with disk.

---

**YES!**

Convert my Antic Subscription to START magazine with disk.

I am interested.

Tell me how many issues of START I will get.

Clip this coupon and mail to:

Antic Subscriptions
P.O. Box 1569
Martinez, CA 94553

Name

Address

City State Zip

[WH9C]
Good King Zurp

The great pasta sauce chase... Don't ask! **Program by Frank Martone**

Collect the ingredients you need to create a super spaghetti sauce in this crazy BASIC game. Good King Zurp will work on any 8-bit Atari computer with at least 24K memory, with disk or cassette, and your trusty joystick.

Good King Zurp, son of Better King Xorpfg, was a powerful, despotic figure in the land of Xjigqh, where they do everything the same as they do here except that they eat gelatin with chopsticks.

During the reign of Xorpfg, the people of Xjigqh lived happy, prosperous lives, growing breadfruit and hiring themselves out as ballasts for the glowing Foon People across the Thpilj River.

Good King Zurp ascended to the throne of Xjigqh when Better King Xorpfg met an unfortunate end while stooping to pick a flower from a hostile grottelbush. Many suspected foul play, but everybody knew that Zurp couldn't possibly have master-minded any plan that required any type of native intelligence.

Zurp's first official act was to build—no, that's not right, his first official act was to legalize pouring beet syrup over your head and boiling grubs in mayonnaise and rutabaga oil. His second official act was to build a vast empire of financial wealth by hiring out his serfs as fruits and vegetables in traveling nutrional morality plays.

But unsurprisingly, hordes of toothpick craftsmen eventually overthrew Good King Zurp by disabling the palace guards, storming the king's rumpus room and overturning Zurp's pool table—with Zurp underneath, also thereby turning Zurp into a flapjack-like entity.

Zurp left Xjigqh reluctantly (and with constant neuralgia) for the sun-
nier pastures of Floobgrute, where he became the town's most popular rickshaw operator, but not before building his famous Horror Pits.

NOW TO THE GAME

The leader of the insurgents, one Fjorj "The Mad Whacko Strange Person" Lipsko, also met an unfortunate demise when he competed in a polo championship but forgot his pony. His will stipulated that the new ruler of Xjjgh would be that man, woman, child or similar creature that could develop a nationally accepted spaghetti recipe.

As luck would have it, buried in Zurp's Horror Pits are hundreds of valuable things, mostly on the order of ingredients necessary for a really good meat sauce for pasta. Your job, would-be despot, is to root through these horrid, glooppy areas for gold nuggets in the form of packages of noodles, containers full of seasonings such as pepper, and cloves of garlic.

If you're lucky—that is, if you survive long enough to tromp through enough pits—you might chance upon some packages of really terrific tortellini, which almost assuredly would put you in the winner's circle.

This sounds easy enough, but this is no mere scavenger hunt, oh, no! No, you've got to avoid poisoned colanders (sometimes called "cul-lenders") and elude horrid, oversized maggots which will cover bits of you with an unpleasant substance—and if you're covered completely, you'll suffocate, which is to say the game will end rather abruptly and tell you that you failed miserably.

Each screen gives you 75 seconds to grab as much as you can before advancing to the next screen. Your score, the time remaining and the number of lives left—you start with nine—are all displayed at the top of the screen.

GETTING STARTED

Type in Listing 1, PASTA.BAS, check it with TYPO II, and be sure to SAVE a copy to disk before you RUN it.

A title screen will appear, followed shortly by a list of objects and their point values. Press [START] to begin. Once the playing field is set up, simply move the joystick to direct your character towards the valuable items—but watch out for those maggots and colanders!

VALUES

Here's how much each ingredient is worth:

- Packets of pasta (fettucini, spaghetti, vermicelli, sometimes gnocchi): 100
- Packets of seasoning (including pepper, oregano and, for some reason, dried horseradish): 250
- Garlic Cloves: 500
- Packets of special pasta (tortellini, tortelloni, ravioli, sometimes agnolotti): 5,000
- Pesto sauce (there isn't any)
- Maggots you die
- Colanders (cul-lenders) you die

BONUS SCREENS

Every fourth screen is a bonus screen filled with rows of groaning colanders. Packets of pork-filled tortellini or spinach ravioli will appear randomly—and very briefly—onscreen. You must grab one before it disappears and pops up elsewhere.

If you've garnered over 60,000 points, first of all you should have a heck of a sauce recipe, but a nasty maggot will accompany you on the Bonus screens.

In order to get the bonus you must successfully grab three pasta packets without touching any colanders or maggots. If you fail, you go bonusless and advance to the next screen.

If you succeed, however, before heading on to the next screen you'll be given a random amount of points between 1,000 and 10,000.

The game gets tougher as your score increases. There'll be more colanders, more maggots and more spice packets and garlic cloves to grab. If you have over 50,000 points, the special pasta may appear on the playing field. If your score exceeds 100,000, the playfield will flash different colors, making it harder to concentrate.

STRATEGY TIP

Maggots tend to dissolve colanders in both the regular playfield as well as in the bonus playfield. Nobody knows why. Fool them into passing over the colanders to dissolve them.
**JUST RELEASED!**

**$99.95**

**SUPER ARCHIVER II**

(for ATARI 1050 drives)

NOW COPY all ENHANCED DENSITY programs plus retains all of the features of our World Famous SUPER ARCHIVER (see below). Allows you to COPY or CREATE simple or ENHANCED density protection schemes (including PHANTOM SECTORS). Completely automatic; compatible with the BIT-WRITER; the ULTRA BACKUP is a plug-in plug (comes complete with recorders, interfaces, printers, modems, cassette, etc.).

**$69.95**

**SUPER ARCHIVER**

(for ATARI 1050 drives)

The new SUPER ARCHIVER, obsolete all copying devices currently available for the ATARI 1050. It eliminates the need for Patches, POB files, Computer Hardware, etc. Copies are exact duplicates of original and will run on any drive, without exception, the SUPER ARCHIVER is the most powerful PROM PROGRAM COPY device available. It replaces ATARI 1050 installations of a plug-in chip and 6 simple solder connections. Softwares included.

Features are:
- **TRUE DOUBLE DENSITY**
- **ULTRA SPEED PLUS**
- **FULLY AUTOMATIC COPYING**
- **COMPENSATION**
- **FLEXIBLE COPY ATTACHMENTS**
- **COMPLETE LETTERS**
- **SUPPORTS EXTRA MEMORY**
- **EIGHT BASIC COMMANDS**
- **SCREEN DUMP TO PRINTERS**
- **TOGGLE HEX/DEC DISPLAY**
- **ADJUSTABLE SCREEN WIDTH**
- **SECTOR on TRACKRACKING**
- **AUTOMATIC DEVICE CHECKS**
- **ADJUSTABLE ☛ AUTOMATIC COPY**
- **DISPLAYS HIDDEN PROTECTION**
- **DASH COINTENTS**
- **AUTOMATIC SPEED COMPENSATION**
- **DESKTOP READY**
- **AUTOMATIC PHANTOM SECTOR MAKER**

**ARCHIVER/HAPPY ARCHIVER**

- Built-in EDITOR - reads, writes, displays up to 35 sectors/track
- Built-in MAPPING - up to 40 sectors/track
- Built-in DISASSEMBLER - up to 40 sectors/track
- Built-in MAPPER - up to 40 sectors/track
- DISASSEMBLY: Double Density HEADERS
- AUTOMATIC FORMAT LENGTH
- AUTOMATIC LENGTH CORRECTION
- SIMPLE INSTALLATION

**DEALER/DISTRIBUTOR/USER GROUP** Discount available call for Info.
PHONE Orders - MASTER CARD, VISA MAIL - Money Orders. Checks

**$39.95**

**RICHMAN'S**

80 Column Word-Processor!

Easy to use, very powerful, and NO ADDITIONAL HARDWARE required! Works with TV or Monitor! This "DISK ONLY" 80 Column Word-Processor is simple to use while offering numerous features such as:
- Completely Menu driven
- Self contained DOS
- Complete text formatting
- Built in Printer Drivers
- Page numbering
- Justify and EndAll
- High Speed Cursor
- Dictionary
- 80 COLUMN ON THE SCREEN!

One expert has compared 40 domestic and foreign word-processors and concluded that the RICHMAN'S 80 Column Word-Processor is the best! Only $39.95 plus $4 S/H/I. (404 required).

**$39.95**

**THE QUINTOPUS!**

The "QUINTOPUS!" is an inexpensive device that provides a tremendous amount of convenience, while eliminating the problems associated with the endless "daisy-chaining" of peripherals (e.g., printers, modems, cassette, etc.). The "QUINTOPUS!" is an I/O port expander that converts a single I/O output/input to five additional output/inputs. Instead of daisy-chaining all of your peripherals (which often times results in passing a signal through 30 feet or more of cable) the "QUINTOPUS!" allows each device to be connected directly to the computer through only three or four feet of cable. It is particularly useful when attempting to use ULTRA SPEED or WARSPEED data transfer rates. The "QUINTOPUS!" also provides the extra I/O ports often needed to connect devices having daisy-chaining capabilities. Cable resistance and capacitivity are greatly reduced thereby significantly improving the reliability of data transfer! Only $39.95 plus $4 S/H/I.

**$59.95**

**THE QUINTOPUS!**

(with SWITCHABLE PORTS!)

This Deluxe version of the "QUINTOPUS!" has all of the above features with the additional benefits of two SWITCHABLE PORTS! This means you can connect two computers to one printer or two printers to one computer; you can switch in a computerprinter combination while you’re printing out a long document, switch in a second computer to access a modem, disk drive, cassette deck, etc! Switch multiple combinations of peripherals or computers without the hassle or re-arranging i/o cables! ... simply flip a switch Only $59.95 plus $4 S/H/I.

**LIMITED SPECIALS**

Previously we listed these products under our FIRESALE ad but many of our customers feared that the software items might be smoke or water damaged. They’re NOT! All of the items listed below are BRAND NEW...only their literature or packaging may have slight cosmetic imperfections. All items are fully warranted! Quantities are limited! NO rainchecks!

<table>
<thead>
<tr>
<th>Item</th>
<th>Normally</th>
<th>Limited Specials</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. PILL (without case)</td>
<td>$7 0</td>
<td>$19 0</td>
</tr>
<tr>
<td>2. SUPER PILL (without case)</td>
<td>$80 2</td>
<td>$80 2</td>
</tr>
<tr>
<td>3. XL FIX ROM</td>
<td>$70 1</td>
<td>$70 1</td>
</tr>
<tr>
<td>4. XL MENDOS</td>
<td>$30 9</td>
<td>$30 9</td>
</tr>
<tr>
<td>5. DISKCRACKER (Newest version)</td>
<td>$50 9</td>
<td>$50 9</td>
</tr>
<tr>
<td>6. ELECTRONIC PHANTOM SECTOR MAKER DELUXE</td>
<td>$60 35</td>
<td>$60 35</td>
</tr>
<tr>
<td>7. RICHMANS 80 COLUMN WORD PROCESSOR</td>
<td>$29 20</td>
<td>$29 20</td>
</tr>
<tr>
<td>8. MIRACLE (Disk Only Version of Impossible)</td>
<td>$70 9</td>
<td>$70 9</td>
</tr>
<tr>
<td>9. IMPOSSIBLE for 800 or 800XL</td>
<td>$150 50</td>
<td>$150 50</td>
</tr>
<tr>
<td>10. XL MATE</td>
<td>$30 15</td>
<td>$30 15</td>
</tr>
<tr>
<td>11. COMPACTOR</td>
<td>$30 15</td>
<td>$30 15</td>
</tr>
<tr>
<td>12. KLIN II (Generic Happy Backup)</td>
<td>$100 75</td>
<td>$100 75</td>
</tr>
<tr>
<td>13. SILENCER</td>
<td>$30 19</td>
<td>$30 19</td>
</tr>
<tr>
<td>14. BLACK PASTE (MASTER)</td>
<td>$50 25</td>
<td>$50 25</td>
</tr>
</tbody>
</table>
| 15. BLACK PASTE DATA DISKS 1 & 2          | $20 10   | $20 10

*Order 5 Items or more and we will pay the freight.*

These SPECIALS are for C.S.S. customers only - NO DEALER DISCOUNTS - NO RAINCHECKS

- LIMITED QUANTITIES
Superhop ACTION!

Faster fractals grow in your Atari. By Douglas Skrecky

Watch elegant fractal patterns come alive onscreen at top speed with this fast and friendly ACTION! program. It works on 8-bit Atari computers with at least 48K memory and disk drive. If you are typing in the program, you will need the ACTION! cartridge from ICD. But if you own this month’s Antic Disk, you will find a version that runs without the special cartridge.

In May, 1987, Antic published my fractal generating program, Dot Hopper. That program used floating point arithmetic and was written in BASIC. Consequently, it was quite slow.

After playing with the ACTION! cartridge for awhile, I realized that floating point calculations were completely unnecessary for this language.

Between ACTION!’s inherent speed and integer arithmetic, I found that the program could be made to run many times faster than in BASIC. Using ACTION!, the Atari could generate fractal patterns in seconds. The fractals appear to “come alive” and grow right before your eyes.

You don’t have to know anything about the mathematics of fractal geometry to appreciate these elegant patterns. Some are startlingly organic, others are reminiscent of snowflakes and lace. If you enjoy the beauty of fractal imagery, you’ll have fun with this program.

GETTING STARTED

Type in Listing 1, SUPERHOP.ACT, and save a copy to disk before you start the program.

Antic Disk users don’t need the ACTION! language cartridge to enjoy Superhop ACTION!. A runtime version, SUPERHOP.EXE, is on this month’s disk. (As is usually the case, this runtime ACTION! translation would be too long to print as a type-in listing.) Copy SUPERHOP.EXE to another disk that has been formatted with DOS 2 or DOS 2.5, and make sure the disk contains a DOS.SYS file. Rename SUPERHOP.EXE to AUTO-RUN.SYS so that the program will load and run automatically.

To start growing fractals, turn off your Atari and insert your prepared Superhop ACTION! disk. Remove all cartridges (XL and XE owners press [OPTION]) and turn on your computer. Superhop ACTION! will load and run automatically.

When the program runs, it first presents the main menu, which lists
the various commands and options. You can return to this menu from any other screen simply by pressing [M].

Pressing [D] activates the demo mode, which generates successive fractals through 15000 iterations each (it doesn't take very long). Demo mode continues generating fractals until you press the [M] key to return to the menu, or press [C] to generate a single fractal.

While fractals are being generated in either Demo or Create mode, pressing the cursor keys (you don't have to press the [CONTROL] key with them) will shift the fractal so that you can view a different portion on the screen. However, the fractal is erased from the screen and the growth process must begin all over again—with the same pattern. The patterns are generated fast enough that this isn't a big inconvenience.

You may use the [>] key to "zoom in" on a portion of the fractal, enlarging it on the screen. The [<] key "zooms out". These keys also start the fractal growth process over again.

Pressing [C] changes the pattern. To see some of the possible patterns, try letting each pattern grow for only a few seconds before pressing [C] again. Then if a new pattern looks particularly interesting you can let it grow, and see how it develops.

Also, try letting patterns sit on the screen for several minutes after they seem to have stopped growing—they may surprise you.

**SAVE A FRAC TAL**

If you see a fractal pattern you like, you can save it to disk by pressing the [S] key. Doing so will read in a disk directory and let you choose a filename to save the screen under. The fractal is saved as a 62-sector Micro-Painter compatible file. Once the image has been saved, the program will return to the pattern in progress, which continues to grow without starting over.

To convert your Micro-Painter image to Micro Illustrator format, you can use Rapid Graphics Converter in the November 1985 Antic.

To load a fractal into Superhop ACTION! press [L]. A directory of the disk in drive 1 will be displayed. Enter the filename of the fractal you want, and it will appear on the screen. (Note that the fractal does not continue to grow after it has been loaded.)

Pressing the [SPACEBAR] will turn off the screen and speed up fractal growth by about 50%, although I realize that this is not an especially useful function!

---

Douglas Streecky is a mainframe programmer who currently lives in Vancouver, Canada. He likes to relax after work with his "trusty Atari 8-bit and perhaps a mug of beer."

Listing on page 37
Ian Chadwick’s “Mapping The Atari” has been one of the core references for Atari 8-bit programmers since the first edition was published in 1983. The book is set up as a comprehensive guide to the memory locations in the Atari 130XE, 65XE, 800XL, 600XL, 1200XL, 800 and 400. But it is much more than that. It is a virtual encyclopedia of indispensable information about the inner workings of the Atari.

This invaluable sourcebook essentially is out of print today, although a few copies of “Mapping The Atari” can still be found in the publisher’s warehouse and occasionally at specialty computer stores. Therefore, it is with great satisfaction that Antic announces we are beginning an exclusive serialization of key excerpts from the revised second edition of Chadwick’s book. This opening segment provides a useful overview of the material that will be covered in upcoming issues.
PEEK and POKE statements. Briefly, however, PEEK allows you to look at the value stored in any one memory location. If you want that value to be printed to the screen, you must prefix the PEEK statement with a PRINT statement such as:

PRINT PEEK (708)

If you haven’t changed your color registers, this will return the number 40 to your screen. All bytes in the Atari can hold a number between zero and 255. POKE allows you to place a value into a byte, such as:

POKE 755,4

By doing this you will have turned your text upside down! You can return it to normal by:

POKE 755,2

Similarly, POKE 710,80 will turn your screen dark purple! As with PEEK, POKE can only involve numbers between zero and 255. You will not be able to POKE into most of the ROM locations since the numbers in many of them are “hard-wired,” “burned” into the chip, and cannot be changed in this manner.

So how does the Atari (or other 8-bit microcomputers, for that matter) store a number larger than 255? By breaking it down into two parts, the Most Significant Byte (MSB), which is the number divided by 256 and rounded down to the nearest whole number, and the Least Significant Byte (LSB), which is the original number minus the MSB. The Atari knows to multiply the MSB by 256 and add the LSB to get the number. For example, the number 45290 is stored as two parts: 234 (LSB) and 176 (MSB). 176 times 256 equals 45056, plus 234 equals 45290.

LEAST-MOST STORAGE

The Atari uses the convention of storing addresses in the LSB/MSB manner in memory (i.e., the smaller part is in the first memory location). For example, locations 88 and 89 store the lowest address of the screen memory. Let’s say the numbers found there are 22 and 56, respectively. To get the decimal address, you take the MSB (stored in 89) and multiply it by 256, then you add it to the LSB at 88. In our case that’s 56 * 256 equals 14336, plus 22 equals 14358. This is the address of the upper left corner of the screen. A simple way to do this in BASIC is:

\[ \text{BYTE} = \text{PEEK (88)} + \text{PEEK (89)} \ast 256 \]

The reverse (to break up a decimal location into MSB and LSB) is done by:

\[ \text{MSB} = \text{INT} (\text{BYTE/256}); \text{LSB} = \text{BYTE} - \text{MSB} \ast 256 \]

This process is easier for assembly language programmers who use hexadecimal numbers, since the right two digits are always the LSB and the two left of them are the MSB. For example:

\[ \$D016 \text{ (hexadecimal for 53270)} \text{ equals } 16 \text{ (LSB) and } D0 \text{ (MSB)} \]

\[ \$16 \text{ equals 22 in decimal, and } D0 \text{ equals 208 decimal. Multiply the MSB by 256 and add 22 and you get 53270.} \]

Throughout the map portion of this book I have provided both decimal and hexadecimal numbers together for ease of reference. In 8K BASIC, you can use decimal numbers only with POKE, and PEEK will return only decimal values to you.

Hexadecimal is a base 16 used instead of the normal base ten system because it is more suited to the eight-bit structure of the computer. So, when we say 2175 in decimal, what we really mean is:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hex</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>

In hex, the same number is $87F$. That breaks down to:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>0</th>
<th>4096</th>
<th>256</th>
<th>16</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hex</td>
<td>0</td>
<td>8</td>
<td>7</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

Rather than multiply each next step up by ten, we multiply by 16. Okay, but where do we get “F” from? Well, if base ten has the numbers zero to nine, base 16 will have to have some letters added to the end to make up for the extra numbers:

So $F$ equals 15 in decimal. Now here’s how it all relates to binary math and bits:

Each byte can be broken up into two parts (nybbles), like this:

0000 0000

If each nybble is considered a separate number, in decimal, the value of each would range from zero to 15, or zero to $F$. Aha! So if all the bits in each group are on (one, or set), then you have:

1111 1111 15 15 15

Decimal

F  F  Hex

You join the two hex numbers together and you get $FF$ (255 in decimal), the largest number a byte can hold. So you can see how we translate bytes from binary to hex, by translating each nybble. For example:

1001 1101 10 13

Decimal

9  D  Hex

$9D$ equals nine times 16 plus 13, or 157 in decimal.

0100 0110

Binary

4  6  Decimal

4  6  Hex

$46$ equals four times 16 plus six,
or 70 in decimal.

<table>
<thead>
<tr>
<th>1111</th>
<th>1010</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>10</td>
<td>Decimal</td>
</tr>
<tr>
<td>F</td>
<td>A</td>
<td>Hex</td>
</tr>
</tbody>
</table>

$FA$ equals 15 times 16 plus ten, or 250 in decimal.

Obviously, it is easier to do this with a translation program or a calculator!

Since I will often be discussing setting bits and explaining a small amount of bit architecture, you should be aware of the simple procedures by which you can turn on and off specific bits in any location (that is, how to manipulate one of the eight individual bits within a byte). Each byte is a collection of eight bits: numbers are represented by turning on the particular bits that add up to the number stored in that byte. Bits can be either zero (0 equals off) or one (1 equals on, or SET). The bits are numbered zero to seven and represent the following decimal numbers:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>128</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The relationship between the bits and the powers of two should be obvious. Adding up all the numbers (all the bits are set) gives us 255. So each byte can hold a number between zero (no bits are set) and 255 (all bits are set).

Sometimes, instead of zero, no bits set is intended to mean 256. That will be noted in the relevant locations. So how do you set a bit? Simple: POKE it with the appropriate number. For example, to set Bit 5, POKE the location with 32. To set Bits 7, 5 and 4, add up their values, 128 + 32 + 16, and POKE the location with the total: 176.

Sometimes you need to set a bit without changing other bits already set, so you:

POKE number, PEEK (number) + decimal value for the bit to be set (i.e.,

POKE 50418, PEEK (50418) + 32)

To turn off a bit, instead of adding the value you would subtract it with POKE number, PEEK (number), minus the decimal value for the bit to be turned off. Binary math is simple and easy to learn; if you don't understand it now, you should do further reading on machine language before attempting any serious use of this guide.

**AND, OR, And EOR**

It is useful to know how to perform Boolean logic on bits. There are three functions used in assembly code for bit manipulation in this manner: AND, OR and EOR (exclusive OR). Each requires you to use two numbers, the one being acted upon and the one used to perform the function. Here is a brief explanation of how these logical functions work:

AND is usually used as a mask—to zero out unwanted bits. You compare two binary numbers using AND; if both bits in the same location are one, then the result is one. If either bit is zero, then the result is zero. For example:

<table>
<thead>
<tr>
<th>Original</th>
<th>Mask</th>
<th>AND</th>
<th>OR</th>
<th>EOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

In this case, 65 is the ASCII 'A'. By ORing it with 128, we get 193, the ASCII inverse 'A'.

EOR “flips” bits in the original if the mask has a one in the same location. For example:

<table>
<thead>
<tr>
<th>193</th>
<th>11000001</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOR</td>
<td>128</td>
</tr>
<tr>
<td>128</td>
<td>10000000</td>
</tr>
<tr>
<td>Result</td>
<td>01000001 = 65</td>
</tr>
</tbody>
</table>

In this case, we have returned the inverse 'A' to the normal ASCII value. An EOR with 255 (all ones) will produce the complement of the number:

<table>
<thead>
<tr>
<th>171</th>
<th>10101011</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOR</td>
<td>255</td>
</tr>
<tr>
<td>255</td>
<td>11111111</td>
</tr>
<tr>
<td>Result</td>
<td>01010100 = 84</td>
</tr>
</tbody>
</table>

Atari BASIC supports AND, OR and NOT; NOT is the logical complement where NOT1 equals zero and NOT0 equals one. If the expression is true, you get a zero; if NOT true, a one is returned—for example NOT (3 + 4) > 6) results in zero.

In general, I have attempted to avoid using 6502 assembly language mnemonics, but have included them where I felt their use described the action to be taken better than a lengthy explanation. Most common are JMP (jump to location), JSR (jump to subroutine), RTS (return from subroutine), and RTI (return from interrupt). Readers should be minimally familiar with machine language in order to understand any machine language subroutines used here.

You can’t hurt the machine by POKEing about in memory, although you may crash any program in memory, so SAVE your program first.
Usually you can salvage it by pushing [RESET], but you may have to turn off the machine and reboot on occasion. You can learn a lot about your machine by simply playing around with it.

ABOUT LANGUAGES

The majority of the information here concerns language-independent locations and can be used regardless of the language you use for your programming. When the location is language-dependent, such as the BASIC or DOS areas, I have noted it in the proper section. You may exert the same control over your Atari in whatever language you chose. You will obviously have to change the commands PEEK and POKE to the proper commands of your language.

BASIC is a good language to start with: you can use it to learn programming, to explore your computer, to experiment with, and to have fun with. However, when you are ready to go on, you will have to learn a more efficient, faster language if you really want to make the best use of your Atari. Many people choose 6502 machine language because of its speed.

Computer languages, whichever you use, are quite exact in their meaning, especially compared to English. Consider that in English, a fat chance and a slim chance both mean the same thing. Yet POKE, PUT, and PUSH have very different meanings in computerese.

GLOSSARY

ANTIC, CTIA AND GTIA, PIA, POKEY: Special Atari chips controlling the 400/800's graphics, color and screen resolution controller jacks and sound, respectively. Located in ROM, locations 53248 to 54783. ANTIC also processes the Non-Maskable Interrupts and POKEY processes the Interrupt Requests. These chips, along with the 6502 microprocessor which runs the rest of the Atari, are housed inside your computer, protected by the metal shielding underneath the plastic cover.

BIT, BYTE: A bit is the smallest size of division in your computer. It is so small that it can hold only one value in it: off (zero) or on (one). Eight bits together form a byte; this is the size of the memory locations discussed in this book. You will sometimes hear programmers talk about a half-byte called a "nibble."

CIO: Central Input/Output routines located in ROM. Controls Input/Output Control Block operations. Briefly, CIO handles the data input and output through the device driver(s) (also known as device handlers), then passes control to those drivers. It's a single interface with which to access all peripherals in a device-independent manner (i.e., uniform handling of data with no regard to the device being accessed). As an example: writing data to a disk file is treated in an identical manner as writing data to the screen; commas insert blanks between elements and both semicolons and commas suppress the End-Of-Line character (EOL).

DCB: Device Control Block, used by Serial Input/Output

DL: Display List. This is a set of instructions which tell the ANTIC chip where to find the screen display data and how that data is to be placed on the TV screen.

DLI: Display List Interrupt. A DLI causes the display to stop processing to temporarily run a user-written routine.

DOS: Disk Operating System. The software loaded from disk file DOS.SYS that controls all disk I/O. The latest edition of DOS is called DOS 2.05 ($ for single density).

DUP: Disk Utilities Package. The software loaded from disk file DUP.SYS that handles the DOS menu functions such as Copy.

EMS: (or sometimes DFMS): File Management System portion of DOS; a dedicated device driver that controls all I/O operations for device "D:"

FP: Floating Point mathematical package in ROM.

I/O: Input/Output.

IOCB: Input/Output Control Block. Area of RAM (locations 832 to 959) used by CIO to define operations to devices such as the disk drive (D-), printer (P-), screen display (S-), keyboard (K-) and screen editor (E-). ZIOCB is the page zero IOC.

IRQ: Interrupt Request used for serial port communication, peripheral devices, timing and keyboard input. IRQ's are processed by the POKEY chip.

NMI: Non-Maskable Interrupt; used for video display and RESET. NMIs are processed by the ANTIC chip.

OS: Operating System. The resident system that runs the Atari.

Although people often refer to the entire ROM area as the OS, this is not correct. The OS ROM is that portion of memory which holds the floating point package, the Atari character set, the device handlers, and both CIO and SIO. The actual operating system itself is the portion of the OS ROM which handles the I/O.

PMG, PM Graphics: Player/missile graphics. Players and missiles are special moveable, user-defined, colored screen objects. They are often used for games, animation, or special cursors. PM graphics are unique in that you can establish the manner (priority) in which they interact with the rest of the screen display and each other.

RAM: Random Access Memory. All memory below the OS area (0 to 49151) which is used for storage, programs, buffers, cartridges, DOS, IOCB, shadow registers, and registers for the special Atari chips. Random Access means you can get to and from these locations at random, not that they store information randomly!

ROM: Read Only Memory. That part of high memory (locations 49152 to 65535) in which the special hardware chips and the OS reside. ROM is also used to describe cartridge memory such as the 8K BASIC ROM, which cannot be user-altered (the cartridge ROM supersedes the RAM).
cannot alter most of the ROM, although some of the locations in the special Atari chips may be temporarily set to a new value.

With both RAM and ROM, we refer to areas with lesser values as being in "low" memory and locations with larger values as being in "high" memory.

**SIO:** Serial Input/Output routines located in ROM. Controls serial operations including the 850 interface (R:) and cassette recorder (C:). Briefly, SIO controls the Atari peripherals as per the request placed in its Device Control Block (DCB) by the proper device driver. It is also accessed by FMS for data transfer.

**VBI:** VBLANK interrupt. A VBI is an interrupt that occurs during the VBLANK interval, causing the computer to jump to a user-specified location to process a short user-written routine during the VBLANK process.

**VBLANK:** Vertical Blank. The interval between the time the TV electron beam turns off after reaching the bottom right corner of the screen and returns to the top left corner and turns back on again. This small time period may be used by machine language programmers for short routines without interrupting the display by writing a VBI (above). There are two VBLANK stages. Stage one is performed every VBLANK cycle (1/60 second). Stage two is performed either every 1/30 second or every 1/60 second when it doesn’t interrupt time-critical code being executed. See the end of the memory map for the processes executed in each stage.

Copyright 1983 and 1985, COMPUTE! Publications, Inc.

MAPPING THE ATARI
$16.95, COMPUTE! Books, P.O. Box 5406, Greensboro, North Carolina 27403, (919) 275-9809.

Ian Chadwick is a Toronto-based freelance writer.

---

### A Graphics Memory Map

This diagram is not to scale; it is merely meant to give you a visual idea of the structure of the Atari memory. The numbers on the right are the memory pointers; these locations point to the addresses shown. The numbers on the left are the actual locations in memory.

### Notes

The bottom of the BASIC RAM depends on whether or not you have DOS files loaded in. Without DOS, LOMEM should be 1792, with DOS 7420. If you increase or decrease the number of disk and sector buffers by modifying DOS, this value will change again. See locations 743, 744 and 1801, 1802.

The size and location of the variable, string and array tables depend on the program use and size. The more variables and arrays, the larger the memory the tables use.

The size and address of the Display List and screen memory depend on the GRAPHICS mode in use.

The first 256 bytes pointed to by LOMEM are the token output buffer. The actual BASIC program starts at the address pointed to by VNCY.

### Atari Timing Values

<table>
<thead>
<tr>
<th>clock frequency</th>
<th>1.79 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 machine cycle</td>
<td>0.558 µsec</td>
</tr>
<tr>
<td>1 frame</td>
<td>1/60 second</td>
</tr>
<tr>
<td>scan lines</td>
<td>262/frame</td>
</tr>
<tr>
<td>color clocks</td>
<td>228/scan line</td>
</tr>
<tr>
<td>color clocks</td>
<td>2/machine cycle</td>
</tr>
<tr>
<td>machine cycles</td>
<td>29868/frame</td>
</tr>
<tr>
<td>machine cycles</td>
<td>114/scan line</td>
</tr>
</tbody>
</table>

VBLANK time = 7980 machine cycles or less, depending on GRAPHICS mode. The shortest 6502 instruction requires two cycles; during that time the electron beam moves four color clocks.

### Horizontal blank time:

<table>
<thead>
<tr>
<th>Wide playfield</th>
<th>18 machine cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal playfield</td>
<td>34 machine cycles</td>
</tr>
<tr>
<td>Narrow playfield</td>
<td>50 machine cycles</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Pointers</th>
</tr>
</thead>
<tbody>
<tr>
<td>65535</td>
<td>Top of memory</td>
<td>Operating System ROM</td>
</tr>
<tr>
<td>60906-65535</td>
<td>Device handler routines</td>
<td>794-831 HATABS</td>
</tr>
<tr>
<td>59716-59805</td>
<td>Serial Input/Output (SIO) utilities</td>
<td></td>
</tr>
<tr>
<td>59099-59715</td>
<td>Interrupt handler</td>
<td>512,513 VDSLST</td>
</tr>
<tr>
<td>58534-59092</td>
<td>Central Input/Output (CIO) utilities</td>
<td>514-527 Vectors</td>
</tr>
<tr>
<td>58533</td>
<td>Operating System vectors</td>
<td></td>
</tr>
<tr>
<td>58496-58533</td>
<td>Initial RAM vectors on powerup</td>
<td></td>
</tr>
<tr>
<td>58488-58485</td>
<td>IMP vectors</td>
<td></td>
</tr>
<tr>
<td>58492-58447</td>
<td>Cassette</td>
<td></td>
</tr>
<tr>
<td>58416-58431</td>
<td>Printer</td>
<td></td>
</tr>
<tr>
<td>58400-58415</td>
<td>Keyboard</td>
<td></td>
</tr>
<tr>
<td>58384-58399</td>
<td>Screen</td>
<td></td>
</tr>
<tr>
<td>58390-58393</td>
<td>Editor</td>
<td></td>
</tr>
<tr>
<td>58367</td>
<td>ROM Character set</td>
<td>756 CHBAS</td>
</tr>
<tr>
<td>57944</td>
<td>Floating Point ROM package</td>
<td></td>
</tr>
<tr>
<td>57943</td>
<td>I/O chips</td>
<td></td>
</tr>
<tr>
<td>55295</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Contents</td>
<td>Pointers</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>54784-55295</td>
<td>Unused</td>
<td>766 CHBAS</td>
</tr>
<tr>
<td>54272-54783</td>
<td>ANITC</td>
<td>755 CHI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>564-565 LPMEN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>560-561 SDLSTL</td>
</tr>
<tr>
<td>54016-54271</td>
<td>PIA</td>
<td>559 SDMCTL</td>
</tr>
<tr>
<td>53780-54015</td>
<td>POKEY</td>
<td>638-639 PTRIG#</td>
</tr>
<tr>
<td>53554-53759</td>
<td>unused</td>
<td>632-638 STICK#</td>
</tr>
<tr>
<td>53248-53503</td>
<td>GTIA or CTIA</td>
<td>624-631 PADDL#</td>
</tr>
<tr>
<td></td>
<td></td>
<td>562 SSKCTL</td>
</tr>
<tr>
<td>53247</td>
<td>Unused 4K ROM block</td>
<td>16 POKMSK</td>
</tr>
<tr>
<td>49151</td>
<td>8K BASIC ROM</td>
<td>704-707 PCOLR#</td>
</tr>
<tr>
<td>or Left cartridge (A)</td>
<td>708-712 COLOR#</td>
<td>644-647 STRIG#</td>
</tr>
<tr>
<td></td>
<td></td>
<td>623 GPRIOR</td>
</tr>
<tr>
<td>40959</td>
<td>Top of BASIC RAM or</td>
<td>106 RAMTOP</td>
</tr>
<tr>
<td></td>
<td>RAMTOP</td>
<td>740 RAMSIZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAM used by OS and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cartridge.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(to bottom of RAM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1151</td>
<td>Operating System RAM</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>Cassette buffer</td>
<td></td>
</tr>
<tr>
<td>511</td>
<td>Printer buffer</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>Stack</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>OS page zero RAM</td>
<td></td>
</tr>
<tr>
<td>144,145</td>
<td>RAM used by DOS and File</td>
<td></td>
</tr>
<tr>
<td>MEMTOP</td>
<td>System Manager</td>
<td></td>
</tr>
<tr>
<td>142,143</td>
<td>Stack for FOR-NEXT &amp; GOSUB</td>
<td></td>
</tr>
<tr>
<td>RUNSTK</td>
<td>14,15 APPMHI</td>
<td></td>
</tr>
<tr>
<td>140,141</td>
<td>String &amp; array table &amp;</td>
<td></td>
</tr>
<tr>
<td>STARP</td>
<td>end of BASIC program</td>
<td></td>
</tr>
</tbody>
</table>

Size and location vary with GRAPHICS mode

Text window screen RAM 40800 for GR.0
Bottom of screen RAM 40000 for GR.0
Display List: 39968 for GR.0
Top of BASIC RAM 741,742 MEMTOP

User-program RAM

The amount of RAM can be ascertained by:

PRINT FRE(0)

Bottom varies: see note below
Depends on buffer area allocated.

RAM used by DOS and File System Manager

144,145 MEMTOP
Stack for FOR-NEXT & GOSUB 142,143 RUNSTK 14,15 APPMHI

Size and location vary with program size

String & array table & end of BASIC program 140,141 STARP

AUGUST 1989
Carnival of Cartridges

Food Fight, Karateka and lots more hits.

Reviewed by David Plotkin

FOOD FIGHT

Food Fight ($24.95) is a very simple (but messy) game. You control Charley, who starts out on the right side of the screen. The object is to guide him to the left side, where a dripping ice cream cone awaits him. Charley has 32 seconds to get across the screen, otherwise the ice cream melts, costing him one of his three lives. Sounds easy, right?

Well, of course, there is more to it than that! There are manholes strewn across the screen, and you must take care that Charley doesn’t fall into one. More importantly, cooks emerge from these manholes, and they chase Charley across the screen, trying to keep the poor boy from his just desserts. If one of these cooks touch Charley, there goes another life.

Fortunately, Charley is not defenseless. There are piles of food on the screen, and he can grab these (one pile at a time) and toss them at the cooks. If he hits a cook with a banana, tomato, pie, etc., the cook becomes so embarrassed he turns red and leaves the screen. Unfortunately, another soon replaces him.

Food Fight is a challenging (and frustrating) game because the placement of the food on the screen appears to be random. Sometimes the cooks start out closer to poor Charley than a pile of food does. The cooks can run faster than Charley, so unless he can get to a pile of food and load up with some throwing weapons, it’s curtains for him.

Still, the action is fast and the graphics are good. You’ll want to use a good joystick for this one.

KARATEKA

In Karateka you are an expert in the martial art of Karate. Your village has been destroyed and your intended carried off by the evil warlord, Akuma. You must defeat Akuma’s warriors in battle to rescue your sweetheart.

You can control the blond hero of Karateka with keyboard or joystick. Using the joystick, pressing the button quickly unleashes a punch, while holding down the button results in a kick. In the heat of battle it’s hard to control whether you punch or kick. Blows can be directed high, medium or low by moving the joystick in the appropriate direction before pressing the button.

The keyboard controls work quite well. The arrow keys direct your alter-ego forward and back, while an easily-reached set of six keys direct punches and kicks. Because each key is assigned its own function, you can achieve very precise control, which is certainly necessary in the upper levels of the game.
Atari's latest release of XE game cartridges—which run on all Atari XL/XE computers with at least 64K memory—revives a number of coin arcade or disk classics that had been unavailable in the Atari 8-bit market for some time.

As you defeat opponents, you begin moving towards the castle. Inside the castle, you will have to face more opponents, generally far more skillful than those you faced outside. You also have to figure out how to get past the diabolically deadly iron gate. And Akuma's war bird will attack to steal your strength.

Karateka is a multi-faceted game, requiring considerable arcade skill, plus the ability to recognize the attack patterns of your opponents. The graphics and animation are superb. The smoothly scrolling high-resolution background is extremely realistic, and the karate moves of both your hero and the opponents are true to life, right down to the swishing of the robes. And if you ever manage to see the final sequence (remember to bow to your intended...?) it will make all the hours spent with this game very worthwhile.

**CRYSTAL CASTLES**

In *Crystal Castles* ($24.95), you
guide Bentley the bear as he makes his way through a variety of three-dimensional mazes. The paths in these mazes are littered with gems, and Bentley must pick them all up before he can move to the next level. He can also pick up pots of honey for extra points.

A whole host of creatures try to prevent Bentley from achieving his life's ambitions. These creatures include animated trees, skeletons, and a swarm of bees. Bentley can grab a magic hat to protect him for awhile. He can also jump over many of his enemies.

The mazes in Crystal Castles are what set this game apart from others—previously as a coin-op hit and now as an XE game cartridge. In extra-sharp colors, the mazes simulate 3-D, with gem-laden paths on multiple levels connected by elevators. There are also tunnels, and when Bentley goes into a tunnel or behind a structure, he becomes an outline so that you can see where he is.

Controlling the game takes a little getting used to, because the joystick diagonals are used to guide Bentley through the maze. But the system can be mastered and the game is a lot of fun.

**DARK CHAMBERS**

**Dark Chambers ($34.95)** is one of the best dungeon arcade games released in XE cartridge. Using the joystick, you must guide your well-animated onscreen character though 26 mazes. He is armed with fireballs, and it's a good thing—these dungeons are full of dangerous beings.

At various places in each level "spawners" create the enemies, which are set up in levels of power. The most powerful is a Grim Reaper. Shooting one of these turns it into a Wizard. Shooting a Wizard turns it into a Wraith, then a Skeleton, and finally a Zombie. Shooting a Zombie kills it. Thus, some of your enemies must be shot five times to kill them!

The spawners keep creating creatures until you destroy them by shooting them. As you have probably have guessed, it takes five shots to destroy a Grim Reaper spawner (through Wizard spawner, Wraith spawner, etc.).

Fortunately, the enemies aren't very bright (But then, you aren't either, or you wouldn't be in this dungeon.) So they simply line up to be killed. You lose life energy if an enemy does touch you, and the game is over when your life energy is gone. You can recharge your energy by eating food and drinking potions which are found in the dungeon.

Other things can be found in the dungeon, such as the Keys necessary to open locked sections. Bombs which destroy all enemies on the screen, treasure, or extra weapons such as shields and more powerful fireballs are also to be found. But look out for the traps and poison, which all cost you life energy.

The smooth animation and fast action, coupled with the excellent playability (at the beginner setting, even a novice should be able to survive quite a while) make this a superb game. The 26 levels provide lots of variations as well.

**CROSSBOW**

**Crossbow ($34.95)** is a game for the Atari light gun. The opening screen shows a large number of locations including a desert, volcano, castle, town and jungle. From your current location, two or three paths are available, and these paths lead to other locales. Unfortunately, the paths don't show on the screen, so you'll probably want to map where the three color paths lead to so that you don't wander aimlessly around the screen!

Once you choose a path, the characters in your party (referred to as your "friends") set out on their journey. The scene switches to the selected location, and your party appears, moving leisurely across the screen while all heck breaks loose around them.

Large numbers of unfriendly creatures try to wreak havoc on your friends. Lightning bolts descend, ghosts appear, rocks careen down the path, attacks come from spiders, scorpions, alligators, pterodactyls (!) and man-eating plants—to name just a few of the different hazards in each location. Of course, if anything touches one of your friends, the poor soul is fried and no longer remains a member of your party.

So what is your job? Well, you must protect the party by shooting the obstacles out of the way with your light gun before they can do any harm. This can be quite a challenge, because not only is the Atari Light Gun not especially accurate, you are required to protect four or five characters from all those hazards.

The game becomes a little easier after some members of your party get fried, but when the last one is gone, the game is over. If you do extra well on a screen, another person is added to your party (lucky you!)

Crossbow, I must admit, is just as a gun to play. The effects are great (watch closely when a friend gets fried), the playability good and addicting—you'll keep coming back for more.

**THUNDERFOX**

Your home planet is under attack (again!). It's up to you in your Federation **Thunderfox ($24.95)** space fighter to destroy the alien transports bringing special crystals to power a deadly war machine.

This game is played in stages. When you first launch, you must approach the transport, dealing with formations of enemy fighters that try to both ram you and fire at you. Additionally, the transport has powerful guns, and part of the transport sticks up so high that you will crash if you try to fly over these portions of the ship.

Your fighter is armed with two weapons—guns which are effective against enemy fighters, and bombs which must be used against the transport. The transport has "ground in-
stallations” which are good for points but should be ignored because they are surrounded by obstacles you can crash into.

The object is to bomb the two “anti-gravity stabilizers”. Each one must be hit five times. Since you can’t carry ten bombs, you must return to your mother ship to reload, then return to finish the job.

If you can manage to disable the stabilizers, a door in the transport opens and you enter the thermonuclear laser room. You must maneuver through some very close quarters and avoid enemy lasers—hurrying all the while because of the room’s radiation. If you survive, the third level has you facing off against a large “Crystal Guardian,” which takes multiple hits to kill.

Thunderfox is simply too hard. Typical games last only seconds, especially since your fighter “bounces” off the invisible boundaries of the scrolling screen—usually into an obstacle. You can have just a single missile on the screen at once, which further limits your shooting.

**CHOPLIFTER**

The 64 delegates to the Peace Conference have been taken hostage by the Bungeling Empire and placed in locked bungalows. Your mission in Choplifter ($24.95) is to pilot a helicopter from a secret base (disguised as a post office) into enemy territory and rescue the hostages.

Your chopper can hold sixteen hostages, which means you must make multiple trips. Each trip gets harder because the enemy adds more sophisticated weapons to make your life miserable.

On the first level you face tanks, which are rather easily avoided by simply staying above their fire and blasting them into tiny pieces with your guns. The danger of the tanks, however, is that the gunfire will kill some of the hostages, so it is best to take out tanks as soon as possible.

You must blast open each bunga-

low. (Or let a tank do it for you, they’re really good at it.) Little figures will then run out and wave at you. Be careful where you set down your chopper, or you might crush a few hostages. When you land, the figures run to the helicopter and climb aboard. You can then fly back to base and unload for the next trip.

Starting with the second trip, more enemies show up—jets firing missiles and homing drones. The drones are rather easily destroyed, but the jets are tough—they come in fast and are hard to hit.

The helicopter can face forward, left or right. To switch directions, you must press the fire button and pull the stick in the direction you want to turn. Unfortunately, this is exactly the same maneuver you make in the heat of battle when you don’t want to turn! Thus, you end up changing directions a lot when you don’t want to, which can shorten your games considerably.

The graphics are incredible. The helicopter, enemies, background and base are highly detailed, and the rescued hostages even wave to you as they get off the chopper. Choplifter is a timeless classic on the Apple and Atari 8-bit computers, and the even more detailed graphics in this version help make it a very good game.

**INTO THE EAGLE’S NEST**

Three of your comrades have been captured in Into the Eagle’s Nest ($24.95). The Eagle’s Nest is a Nazi fortress where top officials are meeting to plan a counterattack. You must try to rescue the three saboteurs and blow up the castle with the explosives they planted. The castle contains locked doors, many floors, an elevator, chests, jewels, spare ammunition and many, many soldiers.

You must negotiate the castle (which looks suspiciously like a maze), blasting Nazi soldiers as you go. They, of course, shoot back (or more accurately, try to hit you), and if you are hit 50 times, you die and the game ends. You can recover from these hits by finding medical kits and food which have been rather carelessly left about the castle.

Pay close attention to ammunition too, because you can only carry 99 rounds, and you run out surprisingly quickly. If you manage to find your comrades, you must attempt to exit the castle with them. This is tough, because they are weak and sick and don’t move too fast.

The Nazi soldiers (This game should sell real well in Germany!) simply line up to be shot, much like the creatures in Dark Chambers. In fact, this whole game is quite similar to Dark Chambers, but not as playable. Your lack of ammunition makes it tough, and a single mistake (like shooting a box full of dynamite) can cost you the game! Overall, Dark Chambers is a better game of this same type.

**CRIME BUSTER**

Crime Buster is an exciting game that uses the Atari Light Gun and provides a “blasting” good time. You play one of the city’s top detectives, out to wipe out crime once and for all.

You choose a one- or two-player game by blasting holes in the appropriate choice. You can, of course, shoot holes in anything you want. The destructive effect is quite gratifying.

You must choose a section of the city from a map—just fire twice at the section you want. If the section of the city you choose is not adjacent to the section you currently occupy, you must travel to the portion of the city in question. Travel is very dangerous, and the best strategy is to avoid it. If you do have to travel, your car appears on the screen, traveling down a road which scrolls from left to right.

You control the car by firing at arrows pointing left and right, which makes your auto move towards that direction. Periodically a gangster car appears and fires at you. If it hits you, your car explodes and you lose one of your three lives.

AUGUST 1989
To fire back, you must shoot your light gun at arrows which determine your direction of fire. This is difficult, since you can’t control the car’s position and fire at the mobsters at the same time. It would have made more sense to let you fire at the mobster’s car with your light gun. If you somehow manage to hit the mobster’s car, it rolls over and explodes.

At your destination, your goal is to plug all the gangsters in a classic shoot-out, while at the same time avoiding the innocent bystanders who don’t have the good sense to get out of the way. Complicating matters, the gangsters sometimes dress up like innocent bystanders to try and fool you into not shooting them.

Each scene is different. Settings include buildings, a ship, and the inside of a warehouse. Gangsters appear in doorways and windows, pop out of boxes and manholes, and even come up from the water.

The graphics in Crime Buster are very good. The mobsters’ cars rolling over and over are very satisfying. When you blast the gangster with the hat on, his face changes and the hat flies off. Other effects are equally good. Overall, Crime Buster is an excellent game which will keep you coming back to try to do better.

**AIRBALL**

In one of the more bizarre scenarios for video games, you become the plaything of an evil wizard who turns you into an **Airball**—with a slow leak! In order to escape this fate, you must find a spellbook and return it to the wizard. Of course, the spellbook is hidden in a 150-room mansion, which also contains spikes, needles and other nasty, pointed objects detrimental to the health of an airball.

You move from room to room using your joystick and make your airball hop into the air by pressing the fire button. The rooms are viewed in three-quarter perspective, which can be confusing.

Bonus objects in the mansion include gold bricks and precious stones. There are also crates which the airball can move, to uncover valuable objects. You may also find such useful objects as a lantern, which can be quite handy in the rooms that have no light! When your pressure gets low, you must try to find an air pump— and then avoid death by overinflation.

The graphics in Airball are very well done and detailed, with many of the rooms containing elaborate decorations. Strategy definitely plays a significant part in winning this game, as it is quite easy to get lost in the huge mansion. Be warned, though. This is an extremely frustrating game, with death just the barest miscalculation away.

**SUMMER GAMES**

**Summer Games** presents eight Olympic events for you to compete in, including diving, skeet shooting, pole vault, two swimming events, gymnastics and track. You may choose to compete in all the events in order, compete in a single event, or practice any given event.

If you decide to compete, you must choose a name and country. Just highlight the flag of the country you want and that nation’s national anthem will play. The impressive opening ceremony has the Olympic theme playing, a runner lighting the torch, and doves flying away.

The events themselves basically test your skill and timing with a joystick. Some work pretty well, while others are exercises in torture.

The diving and gymnastics are similar in that you must press the joystick to various positions to control the amount of “tuck” your contestant uses. Both events end with scoring from the judges.

You dive from the ten-meter board, and you can choose a variety of dives (forward, back, reverse). You control the spinning, twisting diver on the way down. The more difficult the dive, the more possible points you can get.

The Gymnastics event has you leaping onto a springboard, then vaulting into the air (using a vaulting horse). Again, you control the gymnast as she leaps and flies through the air.

The first track event is a four-man relay. Here, you control the runner’s speed, trying to go as fast as possible without using up the runner’s stamina. There is also the small matter of passing the baton. The other track event is the hundred-yard dash, a “joystick buster” — just rattle the stick back and forth as fast as you can.

The two swimming events are similar. One is four-man relay, while the other is just a two-lap race for a single swimmer. You control the timing of your swimmer’s start and kick-turn, and apply power to the swimmer’s arms at exactly the right moment by pressing the fire button.

The best event is the skeet shoot. You control a cursor that represents the aiming point of your shotgun. A standard pattern of skeet launchers throw groups of up to four clay pigeons into the air and you must fire at them. This simple event works very well.

The most difficult and frustrating event is the pole vault. You must precisely time the placing of your pole, leaping up and over, and releasing the pole in order to make it over the bar without knocking the bar off. Unfortunately, correctly timing that final press of the fire button to let go of the pole seems virtually impossible.

No computer game can give you the actual feel of competing in a physical sport—but how many people can seriously hope to compete in the Olympics? Summer Games, with its excellent graphics, animation and sound will provide a good time overall for Olympic dreamers.

Atari Corp.
1196 Borregas Avenue
Sunnyvale, CA 94086
(408) 745-2000
$24.95 or $34.95, 64K XL/XE
SOFTWARE LIBRARY

TYING SPECIAL ATARI CHARACTERS

The Atari Special Characters and the keys you must type in order to get them are shown below:

For [CONTROL] key combination, bold down [CONTROL] while pressing the next key. For inverse [CONTROL] [A] through [CONTROL] [Z], press the [ ] key—or [ ] on the 400/800—then release it before pressing the next key. (Press [ ] or [ ] again to turn off inverse.) For [ESC] key combinations, press [ESC] and then release it before pressing the next key.

Carefully study the chart above and pay close attention to differences between lookalike characters such as the slash key's [ ] and the [CONTROL] [F] symbol [ ].

<table>
<thead>
<tr>
<th>NORMAL VIDEO</th>
<th>INVERSE VIDEO</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR THIS</td>
<td>TYPE THIS</td>
</tr>
<tr>
<td>Ctrl ,</td>
<td>Ctrl S</td>
</tr>
<tr>
<td>Ctrl A</td>
<td>Ctrl T</td>
</tr>
<tr>
<td>Ctrl C</td>
<td>Ctrl V</td>
</tr>
<tr>
<td>Ctrl E</td>
<td>Ctrl X</td>
</tr>
<tr>
<td>Ctrl G</td>
<td>Ctrl Z</td>
</tr>
<tr>
<td>Ctrl I</td>
<td>ESC CTRL -</td>
</tr>
<tr>
<td>Ctrl K</td>
<td>ESC CTRL +</td>
</tr>
<tr>
<td>Ctrl M</td>
<td>Ctrl ,</td>
</tr>
<tr>
<td>Ctrl O</td>
<td>Shift =</td>
</tr>
<tr>
<td>Ctrl Q</td>
<td>CLEAR</td>
</tr>
<tr>
<td></td>
<td>ESC TAB</td>
</tr>
</tbody>
</table>

TYPO II AUTOMATIC PROOFREADER

TYPO II automatically proofreads Antic's type-in BASIC listings. Type in the listing below and SAVE a copy to disk or cassette. Now type GOTO 32000. At the prompt, type in a single program line without the two-letter TYPO II code at the beginning. Then press [RETURN].

Your line will reappear at the bottom of the screen. If the TYPO II code does not match the code in the magazine, then you've mistyped your line.

To call back a previously typed line, type [ ], then the line number, then [RETURN]. When the completed line appears, press [RETURN] again. This is how TYPO II proofreads itself.

To LIST your program, press [BREAK] and type LIST. To return to TYPO II, type GOTO 32000. To remove TYPO II from your program, type LIST "D:FILENAME",0,31999, then [RETURN], then NEW, then ENTER "D:FILENAME", then [RETURN]. Now you can SAVE or LIST your program to disk or cassette.

NY 32100 POKE 842.13: STOP
CN 32110 POKE 842.12
ET 32120 ? "";'POSITION 11,1: "";'"";
CE 32130 C=0: ANS=C
QR 32140 GOTO 2.16: INPUT #3: LINES: IF LINE=":" THEN ? "LINE ":; B; "DELETED": GOTO 32050
UU 32150 FOR D=1 TO LEN(LINES(()]: C=C+1: ANS= ANS+"("&ASC(LINES(()]&D): NEXT D
NJ 32160 CODE=ANS-(CODE=676)
EH 32170 HCODE=INT(CODE=26)
BN 32180 LCODE=CODE-(HCODE=66)
HE 32190 HCODE=HCODE+65
HK 32200 XCODE=0: HCODE=HCODE+66
IE 32210 LCODE=POSITION 0.16: CHR+(HCODE)+CHR+(LCODE)
UG 32220 POSITION 2.13: "If CODE does not match press [BREAK] and edit line above: GOTO 32050

AUGUST 1989
FAST FRAC TALS GROW IN YOUR ATARI!

SUPERHOP ACTION!  Article on page 20

LISTING 1

;SUPERHOP ACTION!
;BY DOUGLAS SKEECY
;©1989, ANTIC PUBLISHING INC.

DEFINE BEGIN="DD", ENDWHILE="DD", ENDIF="FI", END="DD"

;MODULE
CARD ARRAY LINE(192)
BYTE ARRAY RSH3(320), FILENAME(16)
BYTE KEY=76, DISPLAY=559, IOTYPE=850
SCREEN SIZE='R1', 'R2', Z, MENUKEY,
CLEAR, LARGER, SMALLER, UP, DOWN,
LEFT, RIGHT, SPACEBAR, LOAD, SAVE, CREATE,
DELETE, DEMO, DEMOMODE, ON, OFF, PY
CALL CY, X, Y, XX, I, J, IC, JC, X0, Y0
CARD BUFFER ADDRESS=852, COUNT=PX
BUFFER LENGTH=856, SCREEN=88

;PROC INITIALIZE-PLOT
GRAPHICS8*16 SETCOLOR(2,0,0)
FOR PX=0 TO 80
LINE CY=SCREEN+40+PY
END FOR
FOR RX=0 TO 319 BEGIN
RSH3(PX)=PX RSH 3
END FOR
END

;PROC PLOT(PCD BY PLY)
BYTE POINTER PT
BYTE ARRAY ON=128 64 32 16 8 4 2 1
PT=LINE CY+RSH3(PX)
PT=PT+ON
CONNECT
RETURN

;PROC saves:].(8,64)
BYTE ARRAY XREG(16)

;PROC DISKDIR(8)
BYTE ARRAY FILES(520)
GRAPHICS8
SETCOLOR(2,9,0) SETCOLOR(4,9,0)
CLOSE(1) OPEN(1) "1", "D", 6, 0
PRINT(8)
RETURN

;PROC PRINT(8)
INPUT(1, FILES)
ENDWHILE ENDWHILE ENDWHILE
CLOSEC
RETURN

;PROC LOAD-FRAC TAL
DISKDIR
PRINT("T"|FILENAME TO LOGO.PIC | CLOSE)
CLOSE(1)
OPEN(1,FILENAME,4,0)
GRAPHICS8*16 SETCOLOR(2,0,0)
IOTYPE=7 BUFFER ADDRESS=SCREEN
BUFFER LENGTH=7680
CLOSE(1)
KEY=CLEAR
DO UNTIL KEY=LOAD OR KEY=DEMO OR
KEY=CREATE OR KEY=MENUKEY
END INITIALIZE-PLOT

;PROC SAVE-FRAC TAL
BYTE ARRAY SAVE=SCREEN(7680)
BYTE ARRAY SCREEN(7680)
GRAPHICS8 SCREEN SCREEN 7680
DISKDIR
PRINT("T"|FILENAME TO LOGO.PIC | CLOSE)
CLOSE(1)
OPEN(1,FILENAME,8,8)

INPUT(1, FILES)
WHILE FILES(16) "S BEGIN
PRINT(8)"}

continued on next page
ANIMATION WITHOUT PLAYER/MISSILES

HI-RES PUT AND GET

LISTING 1

CD 3 REM HIGH RESOLUTION PUT AND GET
VV 4 REM REVISION 1.1
XT 5 REM DRAW TIMMINS
FJ 6 REM <C>1989, ANTIC PUBLISHING INC.
DW 7 REM ADD ONE TO XWIDTH AND YLENGTH
UT 8 REM TO CALCULATE STORAGE SPACE IF
PS 9 REM IMAGE IS TO BE STORED STRING
RH 10 DIM IMAGE(1182):XSTART=11:YSTART=11
|XWIDTH=18:YLENGTH=19:HEGT=0:INPUT=1:GM
ODE=7:DX=1:DY=1:P=0
VV 20 GRAPHICS 7+16
MA 21 COLOR 4,8,0:SETCOLOR 1,3,4:SETCO
LOR 0,1,2
MG 40 COSUB 1000
Z 50 REM DRAW BORDERS********
UG 60 REM DRAW #1:DRAW 0,0:DRAWTO 159,0:DRA
W 0 159,95:DRAWTO 0,95:DRAWTO 0,0
FP 70 REM DRAW BALL**********
IF 80 C=1:R=2:R=8
CL 90 FOR I=0 TO 1

IF I>0 AND I<319 THEN PLOT(I,1)
ENDIF
IF I>0 AND I<319 THEN PLOT(I,1)
ENDIF

ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
EndElement

CT 10 FOR X=-9 TO R
KJ 110 IF C=1 THEN C=2:0=1:GOTO 130
LR 120 C=1:D=2
EJ 130 Y=50+R(R-RM)*
SJ 140 COLOR 1:COLOR 20+X,20+Y:DRAWT 20+X
Z 20+Y
HD 150 COLOR A:COLOR 20+X,20+Y:DRAWTO 20+X
Y 20+Y
LX 160 NEXT X
UM 170 REM **********SAVE IMAGES**********
QP 180 HGET=USR(1536).XSTART.YSTART.XWIDTH
JY 190 IF YSTART=1 OR XSTART=158-XWIDTH T
IM 200 REM **********MOVE BALL**********
HP 210 IF XSTART=1 OR XSTART=158-XWIDTH T
HE 220 IF YSTART=1 OR YSTART=94-YLENGTH T
80 DFS=DY=DX
WH 230 P=P+91:IF P=182 THEN P=0

38 ANTIC SOFTWARE LIBRARY
BASIC VARIABLE RENAMING TOOL

VRENAME

LISTING 1

Don't type the
TYPE II Codes!

HE 9 GOTO 31500
ML 31500 REM VARIABLE RENAMING UTILITY
UL 31501 REM BY DOUG WHITE
EU 31502 REM (C)1989 ANTID PUBLISHING INC
DF 31503 REM """"POSITION 2.2
ZC 31505 ? """"VARIABLE RENAMING UTILITY"

NH 31506 ? """"BY DOUG WHITE"

CE 31507 ?

DF 31508 CLR
KD 31509 DIM VARNAMES(2000),NAMES(128)
US 31510 DIM YNS(1)
II 31511 UNT=PEEK(130)+256*PEEK(131)
EL 31512 N=1:FLAG=0
NC 31513 REM ... G GET NEXT VARIABLE NAME
WJ 31520 BYTE=PEEK(UNT)
LD 31521 NAMES(N,3)=CHR$(BYTE)
NB 31522 REM ... END OF TABLE
ZB 31526 IF BYTE=0 THEN 31602
EQ 31528 REM ... CHECK FOR INVERSE CHAR.
NH 31530 IF BYTE<128 THEN UNT=UNT+1:N=N+1:
GOTO 31520
DM 31532 NAMES(N,3)=CHR$(BYTE-128)
MM 31534 IF NAMES(N)="VARNAMES" THEN 31602
PL 31536 ? ? NAME=$?
AN 31538 ? "DO YOU WANT TO CHANGE THIS VA-
MM 31540 IF NAMES(N)="VARNAMES" THEN 31602
PM 31542 IF YNS="Y" THEN 31558
UL 31544 REM ... PUT OLD NAME IN VARNAMES
UU 31546 NLEN=LEN(NAMES(N))
CP 31549 NAME(N,3)=CHR$(BYTE)
EP 31550 VARNAMES=NLEN+1,NLEN+1=NLEN+1
NZ 31552 UNT=UNT+1:N=N+1:NAMES=""
CU 31554 GOTO 31520
D1 31556 REM

31558 FLAG=1:REM ... DENOTES A CHANGE
HE 31560 NAMES=""
UL 31562 ? ? """"INPUT NEW VARIABLE NAME"
UA 31564 ? """"IN THE LAST CHARACTER OF A STR-
JZ 31566 ? "0 MUST BE 'S'"
UE 31567 ? """"IN THE LAST CHARACTER OF AN AR-
KH 31570 ? "E MUST BE 'E'"
DD 31572 INPUT NAMES
HP 31574 NLEN=LEN(NAMES(N))
EO 31576 IF BYTE=164 AND NAMES(NLEN,NLEN)
DJ 31578 IF BYTE=168 AND NAMES(NLEN)
FB 31580 IF BYTE=168 AND NAMES(NLEN)
BH 31592 NAMES(NLEN,NLEN)=CHR$(ASC(NAMES(N)))
CZ 31594 VARNAMES=NLEN+1,NLEN+1=NLEN+1
EL 31596 UNT=UNT+1:N=1:NAMES=""
LU 31598 GOTO 31520:REM ... GET NEXT NAME
BS 31600 REM ... PUT NEW NAME IN VARNAMES
DB 31602 REM ... STORE OLD POINTERS
AE 31618 UNT=PEEK(130)

continued on next page

AUGUST 1989

39
THE GREAT PASTA SAUCE CHASE...DON'T ASK!

GOOD KING ZURP

Article on page 17

LISTING 1

XZ 0 REM GOOD KING ZURP
XG #1 REM DAVE FRANK MARTONE
FF #2 REM ©1989. ANTIC PUBLISHING INC.
DW 3 DIM F$<13>:GRAPHICS 17:POKE 700,15:S
NP #9:POSITION 4,10
PU 4 ? #6:"ANTIC PRESENTS":GOSUB 1000:GOS
UB 36000:GOSUB 400
PC 6 GRAPHICS 17:TIME=9:SC=8:BR=30:DIFF=4
PD 0:6319:152=28:CTT=0185:53=1:54=23
PE 6 POKE 756.CH=256:55=0:56=3:65=16:GO=8
PF 7:SC=PEEK(88)+256*PEEK(89)
PG AF 7:GOSUB 11111:POKE 71,255:POKE 700,13
PH 7:GO=8 TO 22:POSITION 0,0:G#6:"GOOD
PK 8 FOR G=2 TO 7:POSITION 0,0:G#6:"GOOD
PL 9:GO=8 TO 22:POSITION 0,0:G#6:"GOOD
PM END
PN
PQ 5J 11 FOR B=1 TO BR$;K1=INT(RAND(0)*19):K2
PR ZM 12 ? #6:"O":SOUND 0.20:10.10:FOR HJ=0
PS TO 1:NEXT HJ:SOUND 0.0.0.0:NEXT B
PT FOR T=1 TO 16:G#1=INT(RAND(0)*19):E2=
PU INT(C0)+203+3:POSITION 1,E2
PV JJ 14 ? #6:"":SOUND 0.120.10:FOR HJ=1
PW TO 3:NEXT HJ:SOUND 0.0.0.0:NEXT T
PX CY 15 FOR C=1 TO G00:POSITION AND(0+18,R
PY ND C0*18.3:?:G#6:"SOUND 0.0.10.50
PZ UND 0.0.0.0:NEXT C
Q0 ZE 16 Y=10:Y=12:V=10:TIME=0:FOR G=10 TO 1
Q1 SJ 7:POSITION 9,G#:G#6:"NEXT G
Q2 TK 17 POKE SCR*X+20*Y.10:FOR G=1 TO 80.50
Q3 UND G=2.0.10:NEXT G:SOUND 0.0.0.0:PO
Q4 SITION 16.1.?:G#6:"TIME=30
Q5 XL 18 POSITION 0.0.0:G#6:COLOR 32:PLTH X.Y
Q6 QK 20 POSITION 9.1?:G#6:INT(LE):"":GO

THE GREAT PASTA SAUCE CHASE...DON'T ASK!

GOOD KING ZURP

LISTING 2

BL 10 REM SHELL SORT
BJ 15 REM BY DOUG WHITE
BK 17 REM ©1989. ANTIC PUBLISHING INC.
BL 20 DIM A<1000>
LR 30 GRAPHICS 0
LQ 40 ? "$ SHELL SORT"
LH 50 ? "$ HOW MANY # 5 ARE YOU SORTING
LK 60 INPUT NUM
LM 70 FOR I=1 TO NUM
LN 100 ? "### FILLING THE ARRAY WITH UNSORTED
LN 100 ED #:5"
LO 90 FOR N=1 TO NUM
LP 100 A(N)=INT(1000*SIN(N))
LQ 110 NEXT N
LR 120 ? "$ ### SORTING ###"
LS 130 POKE 180:POKE 19,0:POKE 20,0
LT 140 REM
LU 500 REM BEGINNING OF SHELL SORT
LV 510 PSIZE=100
LW 520 PSIZE=PSIZE*3+1
LX 530 IF PSIZE<NUM THEN S20
LY 540 REM
LZ 550 PSIZE=INT(PSIZE/3)
MA 560 ? "PARTITION SIZE = ",PSIZE
MB 570 REM
MC 580 FOR N=1:PSIZE TO NUM
MD 590 FOR PW=1:PSIZE = TEMP
ME 600 K=N-PSIZE
MF 610 FOR J=K TO 1 STEP PSIZE
MG 620 IF TEMP<ACJ THEN TEMP
MH 630 ACJ=PSIZE=S;
MM 640 NEXT J
MN 650 ACJ=PSIZE=TEMP
MO 660 NEXT N
MP 670 IF PSIZE<1 THEN S50
MQ 500 REM END OF SHELL SORT
MR 600 HU=1000 REM
MS 1810 T=PEEK20+256*PEEK(19)
MT 1920 SOUND 3.100.16.2
MU 1030 FOR N=1 TO NUM?:N.A=NEXT N
MV 1040 ? ?:T=60:" SECONDS"
MW 1500 END

THE GREAT PASTA SAUCE CHASE...DON'T ASK!

GOOD KING ZURP

LISTING 1

XZ 0 REM GOOD KING ZURP
XG #1 REM DAVE FRANK MARTONE
FF #2 REM ©1989. ANTIC PUBLISHING INC.
DW 3 DIM F$<13>:GRAPHICS 17:POKE 700,15:S
NP #9:POSITION 4,10
PU 4 ? #6:"ANTIC PRESENTS":GOSUB 1000:GOS
UB 36000:GOSUB 400
PC 6 GRAPHICS 17:TIME=9:SC=8:BR=30:DIFF=4
PD 0:6319:152=28:CTT=0185:53=1:54=23
PE 6 POKE 756.CH=256:55=0:56=3:65=16:GO=8
PF 7:SC=PEEK(88)+256*PEEK(89)
PG AF 7:GOSUB 11111:POKE 71,255:POKE 700,13
PH 7:GO=8 TO 22:POSITION 0,0:G#6:"GOOD
PK 8 FOR G=2 TO 7:POSITION 0,0:G#6:"GOOD
PL 9:GO=8 TO 22:POSITION 0,0:G#6:"GOOD
PM END
PN
PQ 5J 11 FOR B=1 TO BR$;K1=INT(RAND(0)*19):K2
PR ZM 12 ? #6:"O":SOUND 0.20:10.10:FOR HJ=0
PS TO 1:NEXT HJ:SOUND 0.0.0.0:NEXT B
PT FOR T=1 TO 16:G#1=INT(RAND(0)*19):E2=
PU INT(C0)+203+3:POSITION 1,E2
PV JJ 14 ? #6:"":SOUND 0.120.10:FOR HJ=1
PW TO 3:NEXT HJ:SOUND 0.0.0.0:NEXT T
PX CY 15 FOR C=1 TO G00:POSITION AND(0+18,R
PY ND C0*18.3:?:G#6:"SOUND 0.0.10.50
PZ UND 0.0.0.0:NEXT C
Q0 ZE 16 Y=10:Y=12:V=10:TIME=0:FOR G=10 TO 1
Q1 SJ 7:POSITION 9,G#:G#6:"NEXT G
Q2 TK 17 POKE SCR*X+20*Y.10:FOR G=1 TO 80.50
Q3 UND G=2.0.10:NEXT G:SOUND 0.0.0.0:PO
Q4 SITION 16.1.?:G#6:"TIME=30
Q5 XL 18 POSITION 0.0.0:G#6:COLOR 32:PLTH X.Y
Q6 QK 20 POSITION 9.1?:G#6:INT(LE):"":GO

THE GREAT PASTA SAUCE CHASE...DON'T ASK!

GOOD KING ZURP

LISTING 2

BL 10 REM SHELL SORT
BJ 15 REM BY DOUG WHITE
BK 17 REM ©1989. ANTIC PUBLISHING INC.
BL 20 DIM A<1000>
LR 30 GRAPHICS 0
LQ 40 ? "$ SHELL SORT"
LH 50 ? "$ HOW MANY # 5 ARE YOU SORTING
LK 60 INPUT NUM
LQ 70 FOR I=1 TO NUM
LN 100 ? "### FILLING THE ARRAY WITH UNSORTED
LN 100 ED #:5"
LO 90 FOR N=1 TO NUM
LP 100 A(N)=INT(1000*SIN(N))
LQ 110 NEXT N
LR 120 ? "$ ### SORTING ###"
LS 130 POKE 180:POKE 19,0:POKE 20,0
LT 140 REM
LU 500 REM BEGINNING OF SHELL SORT
LV 510 PSIZE=100
LW 520 PSIZE=PSIZE*3+1
LX 530 IF PSIZE<NUM THEN S20
LY 540 REM
LZ 550 PSIZE=INT(PSIZE/3)
MA 560 ? "PARTITION SIZE = ",PSIZE
MB 570 REM
MC 580 FOR N=1:PSIZE TO NUM
MD 590 FOR PW=1:PSIZE = TEMP
ME 600 K=N-PSIZE
MF 610 FOR J=K TO 1 STEP PSIZE
MG 620 IF TEMP<ACJ THEN TEMP
MH 630 ACJ=PSIZE=S;
MM 640 NEXT J
MN 650 ACJ=PSIZE=TEMP
MO 660 NEXT N
MP 670 IF PSIZE<1 THEN S50
MQ 500 REM END OF SHELL SORT
MR 600 HU=1000 REM
MS 1810 T=PEEK20+256*PEEK(19)
MT 1920 SOUND 3.100.16.2
MU 1030 FOR N=1 TO NUM?:N.A=NEXT N
MV 1040 ? ?:T=60:" SECONDS"
MW 1500 END
Tech Tips

THREE COLORFUL GRAPHICS DEMOS

By Mark Proudfoot

These three colorful BASIC graphics demonstrations were sent by Mark Proudfoot, 14, a high school student from Brentwood, California.

Easter Egg Maker produces a beautiful, flashing Easter egg, using pastel shades for excellent effect. Rainbow is a Graphics 11 demo with sound, producing a different set of "rainbows" each time. Pipe is in Graphics 9 and produces a pipe with water running out of it. The delicate shading gives an almost three-dimensional effect.

LISTING 1

FO 1 REM EASTER EGG MAKER
OF 2 REM BY MARK PROUDFOOT
TO 3 REM ©1989, ANTIC PUBLISHING INC
QH 10 GRAPHICS 7:POKE 752,1:POKE 768,14:C
ORL 1:POKE 710,268:POKE 712,268:POKE 709,0
CU 12 ? "The Easter Egg Maker by Mark Pro-
udfoot Press SELECT to create another
egg."
US 13 ? " Press SELECT to end." : IF
R F=1 TO 2
TW 15 R=1:1:IR=1:RP=1:NP=6,28318531:NP=360
DG 20 NX=60:NY=40:SIZE=42:DX=RP=NP=NP
PB 30 X=0:Y=SIZE
AC 40 FOR I=1 TO 80 STEP 1
YM 50 PLOT X+NX,Y+NY: PLOT NX+X,Y+NY-Y: PLOT
B 80: IF F=2 THEN GOSUB 200
KA 70 X=I: Y=I-1: X=I+1: Y=I-1: X=I+1: Y=I+1:
BH 70 TRAP 130:NEXT I:NEXT F:GOTO 190
QE 80 IF F=1 THEN RETURN
QZ 90 GOSUB 160
TY 92 SETCOLOR 1, RND(I)*15, 14
SM 100 POKE 765, RND(I)*3
UT 100 XI=18,y=0,r="5"
ZB 120 RETURN
PB 130 F=2:GOTO 15
QW 150 REM
BQ 160 SETCOLOR 1, RND(I)*15, 14
CG 170 SETCOLOR 1, RND(I)*15, 14
ZN 180 RETURN
CO 190 GOSUB 160:GOSUB 200: GOTO 190
DU 200 IF PEEK(32793)=6 THEN RUN
NU 210 IF PEEK(32793)=5 THEN GRAPHICS 8: E
NE 220 RETURN

LISTING 2

RW 1 REM RAINBOW
OF 2 REM BY MARK PROUDFOOT
TO 3 REM ©1989, ANTIC PUBLISHING INC
UR 18 GRAPHICS 11:F=3:POKE 712,13

LISTING 3

PA 5 REM PIPE
LJ 6 REM BY MARK PROUDFOOT
TX 7 REM ©1989, ANTIC PUBLISHING INC
QR 10 GRAPHICS 9:POKE 712,16:DEG
MN 40 FOR B=1 TO 61
PZ 50 FOR A=10 TO 17:COLOR A-9
JT 60 PLOT A:B:NEXT A
TU 70 FOR A=18 TO 25:COLOR 25:(A-4)
TS 80 PLOT A:B:NEXT A
DJ 90 FOR A=31 TO 46:COLOR K
JY 100 PLOT 25-INT(K*3),A:DRAINTO 48.A
PP 110 NEXT A:C=16
CX 125 FOR A=46 TO 61:COLOR K
JJ 140 PLOT 25-INT(K*3),A:DRAINTO 48.A
AF 125 NEXT A:C=16:SOUND 1,25,0.2
TC 130 X=41:Y=INT(A)AND(0)*3+5:G=INT(A)*
2+2
HE 135 SOUND 0:PEEK(3770)*10.2:SOUND 0,P
LE 140 COLOR KAND(0)*8+1:G=6*1.5:PLOT X,
Y
MK 150 X=X+2:Y=Y-6
BU 160 IF X>778 OR Y>179 THEN GOTO 130
PJ 170 GOTO 135

Antic pays $25 for every original and exclusive Tech Tip submission that we publish. Send your 8-bit or ST disk and printout to: Antic Tech Tips, 544 Second Street, San Francisco, CA 94107. Tech Tips welcomes very short programs that demonstrate the Atari's powers, simple hardware modifications, or useful macros for popular software.
Antic Classifieds

Munich Atari XL/XE Public Domain Association is looking for members. Great PD selection.
We sell ($2/25), change and buy PD-software.
Free list. Write to: Florian Bechter, August-Popp-Str.17, 8000 Munich 50, West-Germany (FRG) (8/89)

Try us for your Atari Public Domain software needs. Good prices, fast service. Write for free catalog. Vulcan Software, PO. Box 692 Manassas, VA 2211-0692 (9/89)

PRINTSHOP GRAPHICS. Holidays $15.00. LDS Prophets and Temples $15.00. LDS Church Auxiliaries $15.00. 120 pictures each. Free Catalog. PCS Software Box 10574 Kears, UT 84418. (10/89)

130 XE, XEP80, 1.C.D. MIO (256K) & CABLES. ALL DOCS. IN ORIGINAL CARTONS. USED LESS THAN 6 MONTHS. $375.00. ALSO ADAP'TEC 4000A HD CONT. & 50 PIN SCSI CABLE. $60.00. AFTER 6 P.M. CDI 205-649-5004. (8/89)

Treasure Hunters Wanted to play a new action text game. Cash Prizes for info SASE MEADOWBROOK GAMES 30860 Alicante, Homeland, CA 92348 (8/89)

10 disks with the best P.D. games ever made for the Atari 8-bit for only $10.00 plus 2.00 S/H. Order NOW! E. Lopez, PO. Box 6772, Pico Rivera, CA 90661-6772 (8/89)


EXCELLENT PD SOFTWARE FOR 800/XL/XE. Large selection, low prices, and fast service. Send SASE: CLASSIC SOFTWARE, PO. BOX 373, RUNNEMEDE, NEW JERSEY 08078 (8/89)

BEST PD DISKS/PRICES—fastest service—FREE bonuses—8-bit ST. Automatic DISK-A-MONTH program. Full disks less than $1.50 each. Super sample disk with catalog $3.50. Specify computer. PATNOR, P0B 370782, El Paso, TX 79937 (8/89)

ENJOY selected PD software! Games—productivity—utilities—more! Demo disk, $3.00. catalog FREE. Homemade, 6011 Hyde Park Circle, Suite 208, Jacksonville, FL 32201 (8/89)

MUSIC LOVERS! MIDI YOUR ATARI!! We have music software, Midi interfaces, and hundreds of pre-programmed songs available for the Atari XL/XE and Atari ST. For catalog send $1.00 to The MIDI Inn, PO. Box 2362, Dept. AN7, Westmont, IL 60559. Specify your computer, music software and Midi instruments on hand. (9/89)

CREATE/PRINT MUSIC MANUSCRIPT. 48K 800/XL/XE. Words and music. WYSIWYG. Epson, Star, Panasonic printers, or easily create your own driver. Very high quality display and output. A SERIOUS TOOL for musicians; a NEAT TOY for music-lovers. Written by an M.A. in music. $19.95. 30-day MONEY BACK GUARANTEE. John Kelleher, 4100 New Jersey Street, San Diego, CA 92116 (10/89)

Your choice of 8-bit P.D. software on each side of every disk. Great selection, competitive prices. Send SASE for catalog. Dino’s RYO, 664 W. Arrow Hwy, Suite 197, San Dimas, Calif. 91773 (8/89)

CHRISTIAN SOFTWARE AND PUBLIC DOMAIN PD low as $1.80/4 disk. Send SASE for Catalog. AGAPE GAMES Disk five Christian Games $12.95 (8 Bits W/48K). Dan Sharpe GPO Box 911 Chicoเกม MA 01021 (9/89)

USED 8-BIT EQUIPMENT, SOFTWARE, ACCESSORIES, SEND S.A.S.E. (45 CENTS) BUSINESS SIZE, OR 45 CENTS STAMPS FOR PRICES AND BIG LIST. L. CHRISTOFFER, 1806 PATRICK, BURLINGTON, WA 98233 (8/89)

Stop overpaying for P.D. Software!!! ALL 8-bit disks only $2.00 each, ST only $3.00. Quantity discounts, great selection. Specify computer. Send large SASE: MWFPS, 890 N. Huntington St., Medina, OH 44256 (8/89)

SI gives you MORE for LESS! Our SUPERIOR ATARI 800/XL/XE P-D Theme Collections average 35 programs (7 cents each) per D-S disk. Shipping FREE. Send only $3.00 for MLX GAMES’ GREATEST HITS (our Most Popular title)—receive SI’s newest Catalog FREE! Catalog alone $2.00. SOFTWARE INFINITY, 642 East Waring Avenue, State College, PA 16801 (7/89)


Atari 800/XL/XE public domain disks. High quality disks, fast service, free list. High quality disks, fast service, free list. 1st Byte, PO Box 130822, Tyler, TX 75713 (8/89)

250+ DS/DD DISKS OF PD SOFTWARE FOR YOUR ATARI 8-BIT. PLEASE SEND $3 TO: AAPDS 6-18TH EDGEMORE, HUTCHINSON, KS 67502 (8/89)


BATTERY BACKED UP RAMDISK. Never lose your memory, Boot from Ramdisk! No installation. Ramdisk cartridge plugs into Atari 400/800 or XL/XE. 32K $99.95, 64K $129.95, 128K $249.95 Certified check, Money order, VISA accepted. CANOE COMPUTER SERVICES, 1100-155 St., Edmonton, AB, T5P-2N3 Phone (403) 437-4619 (8/89)

Advertisers Index

<table>
<thead>
<tr>
<th>Advertiser</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA SYSTEMS</td>
<td>11</td>
</tr>
<tr>
<td>AMERICAN TECHNOVISION</td>
<td>2</td>
</tr>
<tr>
<td>ANTIC</td>
<td>46,47</td>
</tr>
<tr>
<td>B &amp; C COMPUTERVISION</td>
<td>IBC</td>
</tr>
<tr>
<td>C.S.S.</td>
<td>19</td>
</tr>
<tr>
<td>SAN JOSE COMPUTER</td>
<td>6</td>
</tr>
<tr>
<td>START</td>
<td>4</td>
</tr>
</tbody>
</table>

Advertising Deadlines

October 1989

Insertion Orders: June 30
Artwork: July 7
On Sale: Last week of August

November 1989

Insertion Orders: August 1
Artwork: August 8
On Sale: Last week of September
BUY
SELL
TRADE
SWAP

Antic Classifieds
544 Second Street, San Francisco, CA 94107
Telephone (415) 957-8886

An ad in Antic reaches more than 100,000 serious Atari users

SUPER BONUS PROGRAM
on Antic Disk!

You get even more value when you get the Antic disk... a Bonus Program that’s too large to type-in program to fit in the magazine.

Plus you get all 172K of this month’s Antic programs on disk. Great 8-bit software without typing!

Order your disk now!
Phone toll-free:
1-800-234-7001
(Visa/Mastercard orders)
Ask for ADS889

Only $5.95
(Plus S&H shipping & handling)
BACK ISSUE SALE
Complete Your Atari "Toolkit"

SAVE 25% TO 35%

How To Order:

Simply give the Order Numbers of the issues you want, and include the letter corresponding to the following selections:

A- magazine & disk
B- magazine only
C- disk only

Example: 1A- is for "April 1983, magazine & disk"

If no letters are indicated in your order, we will automatically send you the magazine only (B).

Prices:

<table>
<thead>
<tr>
<th></th>
<th>1 To 5 Issues</th>
<th>If you order more than 5 Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magazine &amp; Disk</td>
<td>$7.00 each</td>
<td>$6.50 each</td>
</tr>
<tr>
<td>Magazine only</td>
<td>$3.00 each</td>
<td>$2.75 each</td>
</tr>
<tr>
<td>Disk only</td>
<td>$5.00 each</td>
<td>$4.75 each</td>
</tr>
</tbody>
</table>

Please include shipping & handling charges:
1 to 5 issues: $1 for each issue
6 and over: $5 plus 50c for each issue over 5.

FOR ORDERS ONLY CALL TOLL-FREE (800) 234-7001 (7-5pm PST.)
MasterCard & Visa Only

Or send your check to: ANTIC BACK ISSUES
544 Second Street San Francisco, CA 94107
California residents add 6 1/2% sales tax.

---

Back Issue Order Form

<table>
<thead>
<tr>
<th>Order #</th>
<th>Selection A, B, or C</th>
<th>Order #</th>
<th>Selection A, B, or C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Issues Ordered:</td>
<td></td>
<td>Total Issues Ordered:</td>
<td></td>
</tr>
</tbody>
</table>

| No. Issues of Sel. A: | x $7 ea. | x $6.50 ea. | $ |
| No. Issues of Sel. B: | x $3 ea. | x $2.75 ea. | $ |
| No. Issues of Sel. C: | x $5 ea. | x $4.75 ea. | $ |

Total $ Shipping & handling $ Amount enclosed $

☑ Check Enclosed ☑ VISA ☑ MasterCard

Name _______________________________
Address ________________________________________
City ______ State ______ Zip ______
Credit Card # ___________________________
Exp. Date ___________________________

Canada, Mexico and all other foreign add $1 per issue. Payment in U.S. funds must accompany all foreign orders. California residents add 6 1/2% sales tax.

 ремонт

Antic

the atari resource
<table>
<thead>
<tr>
<th>ORDER #</th>
<th>ISSUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>April 1983: Games—3-D Maze, Voyeur (No Disk)</td>
</tr>
<tr>
<td>2</td>
<td>May 1983: Telecomputing—Microids, Tele Chess (No Disk)</td>
</tr>
<tr>
<td>3</td>
<td>June 1983: Databases—Stargazing, Dancin’ Man (No Disk)</td>
</tr>
<tr>
<td>4</td>
<td>July 1983: Adventure Games—Dragonsmoke, Shoot-em-up Math</td>
</tr>
<tr>
<td>5</td>
<td>August 1983: Graphics—3-D Fuji, Keystroke Artist</td>
</tr>
<tr>
<td>6</td>
<td>September 1983: Education—P/M Tutor</td>
</tr>
<tr>
<td>7</td>
<td>October 1983: Sports Games—AutoCassette</td>
</tr>
<tr>
<td>8</td>
<td>November 1983: Sound &amp; Music—Air Raid, Casting Characters</td>
</tr>
<tr>
<td>9</td>
<td>December 1983: Buyer’s Guide—AUTORUN.SYS, Automate Player/Missiles</td>
</tr>
<tr>
<td>10</td>
<td>January 1984: Printers—Pocket Calendars, Screen Dump</td>
</tr>
<tr>
<td>11</td>
<td>February 1984: Personal Finance—TYPO, Gauntlet</td>
</tr>
<tr>
<td>13</td>
<td>April 1984: Games—Risky Rescue, Math Wizard</td>
</tr>
<tr>
<td>14</td>
<td>May/June 1984: Exploring XL Computers—Escape from Epsilon, Scroll to the Top</td>
</tr>
<tr>
<td>15</td>
<td>July 1984: Communications—AMODEM, BASIC Animation Secrets</td>
</tr>
<tr>
<td>16</td>
<td>August 1984: Disk Drives—Horseplay, Recall</td>
</tr>
<tr>
<td>18</td>
<td>October 1984: Computer Learning—Bouncing Ball, Antic 4/5 Editor/Animator</td>
</tr>
<tr>
<td>20</td>
<td>December 1984: Buyer’s Guide—Infobits, Bifidrop</td>
</tr>
<tr>
<td>21</td>
<td>January 1985: Super Utilities—TYPO II, DISKIO</td>
</tr>
<tr>
<td>22</td>
<td>February 1985: Finances—Home Loan Analyzer, Drum/Bass Synth</td>
</tr>
<tr>
<td>23</td>
<td>March 1985: Printers—Kwik Dump, Font Maker</td>
</tr>
<tr>
<td>24</td>
<td>April 1985: Computer Frontiers—Dot Matrix Digitizer, Speech Editor</td>
</tr>
<tr>
<td>25</td>
<td>May 1985: New Super Ataris—Son of Infobits, Arena Racer</td>
</tr>
<tr>
<td>26</td>
<td>June 1985: Computer Arts—View 3-D, The Musician</td>
</tr>
<tr>
<td>27</td>
<td>July 1985: Computer Challenges—Miniature Golf, Guess That Song</td>
</tr>
<tr>
<td>29</td>
<td>September 1985: Power Programming—One-Pass Disk Copy 130, Crickets</td>
</tr>
<tr>
<td>30</td>
<td>October 1985: Mind Tools—Graph 3D, GEM Color Cascade</td>
</tr>
<tr>
<td>31</td>
<td>November 1985: New Communications—TYPO II Double Feature, 130XE Memory Management</td>
</tr>
<tr>
<td>32</td>
<td>December 1985: Shoppers Guide—DISKIO Plus, Box-In</td>
</tr>
<tr>
<td>33</td>
<td>January 1986: Atari Products are Back—Appointment Calendar, Dungeon Master’s Apprentice</td>
</tr>
<tr>
<td>34</td>
<td>February 1986: Printer Power—T-Shirt Construction Set, Forth Escapes</td>
</tr>
<tr>
<td>35</td>
<td>March 1986: Practical Applications—Lunar Lander Constructor, Lie Detector</td>
</tr>
<tr>
<td>36</td>
<td>April 1986: Computer Mathematics—Fractal Zoom, 3-D Fractals</td>
</tr>
<tr>
<td>37</td>
<td>May 1986: 4th Anniversary—Digital Gardener, Molecular Weight Calculator</td>
</tr>
<tr>
<td>38</td>
<td>June 1986: Summer Computing—Weather Wizard, Bomb Squad</td>
</tr>
<tr>
<td>40</td>
<td>August 1986: Online Communications—Ultrafont, Floppy Filer</td>
</tr>
<tr>
<td>41</td>
<td>September 1986: Weather—WEFAX Decoder (8-bit/ST), BASIC Tracer</td>
</tr>
<tr>
<td>42</td>
<td>October 1986: Hard Disks—Video Stretch, TYPO ST</td>
</tr>
<tr>
<td>43</td>
<td>November 1986: Personal Finance—Budget database, V-Graph</td>
</tr>
<tr>
<td>44</td>
<td>December 1986: Shoppers Guide—Stepper Motors, Nuclear Waste Dump</td>
</tr>
<tr>
<td>45</td>
<td>January 1987: Talking Atari—Talking Typewriter, Rebound</td>
</tr>
<tr>
<td>47</td>
<td>March 1987: Dvorak Keyboard, Multi-AUTORUN</td>
</tr>
<tr>
<td>48</td>
<td>April 1987: — Designer Labels, Taxman</td>
</tr>
<tr>
<td>49</td>
<td>May 1987: 5th Anniversary—A-Rogue, Poker Slot Machine</td>
</tr>
<tr>
<td>50</td>
<td>June 1987: Animation—Verbot Commander, Citadel</td>
</tr>
<tr>
<td>51</td>
<td>July 1987: Print Anything—Ghost Writer, Your Net Worth</td>
</tr>
<tr>
<td>52</td>
<td>August 1987: Atari Muscle—Sideways Spreadsheet, Diamond Dave</td>
</tr>
<tr>
<td>53</td>
<td>September 1987: Work/Play—Mighty Mailer, Maximillian B.</td>
</tr>
<tr>
<td>54</td>
<td>October 1987: Football Predictor—Antic Prompter (8-bit/ST), Spelling Checker</td>
</tr>
<tr>
<td>55</td>
<td>November 1987: Practical Applications—Critical-Path Projects, WYSIWYG Cassettes</td>
</tr>
<tr>
<td>56</td>
<td>December 1987: Print Holiday—P.S. Envelope Maker, Antic Publisher</td>
</tr>
</tbody>
</table>