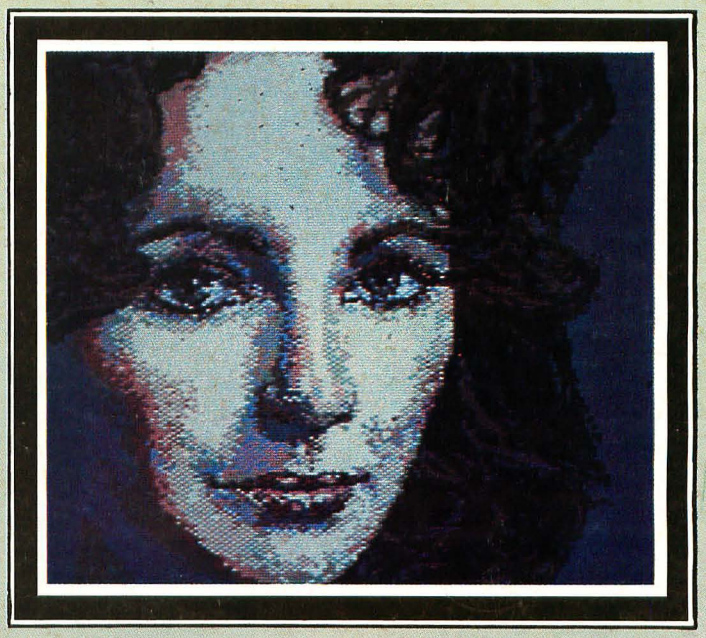
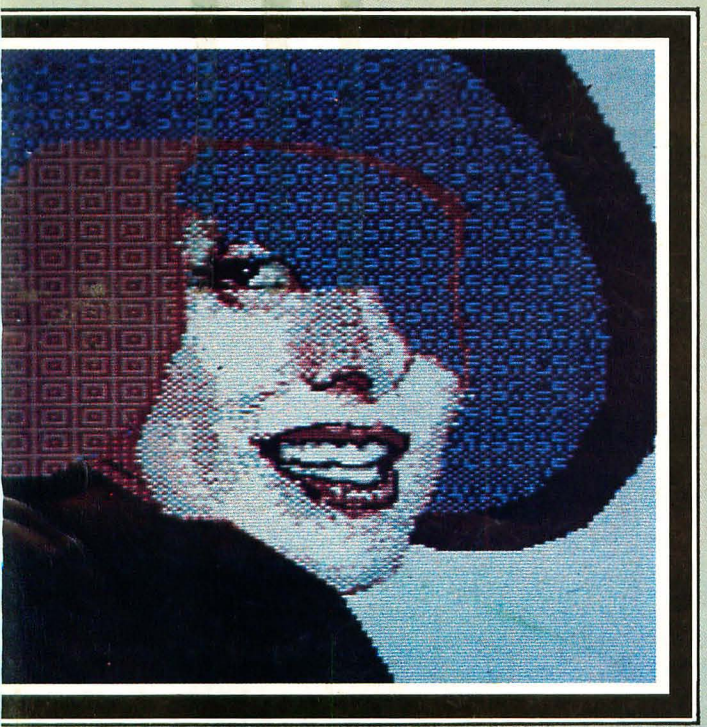
SoftSide™

#33
THREE DOLLARS

4th ANNIVERSARY ISSUE

Computer Graphics





- SHAPE WIZARD
- SPIRALGRAPHICS

"SOFTALK" REVIEWED HOME FINANCIAL PROGRAMS AND HERE'S WHAT THEY HAD TO SAY...

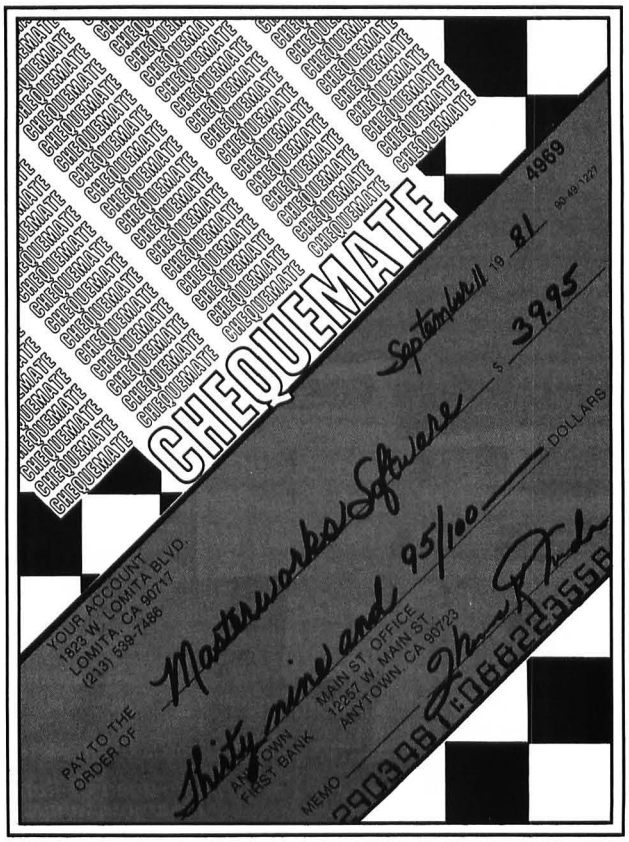
"At $39.95 ChequeMate™ is the Bargain of the Bunch."

Masterworks delivers more **value.** Our ChequeMate was reviewed in SOFTALK as the best bargain in home financial programs.

But as popular as ChequeMate is, Masterworks doesn't stop there. Our full line of home and business software programs offers you one of the most complete packages available. At the most attractive, "best selling" prices.

Here's a shopping list of our innovative programs:

ChequeMate
ChequeMate PLUS
(available for IBM PC & APPLE)
MasterDisk
MasterDOS
PAYMaster
ACCOUNTS Payable
ACCOUNTS Receivable



General LEDGER
ORDER Entry
FIXED Assets
BUSINESS Master

What's so great about Masterworks?

Plenty. Including the fact that with Masterworks Software you can get many of the features of large computer systems, previously unavailable for micros, for your APPLE II and IBM Personal Computer. This is made possible through Masterworks' unique programming system, exclusive to this complete software line.

You want value for your money — you want Masterworks software. For more information or the dealer nearest to you call (213) 539-7486.

Masterworks Software, Inc.

(213)539-7486.

25834 Narbonne Ave.
Lomita, California 90717

master charge
VISA

All disks have two sides.

But only Omni's Flip/Floppy lets you use both sides in single-sided drives.

Double your capacity.

Now you can **double** disk storage capacity as well as save on media costs if you have an Apple, TRS-80, Zenith, Northstar or **any** single-sided, 5¼" disk drive. Omni's Flip/Floppy lets you record on **both** sides . . . so you can consolidate long programs **and** get twice as much storage in one disk at far less cost than two conventional disks with the same capacity.

Premium quality, too.

Like our full line of flexible disks, each Flip/Floppy is critically certified at more than twice the error threshold of your system; rated for more than 12 million passes without disk-related errors or significant wear; precision fabricated with such standard features as reinforced hub rings; and backed by a full, 90-day money-back guarantee.

OMNI

Omni Resources
4 Oak Pond Avenue, Millbury, MA 01527

Dealer inquires invited.
Software houses:
call for duplicating services.

Plus a FREE storage case if you order now.

With **every** ten-pack you order (even conventional disks), you'll get a $5.00 protective plastic case, **absolutely free.**

Be sure to indicate system name and model # at right.

#of 10 Packs	5¼" Disks	Price Per 10 Pack	Total Price
_____	Flip-Floppy reversible Double side/double density (equivalent to 20 conventional disks)	$40.00	$_____
_____	Single side/single density	$30.00	$_____
_____	Single side/double density	$40.00	$_____
_____	Double side/double density	$45.00	$_____
	Shipping and handling		$ 2.00
	5% sales tax (Mass. only)		$_____
		Total	$_____

Call toll free (800) 343-7620
In Mass: (617) 799-0197

Call if you're not sure which disk is compatible with your system. Call for prices on 96 tpi and special formats. We offer an unconditional money-back warranty.

☐ Check ☐ Master Card
☐ COD ☐ VISA

Card #_____ Exp. _____

System and model # _____

Name _____

Address _____

_____

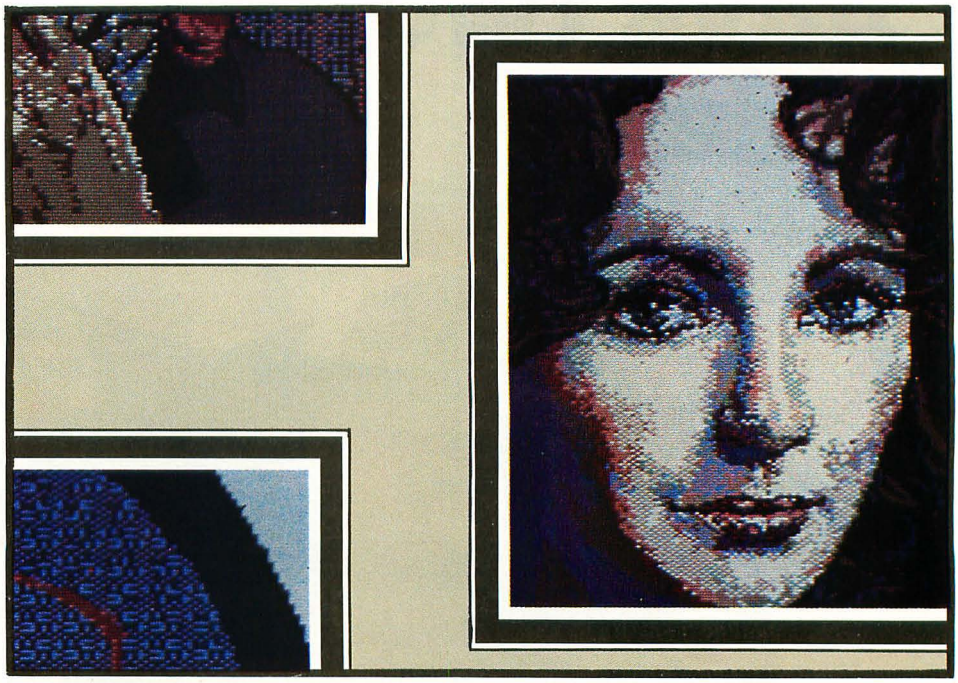
Tel. _____

SoftSide™

SoftSide Vol. 5, No. 12

SUBSCRIPTION INQUIRIES should be sent to *SoftSide* Publications, 100 Pine Street, Holmes, PA 19043. EDITORIAL AND ADVERTISING CORRESPONDENCE should be sent to *SoftSide* Publications, 6 South Street, Milford, NH 03055. Telephone (603) 673-0585.

SoftSide (ISSN 0274-8630) is published monthly by *SoftSide* Publications, Inc., 6 South Street, Milford, NH 03055. Printed at Lorell Press, Avon, MA. Second class postage paid at Milford, NH, Avon, MA, and at additional mailing offices. Subscription rates: US, and Canada, $30/12 issues. First Class US, APO, FPO, Mexico, $40/12 issues. Other foreign countries, $62/12 issues. Media subscription rates: US Magazine and Cassette, $75/12 issues. US Magazine and Disk $125/12 issues. APO, FPO, Canada and Mexico, add $20/12 issues. Other foreign add $50/12 issues. All remittances must be in U.S. funds. POSTMASTER: Please send form 3579 to *SoftSide* Publications, 100 Pine Street, Holmes, PA 19043.



COVER FEATURE

FEATURES

DEPARTMENTS

CONTENTS

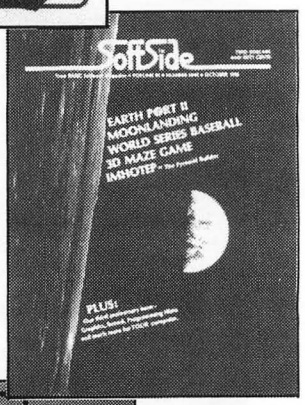
APPLE™/SIDE

TRS-80®/SIDE

ATARI®/SIDE

4th Anniversary

OCTOBER 1979



OCTOBER 1980

OCTOBER 1981

OCTOBER 1982

appleFest®

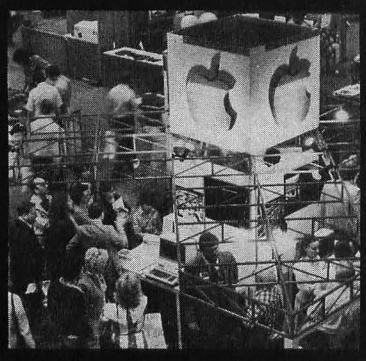
APPLE EUPHORIA

"Absolutely the best show I've ever been to in my life."

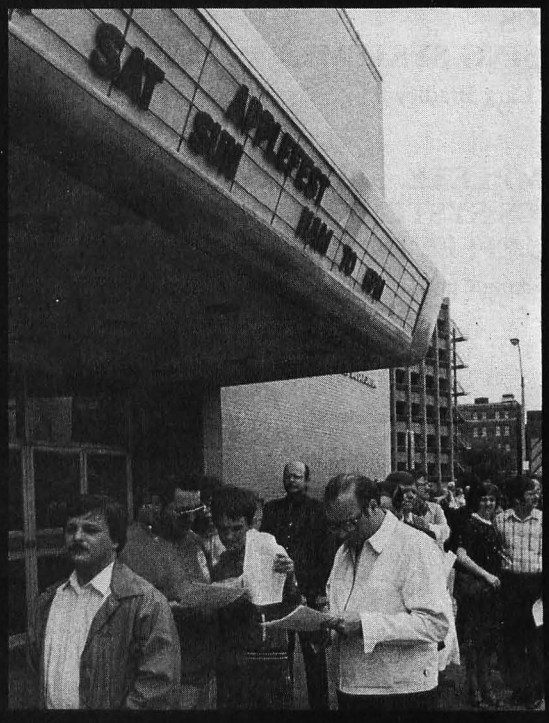
"It was incredible—I never knew there were so many products available for the Apple."

These are some of the things people are saying about Applefest, the world's largest exposition of Apples and Apple compatible products.

At Applefest, you can see and try out everything new for your Apple. Software for every conceivable application from arcade games to investment programs, music to machine language, teaching systems to accounting packages, word processors to graphics processors. Hundreds of peripherals, including printers, hard disks, modems, memory cards, video displays and synthesizers, plus accessories, publications, support services and more! Over 5,000 products are displayed and available for purchase at super show prices.



Get a taste of Apple Euphoria. Don't miss Applefest when it comes to a city near you.



Applefest/Minneapolis
Thursday-Sunday
September 16-19, 1982
Minneapolis Auditorium
Admission: Adult $5 per day, Child $3 per day
Show Hours: Thurs. 11 AM to 7 PM, Fri. and Sat. 11 AM to 7 PM, Sun. 12 NOON to 6 PM

Applefest/Houston
Thursday-Sunday
October 28-31, 1982
Houston Civic Center
Admission: Adult $5 per day, Child $3 per day
Show Hours: Thurs. 11 AM to 7 PM, Fri. and Sat. 11 AM to 7 PM, Sun. 12 NOON to 6 PM

Applefest/San Francisco
Thursday-Sunday
November 18-21, 1982
Brooks Hall
Admission: Adult $6 per day, Child $3 per day
Show Hours: Thurs. 11 AM to 7 PM, Fri. and Sat. 11 AM to 9 PM, Sun. 12 NOON to 6 PM

For information on exhibit space or tickets, call or write:
Northeast Expositions, 824 Boylston Street, Suite 202, Chestnut Hill, MA 02167 (617-739-2000)

EDITORIAL

Opening Night

by Randal L. Kottwitz

The audience is gathering outside the theatre. Many of them have had their tickets for months in anticipation of this night. They read in the trade magazines a year ago that a dynamic production of a new script was being prepared. The best stars had consented to play the leading roles and one of the true pioneers of the industry was to direct the production. Over the months, gossip continued to build the credibility of the initial reports, as rumor after rumor told of visitors to the rehearsals raving as they left the theatre. When the ads hit the newspapers, it was obvious that a hit was being born.

Beautiful art graces the marquee of the theatre and the posters for the play are already becoming "hot" items in the souvenir shops. Disappointed faces are leaving the ticket office, having been told that all seats have been sold out for the next three months. Finally, the lights are flashing in the lobby and the audience scrambles to their seats. Even the front curtain on the stage has been replaced by a startling painting representing the theme of the production. Obviously, no expense has been spared. The house lights dim, a hush settles over the audience and, at long last, the curtain rises.

Half an hour later, startled ushers watch as couple after couple dribble out the theatre doors. "How could it have happened?...Who could have guessed that such good people could have produced such schlock!...It's an interesting concept, but it's just not ready for an audience..." Such are the comments from an audience led down the rosy path of good marketing, lacking a good product to back it up.

This scenario, and many like it, has happened at theatres throughout the world. Regrettably, many a piece of software has followed the same path. Some of the greatest minds in the industry have combined to produce some of the worst software ever to hit the screen of a microcomputer. The trade magazines have raved about "secret" projects, destined to change the computing world, going on behind closed doors at major companies. Full color ads have graced the pages of magazines for months in advance, announcing the birth of revolutionary programs...the best thing since Mom and apple pie...only to have disappointed users throw up their arms in disgust when the software misses the mark. Padded binders and $300 plus price tags have carried products into featured positions in computer stores, only to have half of the packages returned and new sales screech to a halt as disenchanted users warn their friends.

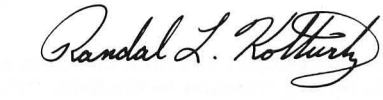
There are many reasons for these horror stories. In some cases, a product has been brought to market before it was truly ready. In others, developers have been allowed to create their dream package with no one sitting down and determining if it has a use in the "outside world." However, in many cases, good software has missed the mark because, once the user got inside the package, the program stopped "selling" him. No graphics or sound were utilized because it was a business package. Months were spent on the development of beautiful displays and complex music routines, but no attention was given to whether a game was actually interesting to play or not. In many cases, developers have spent too much time "blue skying" about what features should be included in a package, only to find that when time ran short on the promised delivery date, they'd bitten off more than they could chew. Rather than suffer the embarrassment of telling their customers that the product wasn't ready yet, they did what they could and delivered an inferior result.

The analogy between theatre and software is a very accurate one. No matter how complete and complex the development stages of any one production, they both serve one master — the final audience member/user. A theatrical production requires attention to every detail. I shudder when I think how many times I've seen a great and beautiful actress make a long awaited entrance on the stage, only to have the audience laugh because her cheek rouge resembled clown makeup or her dress made her walk like a duck. The same chill runs down my spine when I read the manual for a complex and exciting game, only to have the display come up on the screen looking like a novice programmer's *Hangman.*

There is no reason why a programmer should not pay attention to his audience and ask for their applause. But in order to do so, he must not only provide the tool to let them perform the desired task, but keep them interested, if not entertained, while they're doing it. At the same time, he must not get carried away with his presentation and ignore the meat of the matter — the ability of the program to perform its function. Above all, he must maintain a continuity of quality throughout his program. Many a play with a dynamite first act has left the audience with a sour taste in their mouths when they left the theatre after a boring second act. In most cases, this is because months of rehearsal were spent polishing the first act and only a few weeks rushed the second act into production. How many programs have you seen which got you excited in the initial phases of use, but left you disappointed when you got into some of the latter stages?

There are software packages on the market which pay definite attention to their audience — games that intrigue the mind as well as entertaining the eyes and ears — database managers and word processors which invite pleasurable hours spent in front of the screen. Indeed, the authors of many of these packages have found the extra features added by sound, color and graphics can add another level of information and clarity to their performance. These are the packages which will survive the longest and put their producers into the financial upper classes.

I spent several years working in educational and professional theatre before I came to the microcomputer industry. In those years I learned the definition of a "true professional." He is the creative genius who will first explore the potential of his role and stretch it to its ultimate, no matter how ridiculous it may seem. He then spends the remainder of his preparation time pulling the ridiculous back to the practical. Along the way, he has discovered all that his role has to offer and chosen only the best for the final performance to his audience. The software industry could learn a valuable lesson by applying the same principles.



Randal L. Kottwitz
Editor-in-Chief

INPUT/OUTPUT

From our readers

INPUT

DISKETTE LIFE

Dear *SoftSide,*

Your magazine is getting better all the time. Keep up the good work.

I like the three system format. It is great to be able to see other programs and programming languages. Another nice addition is the articles. I particularly enjoyed "The Big Crash" and "Video Mania," both from Issue 30. I appreciate "Hints & Enhancements," also.

I have a few complaints, however. In your DV and CV ads, it is stated that "all programs are tested and ready to run." Why, then, is there a "Bugs, Worms and Other Undesirables" column? If all programs were tested and ready to run there would be few, if any, bugs. If there is a good reason for the errors, how about telling us how they slip in. It's very annoying when you want to use a utility or play a game which you know is great (most of your programs are), but can't because of an error. Right now, I'm an amateur programmer and can't fix most errors.

My next complaint concerns disks. Almost every ***SoftSide*** disk I own is either going or gone and it's not my drives. A prime example is the October, 1981 disk, for which I waited four or five months. When I finally got it, it was almost lost. I barely saved it.

All in all, however, I think you have a great magazine. Keep up the good and keep out the bad.

Matthew Henry
Oak Forest, IL

Editor's Reply: To repeat an axiom which is becoming common in the field, "No piece of software is ever finished." As a result, there's always one more bug and, no matter how thoroughly we test a program, some will slip through. Regrettably, we are bound to a monthly cycle of preparing our software and the day comes when the material must go to our art department. We do the best we can and as soon as we (or you) find any bugs, we publish them as soon as possible. As for the life of our disks – we found, several months ago, that some of the disks we were using for duplication bore a lubricant coating which decreased the life of the disks for archival storage. Our new duplicator, Allenbach Industries, pointed up the problem and it has been corrected. We encourage our media subscribers to make a backup of every disk or cassette they receive in order to insure a long life for their software.

ADVENTURE SUBMISSIONS

Dear *SoftSide,*

I would like to know how I can submit an adventure program for 32K diskette. Also, how does an author submit an adventure for possible publication as an *Adventure of the Month*? For your information, I have found an error in the TRS-80® version of the *SWAT* program. The line that prints the heading should be TAB(16) and not TAB(17) so the underline will appear under the heading.

D.F. Chuang
Winchester, VA

Editor's Reply: We maintain strict memory requirements for our *Adventures of the Month* so that subscribers can be sure that each month's adventure will run on their system before they subscribe. (See the *Adventure of the Month* ad or bind-in card elsewhere in this issue for details.) We will, however, be glad to consider adventures which do not meet those requirements for publication elsewhere. Our most stringent requirement for adventure submissions is that you include a solution sheet in order to expedite our evaluation of the programs.

I SPEAK BASIC – REVISIONS

Dear *SoftSide,*

I am writing in response to an article which appeared in Issue 31 of ***SoftSide***. It was entitled "Why Johnny Can't Program – Materials for Computer Literacy" by Dean F. Hayden Macy. In the article, Mr. Macy mentioned that he would recommend, as a teaching manual, the book entitled *I Speak BASIC To My Apple* by Aubrey B. Jones, Jr., available from Hayden Publishing Company.

I recently had the opportunity to study this book to determine its suitability for programming classes, and I must strongly disagree with Mr. Macy's assessment. At first glance, it did look good. It even said it was field-tested, but I certainly don't know how it could have been. Many programs don't work at all, or at least not as indicated.

I found dozens of errors in the book. Although many of them are minor, there is really no such thing as a minor error in a program. This is especially true if the book is being used by persons with no computer experience, as the manual mentions it can be.

I have written to both the author and the publisher regarding this book. There is really no excuse for publishing a book containing as many mistakes as this one. I hope you will be able to inform your readers about it.

Thank you for your outstanding magazine. I look forward eagerly to every single issue.

Karen Grover
Rapid City, SD

Editor's Reply: A brief phone conversation with Michael Violano, Hayden Publishing's editor for *I Speak BASIC to My Apple*, revealed that they are aware of the errors in the book and are in the process of correcting 32 of them. They have also revised the chapter concerning the cassette recorder to speak more to disk drive systems, the more common configuration in the Apple™ marketplace. It's probably worth checking to make sure that you purchase the second edition.

continued on page 8

COMPUTER GRAPHICS

for the Artist

Special Effects

by David Lubar and Mark Pelczarski

Paint on your computer in over 100 colors with 96 different brushes! Magnify mode for precision touch-ups; mirror image; color reverses; move any part of picture anywhere; packing routine for increased disk storage of any standard graphics screen. Can be used by anyone. Paddle/joystick $39.95. Apple Graphics Tablet $69.95.

for the Designer

THE COMPLETE GRAPHICS SYSTEM II

by Mark Pelczarski

Everything needed for computer-aided design. Easily draw and design in two or three dimensions, mix text and graphics freely, define shapes and create typefaces with unique character generator. Great for presentations, videotape displays, storing designs on disks, graphics for other programs, or just for experimentation. Paddle/joystick $69.95, Apple Graphics Tablet $119.95. 50 additional fonts, $19.95.

for the Programmer

The Graphics Magician

by Mark Pelczarski, David Lubar, and Chris Jochumson

Add fast, smooth animation and hundreds of pictures to your programs. Create animated shapes and draw paths; have up to 32 independently controlled figures on screen. Use a special palette of over 100 colors to create pictures and objects in highly compressed format. Easy to use, and includes a special programming tutorial. Has saved months in development time for several games already on the market. Paddle/joystick $59.95, Apple Graphics Tablet $69.95.

penguin software 830 4th Avenue, Geneva, Illinois 60134 (312) 232-1984

All Penguin applications products are now on unprotected disks for your convenience. Apple II is a trademark of Apple Computer, Inc.

OUTPUT

by Randal L. Kottwitz

"Excuse me, would you mind playing on another computer so I can write "Output" for this month. I know it's the anniversary party, but we have to let our readers in on the celebration, too!"

'Tis the time to raise our glasses of cheer and toast you, the ***SoftSide*** readers, on this, our fourth anniversary! This is also the time for a special "Thank You" to the loyal readers who have been with us since October, 1978. You're all "family" to us and we want to share this special occasion with you.

You may have noticed the small ad in "Market/Side" last month, asking for program, article and review submissions for the IBM PC®. In the next issue, you will see material for the PC start to appear in our pages. We will be introducing our coverage gradually, in order to maintain the quality of information and presentation you've come to expect from ***SoftSide***. In a few months, however, "PC/Side" will be a full-blown section as we welcome the PC into the ***SoftSide*** fold. Regrettably, I must also tell you that we have postponed the introduction of the Radio Shack® Color Computer indefinitely. To be blunt, our decision favoring the PC over the "CoCo" is based on our evaluation of the past and future sales of the machines. Certainly, Radio Shack's recent decision to market the Color Computer through independent distributors under the name *TDP System 100* may force us to reevaluate our decision in the near future.

Now for the bad news and a little egg on our face. We proudly announced in this column and in "Coming Next Issue" last month that the *Envyrn™ Tile Editor* would be published in this issue for all three of the systems we support. Unfortunately, we discovered, after issue 32 had gone to press, that there were some unanticipated complications that make it impossible. As a brief explanation, we ran into memory allocation problems with the Apple™ hi-res screen and must make our initial implementation in the text mode. Ah! What we wouldn't give for a "soft" character set on the Apple™ — a definite hope for the "Super II." The ATARI® presented yet another problem. We are implementing its version of *Envyrn* under Microsoft BASIC and have discovered many complications yet to be worked out, i.e. misprints in the manual and bugs or misunderstandings in the implementation of the language. In addition, our IBM® machines are starting to arrive and we will easily be able to implement *Envyrn* on that computer as well. The end result — no *Envyrn* in this issue of ***SoftSide***. We've decided not to publish the TRS-80® version at this time, either, as much of the supporting material — articles, instructions and illustrations — would have to be repeated when the translations were ready. The net result would be repeated pages of identical material in ***SoftSide*** and we would rather give you more information on other topics than have to repeat ourselves and waste pages. We've not given up on the project in any way and will be bringing you *Envyrn* as soon as possible.

Our customer service department has had its phone ringing off the hook as some of you have received your issues a bit later than usual. We're quite flattered that many of you so strongly look forward to your next issue that you quickly notice its absence. However, we must offer an explanation so the customer service department can get back to their regular duties.

At the time we changed from dated to numbered issues, we altered our production schedule in order to make up some lost time. Major projects such as our Word Processing Issue and C.A.T.S. were each taking a little piece out of the next issue's alotted preparation time, compounding into our being behind schedule by almost a month. We will be making up for the delay in your issue number 31 gradually, and you may rest assured that you will still receive 12 issues for the price of your subscription.

That's all for this issue. We look forward to our fifth anniversary, knowing that the intervening year will bring yet another bundle of surprise developments in the microcomputer industry. We'll do our best to take some of the mystery out of those surprises. Happy Hacking!

"All right, all right! You can have the machine back. How did I ever get involved with all these game freaks?

"I challenge you to a doubles match of..."

HINTS & ENHANCEMENTS

From our readers

APPLE™ *MICROTEXT* IN-LINE EDITING

An inconvenience of *Microtext* (April, 1982) is that when you are editing a line, you must re-type the entire line. I thought editing would be less cumbersome if you could move quickly to the error in the line being edited. Such a feature can be added to *Microtext* with one line:

```
504 IF C=21 THEN C$=MID$(L$(EL),
CHAR,1):GOTO 740
```

With this change, you enter the edit mode, and select the line to be edited, just as before. Now, however, the right-arrow key skips over text that is to remain unchanged. Pressing the right-arrow key along with the REPT key causes this to occur faster.

Randy Rogel
San Diego, CA

APPLE™ *MICROTEXT* ADVANCED FEATURES

Here is an enhancement to *Microtext* (April, 1982) that prevents the need to print an entire file, should the format specifications be set up incorrectly.

```
7058 PRINT : PRINT "ENTER S TO S
    TART, Q TO QUIT?";: GET X$: IF
    X$ = "S" THEN 7060
7059 GOTO 7150
7115 X = PEEK ( - 16384): IF X =
     209 THEN 7150
45  LX = 0:PL = 49
7000  HOME : VTAB 6
7005  INPUT "RESET PAGE SIZE?(DEF
     AULT=50)";X$: IF X$ < > "Y"
      THEN 7010
7008 LX = 0: INPUT "LINES PER PAG
     E?";X$:PL = VAL (X$) - 1
7010  PRINT : INPUT "LEFT MARGIN?
     (DEFAULT=10) ";X$:LM =  VAL
     (X$): IF LM < 1 THEN LM = 10
7070  HOME :P$ = "":CR = 0:I = 0
7130  PRINT TAB( LM);: IF UC THEN
     PRINT P$;: GOTO 7145
7145 LX = LX + 1: IF LS > 1 THEN
      FOR J = 2 TO LS:LX = LX + 1
     : PRINT " ": NEXT J
7147  IF LX > PL THEN FOR J = 1 TO
     66 - LX: PRINT : NEXT J:LX = 0
7150  PR# 0
7610  FOR J = 1 TO LS:LX = LX + 1
     : PRINT " ": NEXT J
7615  IF LX > PL THEN  FOR J = 1 TO
     66 - LX: PRINT : NEXT J:LX = 0
```

Ken Leifheit
Elburn, IL

This enhancement provides control of the line counter and page size, supplying the ability to concatenate files while maintaining control over margins and spacing for each file. With this feature, one may break a report into several segments for entry, and then print the segments in the desired sequence.

This enhancement is not compatible with the cassette-version fix published in the May issue.

ATARI® COLD START SOFTWARE SWITCH

In recent articles involving the use of *S.W.A.T.* (issue 30), ***SoftSide*** has instructed ATARI® users to turn their systems off and then back on in order to clear their variable tables. This practice is, for the most part, unnecessary, and may, over the long haul, be bad for the hardware.

There is a way to "turn the ATARI® off" without actually turning the power off. Simply type "POKE 580,1", press the RETURN key, and then press the SYSTEM RESET key. What this causes is a legitimate "cold start" without interrupting the power.

Although this example uses the direct mode, keep in mind that it can also be incorporated within the body of a program.

Paul Pettennude
Ramsey, NJ

TRS-80® *SOLITAIRE* SCORING

You have again hooked the students of Handley High on one of your programs, *Solitaire* (May, 1982). The only complaint from the students is that the program lacks a scoring routine to tell them how many of the 52 cards they've used. The following enhancement supplies this feature:

```
1433 FORQ=0TO6:Q7=Q7+IN(Q):NEXTQ
1434 FORQ=1TO4:Q7=Q7+F(Q):NEXTQ
1435 PRINT@768,CHR$(31):PRINT@832,"You w
ere able to play";Q7;"cards.":PRINT@896,
" ";
1436 IFQ7>45THENPRINT"You almost made it
!!!!"
1437 IFQ7>30ANDQ7<46THENPRINT"Excellent
work !!"
1438 IFQ7>20ANDQ7<31THENPRINT"That is on
ly an average game."
1439 IFQ7<21THENPRINT"Are you sure you k
now how to play this game?"
1440 PRINT@960,"Care to play again? (Y/N
)";:GOTO1395
```

David Pleacher
Stephens City, VA

TRS-80® *MICROTEXT* PRINTER ENHANCEMENTS

Here are three modifications for *Microtext* (April, 1982).

The first thing I wanted was a TAB key. Line 510 below does this. If the right-arrow

Translation of the Month Contest

You could win a FREE *SoftSide* DV or CV Subscription!

SoftSide's* Translation of the Month** has been so well received by our readers, we're offering a greater author incentive than ever before. No, we can't give you a job at the U.N., but we will award a one-year subscription to ***SoftSide DV or an 18-month subscription to ***SoftSide CV*** for a high-quality translation of one of our past programs. That's a value of $125 for the ***Disk Version*** or $112.50 for the ***Cassette Version*** — you'll be rewarded every month for your translation efforts!

Here are some of the most important qualifications we look for in a translation winner.

Your entry must be a translation of one of the featured programs from a past issue of ***SoftSide***. (We're particularly interested in Apple™ and ATARI® translations of some of our older TRS-80® only issues. Write for a list of suggested candidates.) In general, we're looking for translations of programs which are a CHALLENGE to translate. Some of the programs we publish are written in more or less "generic" BASIC, which can be typed into another computer with very few changes. Although these programs require the least effort to translate, they are also the least likely candidates for contest winners.

Your translation should be thoroughly tested and completely bug-free. Just converting program lines doesn't automatically ensure a workable translation. Be sure to use-test your translation as carefully as you would test a program you had written entirely from scratch.

Your translation should fully utilize the unique features of the computer for which it is written. The objective of a translation is to "fit" the capability and convention of its host computer, not simply mechanically duplicate the operation of the original program. This is especially true of programs which use graphics, and should be kept in mind for such minor features as keyboard layout (use of such special keys as arrows, ESC, CTRL, CLEAR, etc.). Also be careful with screen formatting; a word that spills over into the next line because of a PRINT statement that wasn't properly rewritten betrays such carelessness that we'll probably reject your translation automatically.

Your entry should incorporate any improvements and enhancements you can add to the original program. Don't feel that you have to limit yourself to the boundaries of the original. (On the other hand, don't go overboard and destroy the character of the original by completely rewriting it!) An enhanced translation is much more likely to catch our attention than a line-for-line duplicate, and it will have more value to our readers.

It's not necessary to include extensive documentation with your translation, only that which is different from the original. If most of the originally published documentation applies to your translation, simply say so. You should, however, include descriptions and explanations of any changes or enhancements you've made.

All **Translation Contest** entries must be submitted on disk or tape, with documentation in printed or typed form. Media will be returned only if accompanied by a self-addressed, stamped envelope. Send your entries to:

SoftSide

Translation Contest
6 South Street, Milford, NH 03055

key (unshifted) is pressed, five spaces are inserted by placing them in C$; also, variable CHAR is updated. Note that doing a TAB anywhere but the beginning of a line may cause justification problems.

```
510 IFC$=CHR$(9)THENCHAR=CHAR+4:C$=
STRING$(3,32) 'TAB
540 IFC$=CHR$(10)THENC$=CHR$(92)
'NEW PAGE
550 IFC$=CHR$(25)THENC$=CHR$(94)
 'DOUBLE-WIDTH
```

Next, I wanted a way to signal my Line Printer VII to start a new page or use its double-width feature. The unshifted down-arrow key starts a new page, and the shifted right-arrow toggles between double-width and single-width print. Lines 540 and 550 insert these characters.

The following routine catches the new control characters:

```
7085 GOSUB 7700 'PATCH IN NEW
 SUBROUTINE
7700 'INSERT CONTROL CHARACTERS
7710 DW=-1:LC=-LEN(P$):FORZ=1TO-LC
7720 IFMID$(P$,Z,1)=CHR$(92)THENMID$
(P$,Z,1)=CHR$(11)
7730 IFMID$(P$,Z,1)=CHR$(94)THENMID$
(P$,Z,1)=CHR$(30-DW):DW=NOT
DW:LC=LC-1
7735 LC=LC+1:IFDW=0THENLC=LC+1
7740 NEXTZ:P$=P$+STRING$(LC,0)
7750 RETURN
```

LINE 7720 replaces the down arrow with a form-feed character; line 7730 inserts the appropriate character to toggle between single (CHR$(30)) and double (CHR$(31)) width. You should change these characters to the ones your printer uses. Line 7735 adjusts the character counter, LC. When everything is converted, P$ is padded with null characters to compensate for the extra line length caused by double-width printing. Level II users should replace the phrase "MID$(P$,Z,1) = CHR$(nn)" with "P$ = LEFT$(P$,Z-1) + CHR$(nn) + MID$(P$,Z + 1,255)" in lines 7720 and 7730.

Three final changes enable the program to recognize the form-feed character as a line terminator:

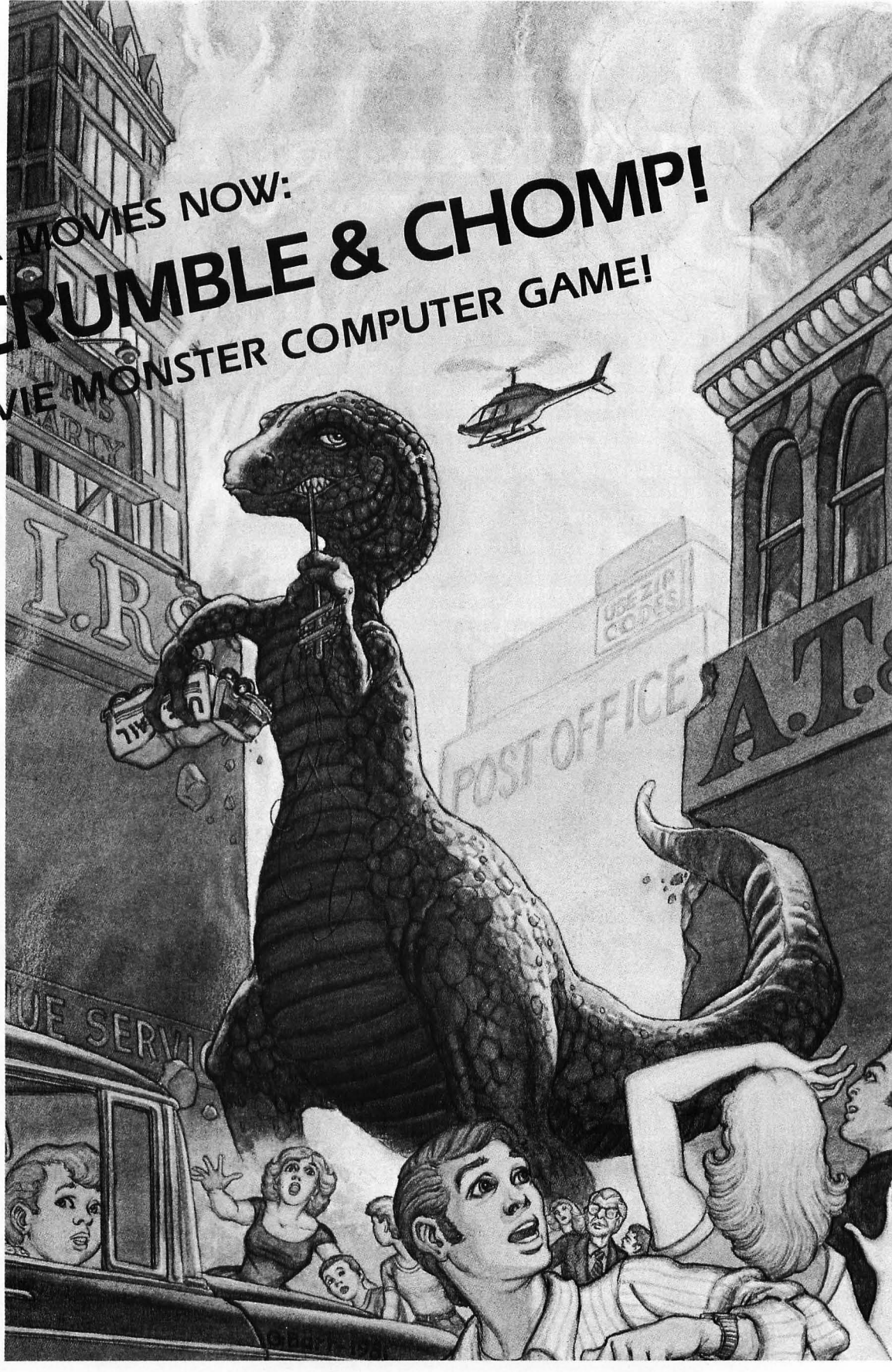
```
780 IFC=RTNORC=92THEN...
7090 IF RIGHT$(P$,1)=CR$ORRIGHT$(P$,1)
=CHR$(92)THEN...
8040 SAME CHANGES AS IN 7090
```

Joe Sewell
Melbourne, FL

FIRST: MONSTER MOVIES NOW:

CRUSH, CRUMBLE & CHOMP!

THE GREAT MOVIE MONSTER COMPUTER GAME!



And guess who stars as the movie monster. You! As any of six different monsters. More if you have the disk version.

You can terrorize and destroy four of the world's largest and most densely populated cities in over 100 possible scenarios. From Tokyo to the Golden Gate, you are the deadliest creature in the air, on the land, or in the sea.

You can be the deadly amphibian who simultaneously smashes street cars, lunches on helpless humans and radiates a ray of death.

If you were a giant winged creature, think of the aerial attacks you could make on the terrified but tasty tidbits beneath you.

But as in all the best monster movies, you're up against everything the human race can throw at you—even nuclear warheads and a strange concoction developed by a team of mad scientists.

For only $29.95 you get 6 stupendous monsters, each with its own monstrous summary card, 4 teeming metropoli displayed in graphic detail on your computer display and mapped in the accompanying 48-page illustrated book, the awesome sounds of monsterly mayhem, and spine-tingling, real-time, edge-of-your-seat excitement.

GET CRUSH, CRUMBLE & CHOMP
now at your local dealer for your APPLE, ATARI,
or TRS-80 . . **before it's too late.**

COMPUTER GAMES THINKERS PLAY
EDYX

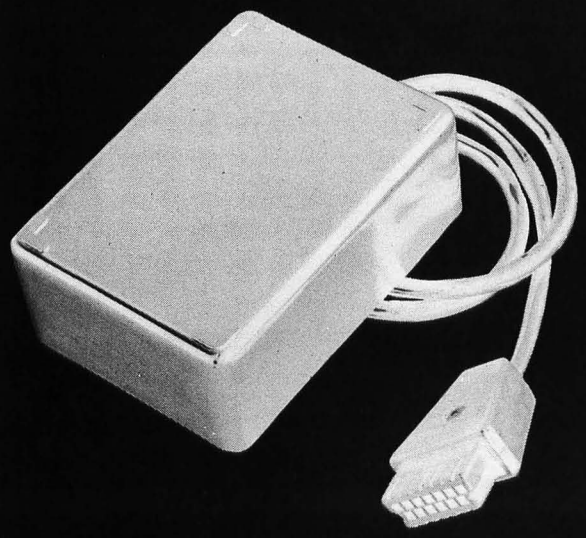
ATARI SAYS ITS FIRST WORD

WITH A VOICE BOX BY THE ALIEN GROUP!

THE ALIEN GROUP has emerged from the underground, daring to offer a full-featured speech synthesis system that is flexible, low in cost, and needs no-accessory devices. No Interface, Cables, Speaker, Amp, or External Power Required! **VOICE BOX** has been designed and programmed by Atari users to become the integral voice of a *400* or *800* computer. Simply plugged into the serial port, **VOICE BOX** automatically routes all speech into the speaker of your television monitor. With the menu-driven operating system supplied, you'll be creating original, intelligible speech within moments after loading disk or cassette. No lengthy or obscure instructions to wade through.

The system includes a dictionary which translates typed text into **VOICE BOX's** phonetic language. The dictionary can be expanded to include as many as *5,000* words of your own custom vocabulary. Unlimited speech can be produced by straightforward phonetic definiton at any time, even if the dictionary should be full.

The speech routines can be called from other programs for any purpose you can conceive. Here are a few suggested uses:



VOICE BOX
Speech Synthesizer

SOUND EFFECTS
- Access 64 phonemes at any of 4 pitch levels to add filtered, contoured sound to the Atari audio repertoire.

GAMES
- Program aliens to hiss threats, moan when destroyed.
- Devise weird, non-human tongues for dungeon dwellers.
- Insert cryptic spoken clues in maze games.

COMPUTER OPERATION
- Code verbal prompts and error messages that command attention and leave the current display intact.

EDUCATION
- Gain an introduction to the principles of phonetics.
- Learn touch typing through spoken feedback from the keyboard.

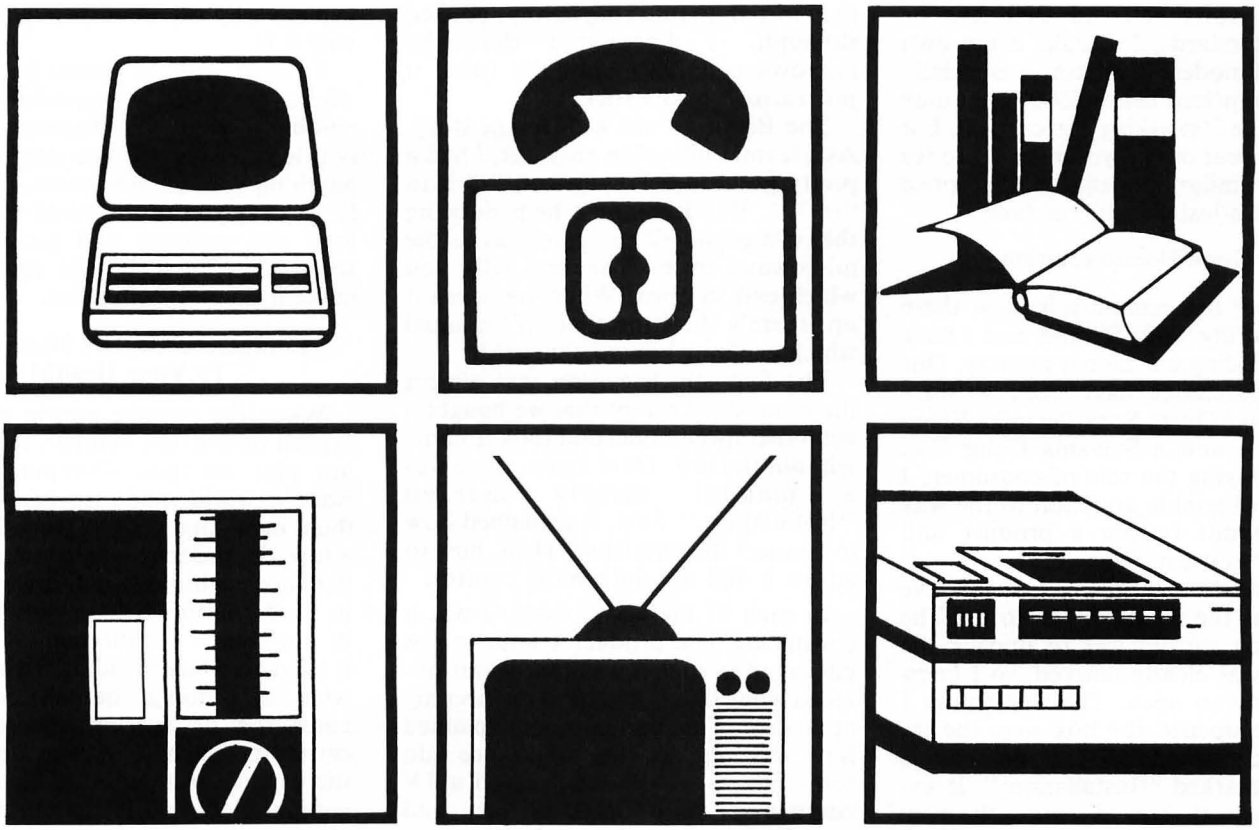
In addition, the *Random Sentence Generator* included in the operating system, which prints and speaks endlessly startling, amusing, even poetic combinations of words supplied by the user, helps teach school children to identify parts of speech and recognize a variety of sentence structures.

A minimum of 16K RAM is required by the operating system. Either disk or cassette includes both 16K and 32K versions. Try **VOICE BOX** for up to 10 days, and if it isn't the finest value you've ever seen in a computer peripheral, the most challenging and provocative addition you've ever made to your system, return it in its original condition for a full refund.

When ordering specify disk- or cassette-based operating system, and enclose check or money-order for $169, or state your **VISA** or **MASTERCARD** number. Send mail orders to: **THE ALIEN GROUP**
27 West 23rd Street
Dept. Dept. SS3
New York, New York 10010

or telephone orders from
10 AM to 6 PM New York time
(212) 924-5546

ALSO AVAILABLE AT LEADING COMPUTER STORES THROUGHOUT THE WORLD.



By Lance Micklus

When I did the review of currently popular modems (see ***SoftSide***, April, 1982), I did not include the *Hayes Smart Modem*. One of the problems with the *Hayes* is that they are so popular, they're sometimes hard to get. After making a lot of phone calls, I have finally managed to get one, however. As much as I hate smart modems, the *Hayes* is very hard to fault. This modem has many features. It's impossible to duplicate what the *Hayes* does with handshake signals alone.

One feature I particularly like is the ability to dial on a touch tone or a pulse dialed phone. The second feature that impressed me was a built-in speaker. It lets you listen to the phone line when you dial so you can hear the connection go through, but automatically goes silent when the connection is established. That one feature clearly proves that the designer had used his creation and realized what a problem it is to dial deaf. I also like the indicator lights. It's nice to see what's going on.

With the *Hayes*, you have total control over all its functions. You can program the number of rings to wait before answering the phone if you're in auto-answer mode. You can have it dial the phone in ANSWER mode instead of ORIGINATE. You can even use it as a computer driven phone dialer. All of this may sound complicated, but if you use the default settings, you can forget about all this other stuff until you're ready to deal with it. That's my kind of design.

Reports from many *Hayes* users have indicated that it is very reliable, but suffers from one major problem – the handshake signals. All of the major host software for TRS-80® computers uses pin 5, CLEAR TO SEND, to determine when the modem has a caller on the line and when the caller has hung up the phone. Unfortunately, *Hayes* uses pin 6 for this function and leaves pin 5 on all the time. This, of course, makes the host program think that it always has a caller on the line. Both SBSG and I have had many calls from both FORUM-80 and MOUSE-NET™ operators who have run into this problem with the *Hayes*. The solution is simple enough. Just remove the wire from pin 5 and connect pins 5 and 6. The wire that used to go to pin 5 doesn't connect to anything.

The second problem is also a handshake problem regarding pin 6 (also pin 5, if you make my suggested change). When you give the *Hayes* a phone number and backspace to correct a typing error, it momentarily turns pin 6 off and then on again. This creates a minor problem with *ST80-III 2.50*, which displays any change in the handshake signals.

On the plus side for *ST80-III 2.50* users, you can can load your 10 User-TEC commands with 10 phone numbers for automatic dialing. Better still, I have a *Hayes* auto-dialer for *ST80-III 2.50* on my system if you want to use the built-in dialing features of *ST80-III*. Just call my board at (802)862-7023 and look for HAYES/DLR in the data base.

I would recommend the *Hayes* to anybody. Still, I'd rather dial the phone myself and use the *Anderson Jacobson AJ-1258*. I really love that 1200 baud. If *Hayes* could get straightened out on the handshake signals, and could come up with a 300/1200 baud version that

automatically selected Bell 212 or Vadic standards, I'd make it my own personal modem in a flash – especially if it were priced below $500. That may sound like I'm asking a great deal, but within a year or two you're going to see several similar modems in that price range. It's just a matter of time.

Good Documentation

I know the economy is slow these days, but my wife, Dianne, and I have been spending our money anyway. Our recent purchases have been a Sears Microwave Oven, Sony Betamax Video Recorder, and a Sylvania Color TV. While playing the role of consumer, I paid considerable attention to the way I felt about buying a product and learning to use it.

Of the three items, the microwave oven was the most foreign to us. The first thing we did was to unpack the box. It was clearly marked, so I knew which end to open. The first things I saw as I opened the box were the instruction manuals. One manual is clearly marked "Installation." It explained exactly how to remove the oven from the box, connect it, and install the racks inside.

Next, Sears made a very smart move. They knew that people like us would immediately want to make the oven cook something. After the instruction regarding the installation of the oven, the manual tells you how to boil a cup of water to make instant coffee. The steps involved explain how to set the temperature and timer. One minute later, you have hot water.

Sears obviously knew that, at this point, the consumer would be ready to sit down and do a little reading. To provide us with hours of entertainment, a very large microwave cookbook was included. First, it described some of the things we'd be able to cook, and then gave us a few, simple things we could cook immediately. The fast cup of instant soup used the temperature probe. We also tried a hot dog and a potato. Then, it told us how to check our dishes to see if they were microwave proof. Sadly, half of our dishes failed the test. Finally, the book is divided up into sections containing hundreds of other recipes.

Like most people, we had to go through that three week period of microwaving everything to see how it all worked. I made several attempts at eggs. I got them to look beautiful, but they tasted awful. They always cooked through. I discovered that the microwave doesn't burn the food, it just turns it into a rock.

The Betamax was a different story. As a former television engineer, I had a pretty good idea how to connect it to the TV. But I couldn't help noticing that it was packed the same way as the microwave oven. The box tells you which end to open. When you open it up, there's the "Installation" manual which explains how to connect it.

The Sylvania television was almost the same story except that we bought a demonstration model and took it home without the box. Once again, there was a manual clearly marked "Installation." First, it explained how to connect the television. Then, how to adjust it and use the remote control.

In each of the above cases, I was a consumer. The product I bought was packaged in such a way that it was obvious what I had to read to connect my appliance. The manual then explained how to make the appliance do something – boil water, record a TV commercial, and make pictures and sound come out of the television set. Finally, each manual explained how to make that appliance operate more efficiently.

A couple of weeks ago, I got a new piece of CP/M software. I consider myself to be modestly intelligent on matters concerned with computers, but it took me over half an hour to figure out how to load the program.

The manual began by explaining how to use some of the advanced features of the product. After looking through the manual a dozen times, I found, on page 40, an explanation of how to read the manual and get the program to run. Page 40? You're kidding. How about putting this information at the beginning of the manual so I know how to get started?

The software I am referring to is very well known and always gets high marks on the quality of its documentation. My reaction is this: If Sears, Sony, or Sylvania ever put out an instruction manual like that, people would get killed trying to learn how to use the product and Ralph Nader would be climbing all over the manufacturers.

I'm not saying that I'm the world's greatest documentation writer, but if any of you have seen the new *ST80-III*® documentation, the above stories explain why the manual is written the way it is.

To start with, it should be obvious where the user is supposed to start reading – page A1. The manual clearly titles each step so you can skip over anything you already know. Ultimately, it leads you to the point where you load the program and get it to do something. Then, it tells you how to make it do everything else.

Computers May Be Hazardous To Your Health!

When the average person reads the typical instruction manual, the following plot emerges: Computer people want a world made up of people of their own kind. We're superior, you know. In order to achieve our objective, instruction manuals are written to be read and understood only by those in our ranks. To the computer idiot, it's Greek. Since the idiot doesn't know what he's doing, he will inevitably enter the destruction sequence that causes the machine to blow up and kill the jerk. Over a period of time, all the computer idiots of the world will be eliminated, leaving only the vastly superior race of computer freaks.

To the average man on the street, this is not as far-fetched as it sounds to us. Many people could easily be convinced that it is true. They've seen computers killing people on TV for years. After looking at the typical instruction manual, it seems as though the author wants people to enter the wrong sequence so the computer will blow up.

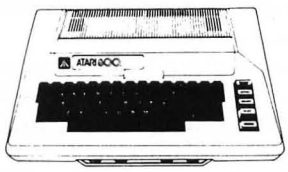
It's interesting that people don't have the same fear of TV sets, Betamax recorders, or microwave ovens. When was the last time you saw James Bond destroy a SPECTRA hideout by overloading their microwave oven? None of these devices seem threatening. When you read the instruction manuals, you even get excited about using the product.

Even the names we use for our computer programs are threatening. Joe Simon is the one who first pointed out to me how user unfriendly we computer people are. At the time, I thought it rather strange that the author of *Bullet-80* would tell me that my *ST80-CC* wasn't as user friendly as it could be. *Bullet-80* sounds like some kind of deadly weapon. If a woman came running down the street screaming that a

continued on page 16

A.S.C., INC
Authorized Service Center

"Call us for discount prices, availability, or advice. Remember, service is our middle name." Joe Guzzo, President, A.S.C. Inc.



ATARI 800

800 16K $654.95
800 32K $745.95
800 48K $815.95

ATARI 400

400–16K $264.95
400-32K $364.95
400-48K $430.95

410 Recorder $76.00
810 Disk Drive $444.95

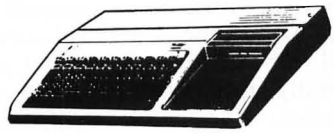
Kits

481 Entertainer $83.00
482 Educator $129.95
483 Programmer $56.95
484 Communicator . . . $339.95

Software

We carry the full line of Atari hardware and software products backed by our Authorized Service Department.

Please call for special prices on all Atari software and APX software.



TEXAS INSTRUMENTS

All models . . . . . Call for prices

MONITORS

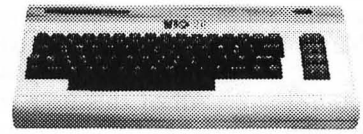
Amdex 12-in. B&W . . . $129.00
Amdex 12-in. Green . . $139.00
Amdex 13-in. Color . . . $349.00
BMC--BM 12A $114.95
NEC 12-in. B&W $169.95
NEC 12-in. Color $339.00
TI 10 . . . . . $344.95
Corvus . . . . . Call for prices
Zenith Z19 12-in. $115.00

DISKS

Elephant . . . . . 10/$21.95
Elephant DD . . . . . 10/$24.95
Dysan 5¼-10/16 . . . . . 10/$38
Verbatim 8-in.SD--SS . . . 10/$32
Verbatim . . . . . 100/$245
Scotch 3M . . . . . 10/$25.95

DISK DRIVES

Corvus . . . . . Call
Percom . . . . . Call
Paradynamics . . . . . Call



VIC-20

VIC-20 Personal Computer $175

GAMES/ PROGRAMS

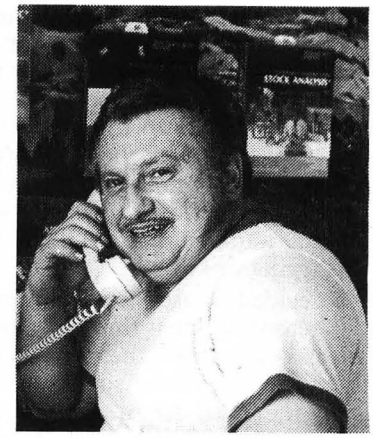
Unlimited Availability . . . . . Call

COMPUTER DESKS

Send for catalog and photos.

DC-2 ROBOT

Operates on any home computer.
Stands 4 feet high.
Call for details and prices.



TERMINALS

Televideo $569.00
910 . . . . . $569.00
912C $694.95
920C $744.95
950 . . . . . $934.95

Call for Computers

MODEMS

Lexicon $149.00
U.D.S. 1030ALP $169.00
Racal Vadic VA 3212 . . $749.00
Prentice Star $179.95

PRINTERS

Brother HR-1 Parallel . . $829.95
Brother HR-1 Serial . . . $880.00
Smith-Corona TP-1 Parallel $665
Smith-Corona TP-1 Serial . $665
C. Itoh Starwriter $1,350
Diablo 630 . . . . . $1,999.95
Epson printers . . . . . Call
Centronics printers . . . . . Call
All name brands . . . . . Call

NEW•NEW

Coleco-vision, Mattel Intellivision, and Atari VCS game machines and game cartridges. Call us for information on the latest, improved models and best prices.

Limited space prevents a full listing of the lines we carry. If you don't see it listed, we can get it for you, for less.

TO ORDER: Phone orders accepted via Mastercard, Visa, or bank wire transfer. Visa and MC service charge of 2%. Mail orders may send charge card number (include expiration date), cashiers check, money order, or personal check (allow 10 business days for personal or company checks to clear). Please add 3% ($5.00 minimum) for UPS shipping, handling, and insurance. APO and FPO include 5% ($7.00 minimum) for postage. California residents add 6% sales tax. Please include phone number on all orders. FOREIGN ORDERS include 1% handling. Credit cards not accepted on foreign orders. All equipment is in factory cartons with manufacturer warranty. Opened products not returnable. Restocking fee for returned merchandise. Equipment subject to price change and availability. Retail prices differ from mail order prices. WE SHIP THE SAME DAY ON MOST ORDERS.

Call Collect (714) 284-5615 7 Days/Week

Monday through Saturday 8am–8pm
Sundays and Holidays 12n–6pm

After 11/6/82 , Call (619) 284-5615

A.S.C. Inc., 7436 University Ave.
La Mesa, CA 92041

man with a *Bullet-80* was trying to kill her, I'd call the police. If the same woman came running down the street screaming that a man with an *ST80-CC* was trying to kill her, I'd doubt her sanity. But I took Joe's advice and selected a very unthreatening name for my next bulletin board: *MOUSE-NET*™. Surely no woman would ever come running down the street screaming that a man with a *MOUSE-NET*™ was trying to kill her.

I decided to use the word *Mouse* for two reasons. One is that I like mice and am associated with them. I also wanted to convey the idea of smallness without using the word *Micro*. Everybody uses *Micro*. There's MicroNet, Microsoft, MicroPro, MicroWorks, MicroLab, and the MicroConnection. I also did not want to use a threatening name. Who's going to be afraid of a mouse?

Now, take *Word Master*, for example. That makes me feel inferior. Scripsit leaves me cold. *Electric Pencil* sounds familiar and friendly. My vote, though, goes to *Lazy Writer*. That name implies all of the right things. When you run some of these programs, look at what you see: "ENTER SOCIAL SECURITY NUMBER (000-00-0000)?" Doesn't anybody ever say "PLEASE?" Then there's the question mark – it's like the computer isn't even sure it wants to know. Of course, some programs use an arrow which is a bossy way of saying, "Put the data right here." Nobody likes to work with people who are that way. Remember the movie *9 To 5*? We all hated the boss, even when he wasn't putting women down. He was a dictator. Somehow, when a machine is a dictator, it's supposed to be okay.

All of this user friendly stuff is fine except for one problem. The more highly paid the consumer, the more impressed they are by complicated, technical documentation, sophisticated names, and the implied threat of assault on one's life. It's these people who are forcing programmers to continue our old, threatening ways.

Imagine that you're the head of a large data processing department and you want to buy a terminal program for your TRS-80®. Do you tell your superiors that you want to buy a *UNITERM*? That sounds like the name of an animal with only one tail. How about an *OMNITERM*? That sounds like an android from some space movie. *VIDEOTEXT* is, of course, the name of a quiz show on TV at 7:30 on Saturday night. But an *ST80*...well, that sounds complicated. It sounds like the guy who wrote that must have written hundreds and hundreds of terminal programs and uses a code to keep track of the versions.

It's a dilemma. If you make things user friendly, which is what people want, their bosses won't be impressed enough to buy the program. The only solution is to fool the boss.

All programs which have a practical application will have two names. One will be extremely complicated, to impress the boss. The other name will be user friendly. If you order an *ST80-MN*, you will receive a *MOUSE-NET* ™. That way, you can put *ST80-MN* on the purchase order and impress all of your superiors. Once the people in your department get the software, they will see *MOUSE-NET* ™on the screen so they'll feel more comfortable.

Of course, *ST80-MN* will cost $50 more than *MOUSE-NET* ™. But for $50 you'll get two instruction manuals. The first one will be called a Training Guide. That's the one you'll read to figure out how to use *MOUSE-NET* ™. The other book is called the Instruction Manual. That's the one you show to your boss when he wants to see what the company bought. It's written in very technical language, just like the CP/M manuals are. I'm even going to include some useless source code listings that nobody will ever figure out.

If the boss should ever ask why the *ST80-MN* program comes up on the screen as *MOUSE-NET* ™, just tell him it was your idea. Show him the picture of the mouse in my logo and say that you decided to change the name to see if the patch worked. Tell the boss that the ability to change the name of the program is just one of its interesting features.

Open the Instruction Manual to page 132. Don't worry about what's on the page; it doesn't matter. Point to the middle of the page and tell your boss it's all explained starting there. Your boss will be so impressed with the fact that you figured out how to change the name of the program by reading what he thinks is the Instruction Manual, he'll give you a raise.

Now, that's what I call a marketing strategy. I get the business, you get the raise, and the boss gets a worthless manual that he paid extra for and with which he couldn't be happier.

"IT'S CALLED VISI-DUMP! IT CAUSES THE SYSTEM TO CRASH AT THE WORST POSSIBLE TIME."

A Fast, Easy, Inexpensive Approach to Word Processing

NEW

PIE Writer:
Word Processing System

Hayden is thinking about your future. We know you're concerned with saving time and money . . . with increasing efficiency and productivity . . . and that is the reason for PIE Writer!

Now you can turn your Apple II into a complete word processing system. PIE Writer, a powerful editor and formatter, offers the versatility and machine efficiency of the larger systems—at a fraction of the cost!

Now all your documents and correspondence—reports, memos, direct mail pieces, even "personalized" form letters—can be composed, edited, formatted, and printed quickly and accurately. Simple keystroke commands let you alter characters, words, sentences, even entire paragraphs!

Learning to use PIE Writer is as easy as learning to type. There are many user-oriented features, designed for first-time users as well as serious programmers: an easy-to-understand, indexed instruction manual, a carefully designed reference card, an interactive tutorial with each diskette, a "help" screen listing key command descriptions for reference as text is edited, and an easel binder for at-a-glance aid.

If you're looking for high-quality at a low cost, PIE Writer is your best investment. Order yours today! 12009, Standard; 13409, Smarterm, Sup'R'Term, Videx

Available at your local computer store

Take a look at some of PIE Writer's powerful new features:

- Incremental Spacing–adds fractions of space between words for text justification.
- Fast page scrolling.
- Call command–allows access to machine language subroutines.
- Saves formatted output on disk file.
- Works with 40 or 80 column display boards.
- Word Tabbing–cursor stops at beginning of every word, in addition to set stops.
- Tab stops can be saved on file.
- Word Delete–deletes entire words at a time.
- Lower case capabilities.
- Prints **BOLD**.
- Centers.

All this and more . . .
for only $149.95!

Upgrade your PIE 2.0: Send $75.00, check or money order, and original disk (at sender's risk) for upgraded version.

or order direct;
1-800-343-1218
in Massachusetts 617-937-0200

Hayden

Mail To: **Dept S, Box 600 • Hayden Software Co.**
600 Suffolk Street • Lowell, MA 01853

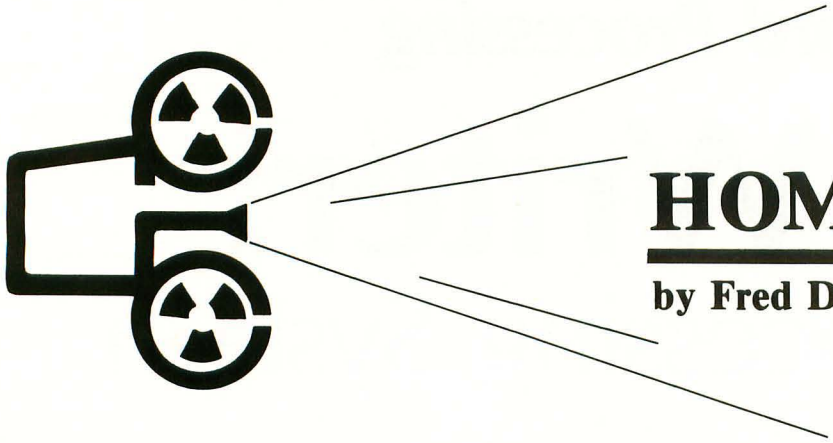
Please accept my order and send me the PIE Writer System for the following display board.

Please Check: ☐ 40 column (#12009) ☐ 80 column (#13409)
☐ My money order for $149.95 is enclosed; or
☐ Please charge my ☐ Visa ☐ MasterCard

Name _____
Address _____ Apt. _____
City _____
State _____ Zip _____
MasterCard/Visa# _____ Expires: _____
Signature _____

Residents of CA, MA, and NJ must add sales tax

ED-001 A501



HOME MOVIES

by Fred D'Ignazio and Allen L. Wold

Five forms of entertainment hold our culture in a strong grasp. Electronic games and personal computers are the fastest growing, followed closely by the new wave of Hollywood movies, including *E.T., Star Trek II, Poltergeist,* and *Tron.* Also having their effect are role-playing games, romance novels and soap operas.

Americans are pumping billions of dollars into these hugely successful forms of entertainment. Right now, each one is a separate industry, using a different media. What would happen, however, if we combined them into the single medium of interactive, digital fantasies, i.e. *computerized home movies?*

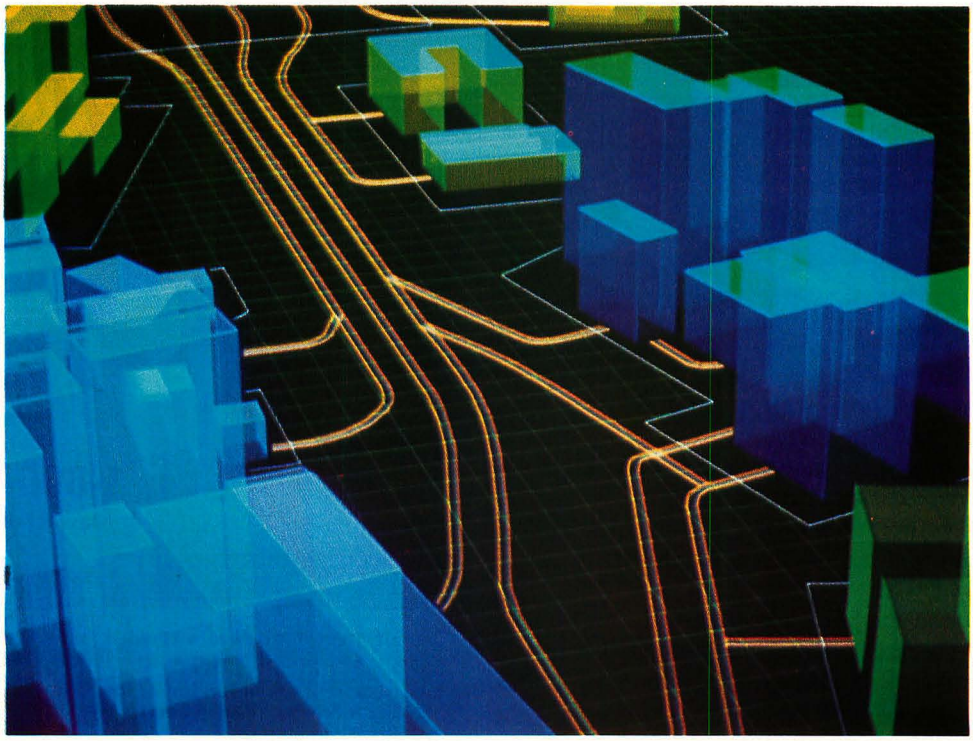
Star In Your Own Video Romance Novel

People love video games, adventure games, soap operas and romance novels for many reasons. Escape, adventure, the thrill of risk and danger, flirting with death, emotional release, sexual titillation; all are experienced through the magic of these media.

Some of these forms of entertainment are very personal. In adventure, role-playing, and video games, *you* are the hero. You can go on a quest and fight villains or monsters all over the universe. But the visual and audio technology supporting these games is still relatively primitive. Your imagination has to supply most of the color and detail.

On the other hand, Hollywood's new high-tech movies, soap operas, and romance novels are much more sophisticated in terms of imagery, special effects, plot and characterization. But, they involve adventures and experiences that happen to other people, not to you. To get into the story, you need to project yourself into the character on the screen or page. You need to get inside the character's skin and identify with him/her before you can experience the fascination, personal involvement, and the rare, delicious emotions that make these forms of entertainment so popular.

What if you could combine the media? What if you and a computer could make movies that are as sophisticated as the ones you see on screen, and personal because you are one of the principal characters? What if you could star in your own video romance novel, hot and spicy soap opera, or Hollywood science fiction epic?



Digital scene simulations like this one appear in TV commercials and movies. You will be able to choose computer-generated scenes for your "home movie" from a large electronic library.

Photo courtesy of Robert Abel & Associates and AT & T.

The Adventure Tailor

Computers can't help you make a movie now, but they might be able to soon. The field of "intelligent computers" - *knowledge engineering* -is making great strides. Computer scientists have already created *expert systems* by mining the brains, creativity, design skills, and problem-solving techniques used by some of the most brilliant human experts. So why can't we postulate the creation of electronic movie experts — film directors, cameramen, sound technicians, and

Out of the darkness, the director's voice booms: "Get ready for Take One, Scene One." You take off your actor's robe and begin.

scriptwriters — all to help you put together your own home movies?

The director of your home movie — your "adventure tailor" — could be a highly skilled computer. No matter what part you played, no matter what sort of script you created, only your computer director would know. When the movie was complete, you'd carry the movie home on a floppy disk, bubble memory, or ROM cartridge. If you preferred, you could be the only person who ever saw your movie.

Rent-A-Studio

It will be some time before you will be able to afford the equipment needed to make a sophisticated digital movie at home. And talented directors are expensive — even if they are computerized. So, you could make your movie at a commercial, do-it-yourself studio. It would be completely computerized — an automat of fantasy and self-expression.

When you actually began making your movie, it would be like working out at a health spa. You'd go from room to room, on your own, using the different equipment. Only you wouldn't be pumping iron. You'd be exercising the "muscles" of your id, your subconscious, and your imagination.

Flirting With The Ticket Lady

Let's step into the future and pretend you're ready to make your own movie. You hop in your car and drive to the nearest *Auto-Movie* studio. You hand your ROM-embedded credit card to the ticket lady at the entrance. She gives you a foxy smile. You have to struggle to remember that she's really just a humanlike automaton — preprogrammed and plastic. Your credit is good. The ticket lady winks and returns your card. A door opens, and you enter the studio.

You sit down in front of the computer scriptwriter. You communicate with the computer using a keyboard and microphone. Step by step, the computer helps you shape the type of fantasy you'd like to create. After administering an interactive questionnaire, the computer displays — in text and simple wire diagrams — several "pilots" of your movie. You react to these pilots and suggest changes. After adding several refinements, you and the computer come up with an acceptable script.

With the script completed, the door opens to the costume room. You outfit yourself in one costume, or several, whatever fits your newly-created script. Perhaps you put on a silk shirt, a vest, a scarf, pantaloons, and leather boots to become a pirate. Or, you might don a futuristic spacesuit. The wardrobe is extensive and it all fits. Other people with other measurements are directed to different wardrobe rooms.

Lights, Camera, Action!

A new door opens. You enter a room filled with a large soundstage. No one is there except you, your computer film crew and director. A few simple props are set up for your first scene. Out of the darkness, the director's voice booms: "Get ready for Take One, Scene One." You take off your actor's robe and begin.

You do the first scene, the next and the next. Sometimes, you must redo a scene twenty or thirty times. The computer director is tireless and exacting. If its cameras haven't caught everything according to the specifications of your script, the director isn't satisfied. Exhausted, you finally stalk off the stage, muttering loudly about inhuman directors. Once back in your dressing room, a soft, gentle voice asks if you'd like to make an appointment to shoot the rest of your movie the next day.

Pulling Kisses Out Of Thin Air

Part of the strain of acting in your own movie comes from the director's relentless nagging and prodding. But another part comes from the fact that you are acting everything out solo, without support from or interaction with other actors. In addition, you use few, if any, physical props, there is no scenery, no backdrop, no artificial lighting that suggests the movie's location. These things will all be added later, by the computer. In the *Star Wars* pictures, Disney's *Tron,* and many of the other new high-technology films, the actors had to work with similar constraints. First, the live action was shot. Then all the scenery, special effects, and animation were added during the post-production stage, often months later.

But, you don't have to wait months. Your movie producer is automated and quick. It can turn out your film in only a few days — about the same time it takes to get back color slides or to have a shoe repaired. Seconds after the live action shooting is over, the electronic director begins creating a movie data base. It has to make trillions of calculations as it converts your scenes from video images and sound waves into

electronic pulses. All the scenes are digitized and stored as bits and bytes in a private movie data base.

The director manipulates the data base and starts to edit the movie. It takes your love scenes and adds a simulated partner — with specified dimensions and qualities. It takes a ridiculous-looking scene, in which you waved your machete at thin air, and transforms it into a terrifying episode in which you singlehandedly battle an enraged Tyrannosaurus Rex. It adds supporting actors, supporting voices, and supporting sounds. To a scene at sea, it adds the waves and the boat. To an outer space scene, it adds the stars, the fiery comets, and the whirling, ringed planets. It edits all these things into your movie during this "post-production" phase.

Invitation To A Private Screening

A week after you finish shooting your movie, you return to the studio for a private screening. You can bring your family, friends, or come on your own. As you sit down in a soft, cushioned seat and stuff a handful of popcorn into your mouth, the theater darkens. A fantastic title sequence appears. Then the actors' names flash across the screen. Your name appears first, of course, in the largest letters. The names that follow are all mythical, invented by the computer.

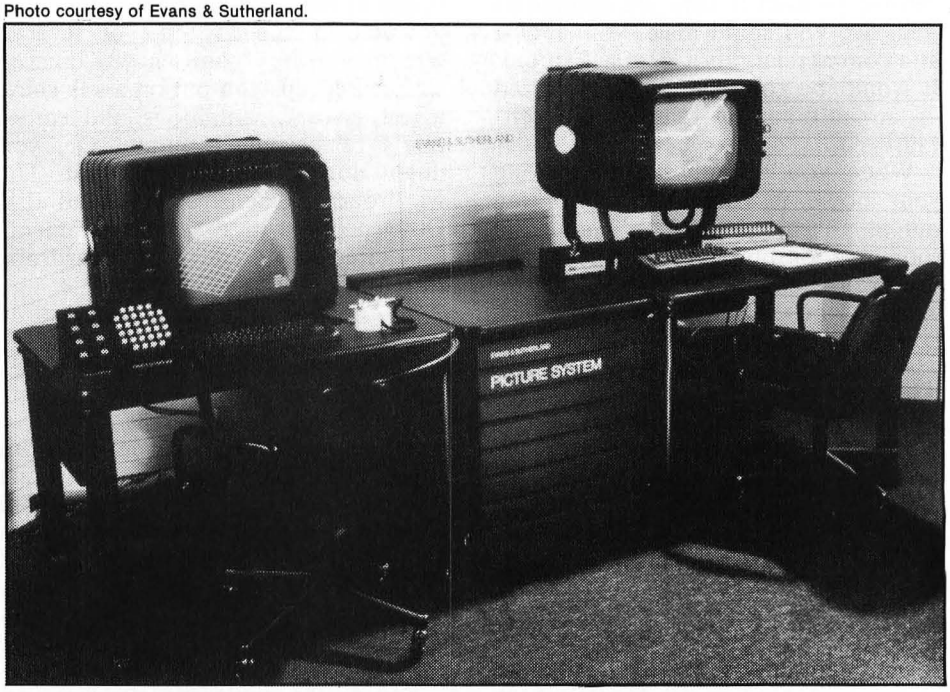
The movie begins, and there you are — a knight riding a beautiful white horse, or perhaps a space commander journeying into dangerous, uncharted sections of the galaxy. You might be a police precinct captain in a violent section of New York City, or a TV producer, rock star, ace reporter, model, jock superstar, heartthrob, race car driver or brilliant doctor.

No matter who you are, you move in a world of explosive, raw emotions where you do spectacular things. Your life is perilous, thrilling, *unpredictable*. Your appetites are gargantuan. You are surrounded by beautiful people, or wizards, or perhaps even aliens. Or maybe you are in Eden, all by yourself. You are God's lone explorer, free to savor all of Creation on your own. The movie's action is stupendous, the scenes are breathtaking.

At the end of the picture, you applaud. "Encore! Encore!", you cry. The automated projectionist gladly obliges. It shows you the film again as many times as you wish.

Late that night, after seeing your movie for the seventeenth time, you stumble out of the studio and onto the street. You blink, and your legs feel weak. In your hand you clutch the ROM cartridge which contains your movie so you can play it later, at home. You can't help smiling. You shake your head. "What a movie," you whisper, "What a movie!"

Photo courtesy of Evans & Sutherland.



The Multi Picture System provides real-time user interaction with complex 3-D images. You will sit at a work station like this one to edit the scenes in your computerized "home movie."

The Adventures of PROFESSOR VON CHIP & ORBIE

TRICKY TUTORIALS (tm)

There are many things that the ATARI computers can do either better, or easier than other small computers. The following series of programs is designed for anyone who is at least familiar with BASIC programming. What each tutorial offers is similar to an extensive magazine article with all discussion in as simple language as possible, plus you get MANY examples already typed in and running. The instruction manuals range from 10 to 50 pages, and some tutorials fill up a complete tape or disk. There is little overlap in what is taught, so anyone wanting to know all they can should buy them all (my banker thanks you). ATARI buys these from us to use in training their own people! Rave reviews have been published in ANTIC, ANALOG, CREATIVE COMPUTING, and even INFOWORLD. You trust INFOWORLD, don't you?

TT #1: DISPLAY LISTS—This program teaches you how to alter the program in the ATARI that controls the format of the screen. Normally, when you say "Graphics 8", the machine responds with a large Graphics 8 area at the top of the screen and a small text area at the bottom. Now, you will be able to mix various Graphics modes on the screen at the same time. The program does all of the difficult things (like counting scan lines). You will quickly be able to use the subroutines included in your own programs.
16K Tape or 24K Disk. **$19.95**

TT #2: HORIZONTAL/VERTICAL SCROLLING—The information you put on the screen, either GRAPHICS or TEXT, can be moved up, down, sideways, or diagonally. We provide the basic methods and leave the rest up to your skill and imagination. Includes 18 examples to get you started, with several using a small machine language subroutine for smoothness.
16K Tape or 24K Disk. **$19.95**

TT #3: PAGE FLIPPING—Now you don't have to redraw the screen every time you change the picture or text. You will learn how to have the computer draw the next screen you want to see while you are still looking at the previous screen, then flip to it instantly. You won't see it being drawn, so a complicated picture can seem to just appear. Depending on your memory size and which graphics or text modes you are using, you can instantly look at up to 50 pages. The basic method takes only 9 lines and the usefulness is infinite.
16K Tape or 24K Disk. **$19.95**

TT #4: BASICS OF ANIMATION—This program shows you how to animate simple shapes (with some sound) using the PRINT and PLOT commands, and it also has a nice little PLAYER/MISSILE GRAPHICS game you can learn from. The P/M example is explained and will get you started on this complicated subject (more fully explained in TT #5). This would be an excellent way to start making your programs come alive on the screen with movement! Recommended for beginning users.
16K Tape or 24K Disk. **$19.95**

TT #5: PLAYER/MISSILE GRAPHICS—Learn to write your own games and other animated applications! The tutorial begins with many small examples that compliment the 50 page manual, then gradually builds up to a complete game where everything you need to know is fully explained. Also included are two machine language utilities that you can use to animate Players with from BASIC. Next we include two of the best editors currently available; one for editing playfield shapes (backgrounds); and one to edit your players, and all in glorious Technicolor!! Everything except the two editors run in 16K Tape or 32K Disk. **$29.95**

TT #6: SOUND AND MUSIC—Unless you have spent many years experimenting with the four voice channels, you will learn a lot from this one! Learn to play standard notes, chords, and whole songs using some simple "tricks". One of the nicest parts are the examples of special sound effects that you can refer to whenever you need a sound for a program or to impress a friend. This program will be of interest to all ages and levels of experience!
16K Tape or 24K Disk. **$19.95**

SPECIAL DISCOUNT

Order the first six tutorials in a 3-ring binder for $99.95, a $30.00 savings!

TT #7: DOS UTILITIES—We at Educational Software have been shocked by some of the prices others are charging to offer you small utilities to help in the use of your Disk Drive. We now offer you all of the following plus explanation as to how each was written, and how to use them: A UNIQUE MENU PROGRAM, AN AUTORUN.SYS BUILDER, DISK INSPECTOR (LOOK AT SECTORS), DISK JACKET PRINTER, AUTOMATIC FORMATTER, RECORD SAVE AND LOAD UTILITY.
32K Disk Only. **$29.95**

MASTER MEMORY MAP (tm)

This book is the most valuable source of information for your ATARI you can buy. It starts out by explaining how to PEEK and POKE values into memory, so that even new computer owners can use many of these "Tricks". Then you are given 32 pages of the memory locations that are the most useful, along with hints on how to use many of the locations. Finally, it includes hints on problems you may be having with the computer and discusses the new Graphics modes 9 to 11. Even ATARI buys this book from us! **$6.95**

CONTACT YOUR LOCAL DEALER or ORDER BELOW

We have other fine programs for ATARI computers.
Write for a catalog.
Send us your programs to sell too!

USER SUBMITTED PROGRAMS

MINI-DATABASE/DIALER—stores and edits up to 8 lines of information such as names & addresses, phone numbers, messages, inventories, or anything you want. It has the usual sort, search, and print options, but it also has an unusual feature: If your file includes phone numbers and your phone company allows touch-tone phone signals, the program will DIAL THE PHONE NUMBER FOR YOU! 16K Tape or 24K Disk. **$24.95**

THE GRAPHICS MACHINE—allows the ATARI to act like more expensive graphics computers using simple commands like line, box, circle, polygon, fill, and savescreen to get a high resolution picture you can save on disk in only five seconds! Many more features! 48K Disk Only. **$19.95**

BOB'S BUSINESS—14 small business type programs accessed from a common menu. 16K Tape or 32K disk. **$14.95**

MINI WORDPROCESSOR—A simple text editor to write, save, and print several pages at a time. 32K Tape or Disk. **$19.95**

KID'S #1—Includes a MATH QUIZ, a children's TREASURE HUNT, and a DIALOGUE program. 16K Tape or 24K Disk. 3 for . . . **$14.95**

KID'S #2—SPELLING BEE, WORD SCRAMBLE, and TOUCH. 16K Tape or 24K Disk. 3 Educational games for . . . **$14.95**

PLAYER PIANO—Turns your keyboard into a mini-piano and more. 24K Tape or 32K Disk. **$14.95**

DOG DAZE—Two cute little doggies race for the fire hydrants, shoot their bones, and just have a lot of fun! A fast action program for all ages. 8K Tape or 16K Disk, in machine language. **$16.95**

GRAPHIC SYMBOL LABELS—for your keyboard to remind you of the built-in Graphics symbols. 2 complete sets for . . . **$2.95**

OUR NEWEST PROGRAMS

*** OUR BEST GAME ***

SPACE GAMES—Our family is being attacked by ALIENS, and only you can save us. A comic book manual will guide you through three games that test your ability in space skills. Includes ALIENS, SURVIVE, and ROBOT ATTACK, and is for all ages. The first two games require 16K for Tape. The last game and all Disk users need 32K. **$24.95**

MATHS FOR FUN—Another ENGLISH import teaching basic math skills. Very colorful and enjoyable to use. For ages 5 to 16. 16K Tape or 24K Disk. **$19.95**

TT #10: SOUND EFFECTS—From laser blasts to ringing phones, this tutorial will show you how to make unique sound effects in all of your programs! 16K Tape or 24K Disk. **$19.95**

MARATHON—This is a unique math quiz for one or two players. You are in a race to move your runner across the screen first! There are four levels of play with five modes of operation for each. The game uses joysticks for all input, so play is easy for young children. This wonderful learning tool is imported from ENGLAND for your learning pleasure. Your kids will never even notice they are playing an EDUCATIONAL program. 16K Tape or 24K Disk. **$19.95**

TT #8: CHARACTER GRAPHICS—Character Graphics is the best way to animate your ATARI! Make letters look like space monsters, gunfighters, or a myriad of other shapes. Use our editor to create these multicolor shapes and then we'll show you how to move them around the screen. This tutorial even shows how our Space Games were written! 16K Tape or 24K Disk. **$19.95**

To order COD, VISA or MasterCard call — **(408) 476-4901**

By mail include $3.00 postage (USA only) or $1.50 Memory Map only—
California residents add 6.5% TAX.—Specify Tape or disk.

Educational Software inc.

4565 Cherryvale Avenue
Soquel, Ca. 95073
(408) 476-4901

THE ELECTRONIC RENAISSANCE:

THE FUTURE IMPACT OF COMPUTERS ON ART AND CULTURE

by Saul Bernstein

The biggest challenge to the working artist is his process of communication...by communication, I mean not only his interaction with his audience but his explanation of a new vision of the world that he personally sees, interrelated to the time in which he lives. In the case of commercial art, intercommunication with corporations, agents, directors, producers, TV and film people is absolutely necessary. When one addresses the processes of communication, he can't help but see television in the forefront. In fact, television's greatness occurs when it is used as a tool for documentation of an event. Who can forget the assassination of John Kennedy or the landing of the first man on the moon?

When I was a young art student taking a course on the History of Architecture, my textbook was one written by Sir Bannister Fletcher. It was an incredible book, for it broke down the whole history of the field into many influences. These included religious, economic, climatic, geographical, historical, etc.. The same influences affect the artist as well. The logic would follow that, in the 16th century, all people who lived in Italy had one common bond...the Catholic religion. This certainly would color all perceptions of the word of God. Today, living in a melting pot such as the United States, the painter creating the same paintings would have to contend with an audience of many religions, cults and persuasions.

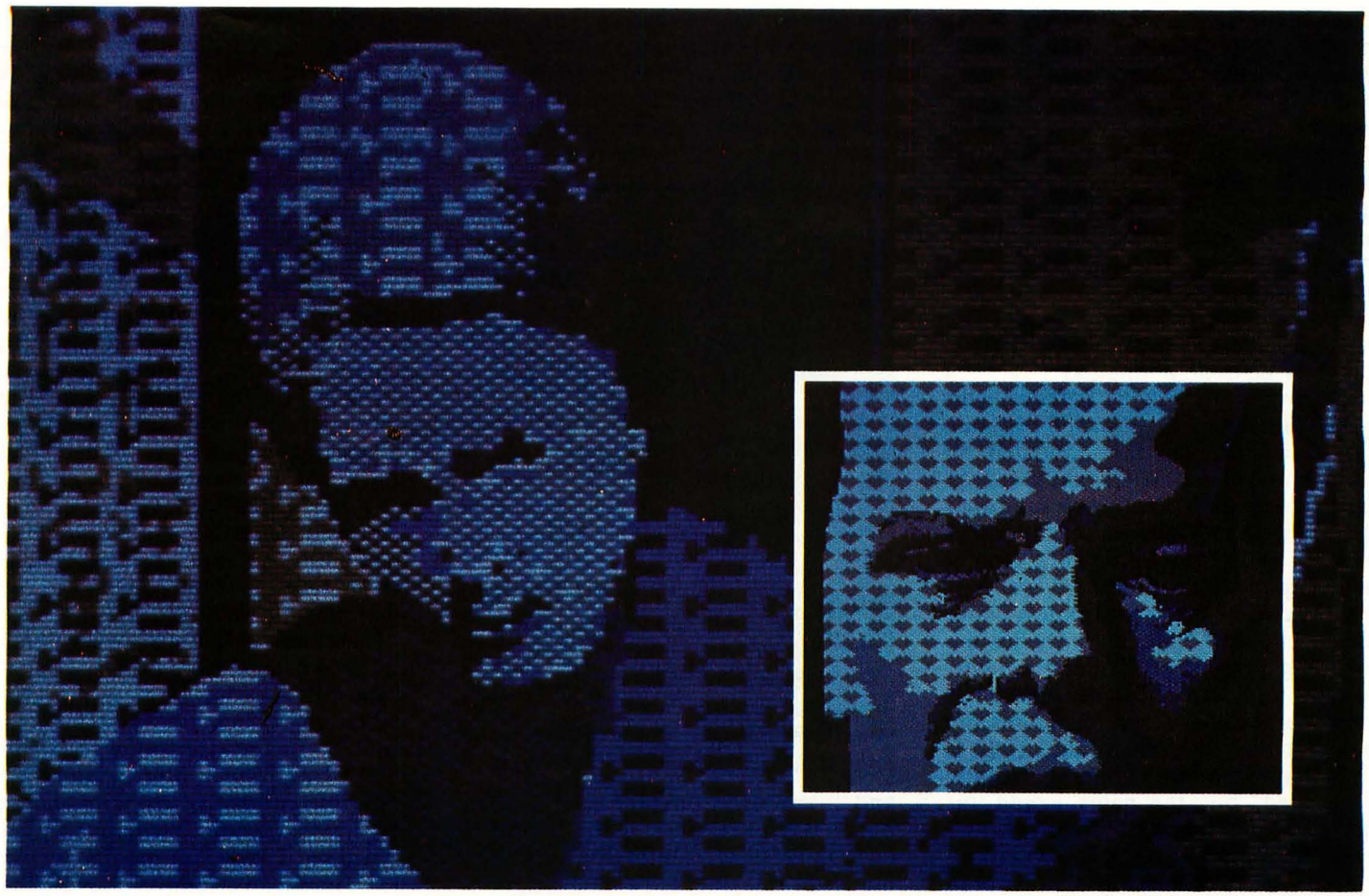
It has been proven true that the country which produces the great works of art leads the world economically, as well. This may be so because interaction between the type of people who produce great wealth is essential for the artist's continuing desire for something new. Or maybe it is because art is a luxury item, and that takes money.

Influences such as climate would also alter the images produced by artists. Imagine, for a moment, that Van Gogh lived in Alaska. The first thing to become apparent would be that the painter could not go outside unless he had the help of science. Heating inventions would be needed that were not available in the 19th century. Suitable climate was necessary in order for Van Gogh to conduct his investigations. As any student knows, the Impressionists did have the help of some inventions. The camera certainly influenced Degas, and the new-found ability to put paint into tubes helped liberate the painter from the studio.

As a student who was stunned by this kind of thinking, I tried to look back on our society from the year 2050. The event that most struck me was the landing on the moon and our ability to transmit the pictures around the world. I felt then and feel now that we will be remembered as a technological, communicative culture of the first kind. Today, in medical technology we can use X-ray, sonar, and infrared photography to analyze the patient. Certainly, if Leonardo da Vinci were alive today, his work concerning human anatomy would be different, for his reference point would differ from that of the 15th century by the gifts that science has provided.

Artists of the past were not only practitioners of the pictorial arts but architects, engineers and scientists as well. When El Greco died, it was discovered that he possessed a very large collection of books on the field of optics. Rembrandt took up a new field of etching called aqua-tint, and Goya helped discover the new field of Lithography, later further developed by Toulouse Lautrec. Michaelangelo was a painter, draftsman, sculptor and architect. Leonardo was all of these, with additional interests in the fields of engineering, anatomy, flight, botany, etc.. From my point of view, science has always played an important role for the pictorial artist.

There have always been, and probably always will be, two kinds of artists. One, the decorator, paints with a goal of entertaining people. The decorator paints for many audiences...from individuals to museums. All too few painters are what I would classify as moralists — teachers. The artist has the opportunity to *teach* the viewer how to see, just as the great composer teaches us how to hear. Great



literature makes us think, and wonderful acting makes us emote. There are very few great painters in the "hall of fame" who deserve the title *Artist* or *Genius*. All too often, painters who show their works in the cities of America and the world are called the geniuses of our time. Their work, over a period of time, begins to pale into the world of decoration. The great *Artists*, who have taught us how to see, begin with Giotto, Rubens, Rembrandt, El Greco, Velasquez, Goya, and Cezanne. Very quickly thereafter come the decorative painters. It is hard to appraise the work of Frans Hals as one of the greats, for instance, even though his work is alive with wonderful technique when compared with his countrymen. Hals' work, indeed, looks surface-oriented and decorative.

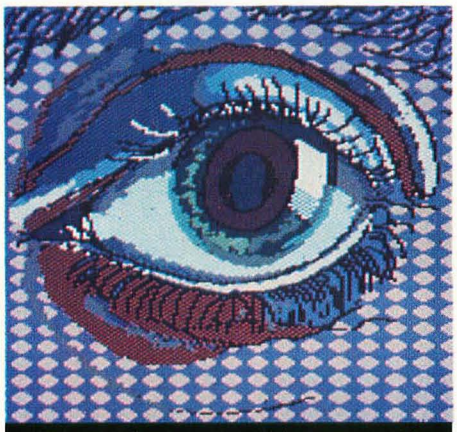
As an educator, I am attracted to the moralist point of view and, in one sense, see the television screen as the largest classroom in the world. Psychologically, people *want* to believe what is occurring on the screen. There are many reasons for this, but an argument can be made in "seeing is believing." As a drawing and painting teacher, I got into the field of electronics for that very reason. During the 60's, when the students on my campus were carrying placards denouncing society, there was one that really came home. It clearly stated, "Make our education relevant." After thinking about this plea for some time, I decided to build my own television studio, for I felt with this tool I could better communicate with the younger generation. I went out and interviewed people in the industry and taped their views concerning what was relevant in the fields of television, film, advertising, etc.. After showing the tapes to my students and finding that the results were even better than expected, I continued my efforts further. The students felt better about what they were learning...they could criticize their work and work from a new, more relevant, point of view.

I was psychologically attracted to the television screen because most kinds of vision are what I call recessive. By that, I mean that when you look at an object you can see it only because of the light that surrounds it. Paintings, sculpture, and nature all fall into this category. There are only two art forms that I can think of where the color is on the "attack." One is stained glass, where painted glass is between the viewer and the light. This may be the reason why stained glass windows are so effective in churches. The other form is television, where light guns illuminate the viewer's screen from the rear. I feel that this experience is very close to hypnosis and may be the reason for the popularity of computer and arcade games. Another fact to be considered by all painters is that people react to not only color, but movement and sound as well. The beauty of the computer image is that it can be either static or animated.

Another advantage of the computer is that when I'm working with an art director, he can leave the studio with a print-out of the day's work, and later, write on the print-out, communicating his particular

needs to me. The fact that the artist need not leave his studio, but can send his pictures using a modem, is also very valuable. The other party can interact with the artist via his keyboard so that further refinement can be made instantly, as well. One can imagine, for instance, the artist who lives in Utah sending his work to his director in Los Angeles, and after a few minutes of consultation, the director sends the results to New York, Florida, or wherever. What took weeks before can now be done in one day.

According to Alvin Toffler, who wrote the book, *The Third Wave*, our future lifestyle will be dominated by cottage industries. There was a survey taken some years ago which indicated that some 35 to 40 percent of the working day is wasted time. They simply added up the time of idle conversation, travel, coffee breaks, tardiness from lunch, etc. and came up with this astounding percentage. Sooner or later, companies will realize that the employee would serve the company better in his own environment. It is true that waste would not be totally eliminated, but the gains would far outreach the losses. I feel it would make for a happier work force than we have now. Commuting time would almost be eliminated. The elimination of that frustration alone might make it worthwhile. In the field of education, things would be different, also. A student would still go to a classroom and receive individualized instruction from his teacher. However, he would return to his house for the majority of the day

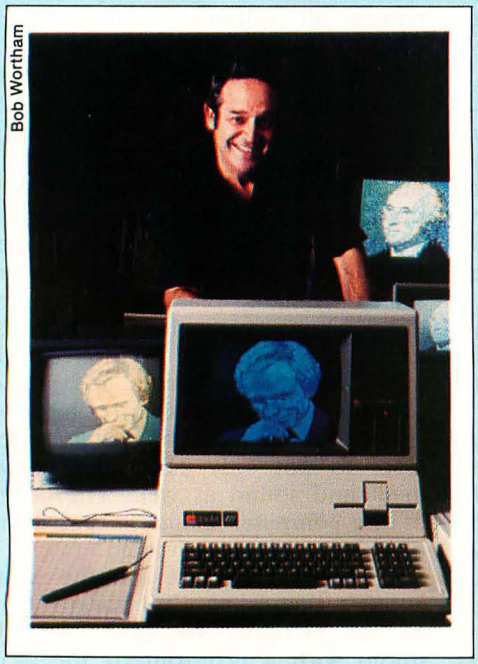


"The "big brother" image still follows the computer when, in fact, it is the tool which is putting control of the arts and our very lives back into our hands."

and insert a lesson tape into a computer or interact with a cable company's transmission of the material. Imagine, for a moment, hundreds of students learning at their own paces using the technology of the day.

These things should bring about a restructuring of the family unit — perhaps not along the same lines that existed during the agrarian civilizations, but with a higher degree of cohesion. We have the highest divorce rate today in our history. A NOW society, a hedonistic society; self-ambition and self-fulfillment to the point where the beauty of life seems to be missed. The possibility that technology can overcome all of this is very real. The husband, wife, and children would all participate in the family welfare. The education of the children would be more in the hands of the parents, and the essential values of Western civilization could be personally enunciated. No more would students get only the theories of harmony — they would act them out with their own families. They would learn the responsibilities to each other and for each other.

There are individuals who predict that American culture is being brought down by its concentration on computers and technology. The "big brother" image still follows the computer when, in fact, it is the tool which is putting control of the arts and our very lives back into our hands. I think we will see a revival of the fine arts and the American family in a high technology era that will make us the envy of the world.



Saul Bernstein — Artist Extraordinaire

Saul Bernstein, also our cover artist this month, began his involvement with art at the tender age of eight years. He holds a Master's Degree in Fine Arts from Otis Art Institute and, since 1960, has been a Professor of Art at California State University at Northridge.

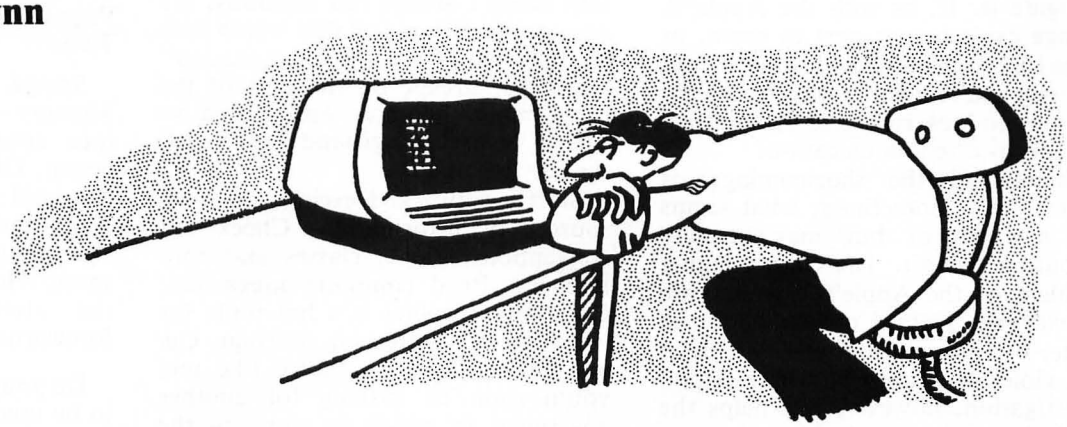
Saul creates his masterpieces on an Apple III™ and a graphics tablet. His system is augmented by a board from Video Associates Lab of Austin TX, which converts the Apple™ video output to standard NTSC broadcast signal. It can then be passed to an ink-jet printer from PrintaColor of Norcross, GA, or to photographic equipment from KineGraphics of Simi Valley, CA. With the Kinegraphics box, he can photograph the image on Polaroid SX70® or Kodak® instant film, as a print or transparency.

His many accolades include an Emmy for Instructional Programming for the series, *Needlework*. The series was also the recipient of the West Award for excellence in educational broadcasting. It has been broadcast nationally on PBS and on the West Coast by ABC.

His work is also featured in the title sequence of *Remington Steel*, a new series on NBC about a detective agency which utilizes computers in its work. It airs on Friday evenings at 10 p.m.

AN ARTIST EYES A COMPUTER

by Ame Choate Flynn



Editor's Note: Ame Choate Flynn, owner/operator of TechniGraphics in New York City, is a "working artist" in the microcomputer field. Her background in the fine arts includes 12 years of formal education, culminating at the Rhode Island School of Design. She is a member of the faculty at The New School in New York, teaching various courses in microcomputer graphics, and chairman of the graphics group of the Big Apple Users Group, also in New York. She has spoken at many computer graphics seminars throughout the US. Her computer system is a 64K Apple II Plus™ with a Graphics Tablet. Her clients at TechniGraphics have included such prestigious companies as McGraw Hill and Merrill Lynch.

I remember walking into my first computer graphics convention. It was mind boggling. Here were all these high-powered machines doing things that I couldn't spell, much less figure out. There I was, an artist, trained in her craft, wondering if I had anything in common with all these technological wonders. Didn't computer graphics include graphics or art? Where would I start a conversation with these people? Did we have anything to say to each other? Would I be laughed off the floor? It was very exciting, but awesome and dismaying at the same time.

It's been two years since SIGGRAPH in Seattle. I've found a niche and have learned more than I ever wanted to know about computers – what makes them go, and what makes them bomb and crash on occasion. I started teaching – passing on the "secrets" of computer graphics to other artists. I've been well rewarded by seeing what they've produced in and out of class. The field is still awesome, but it's more exciting and more "doable" than ever.

The first formal training an artist usually receives is a "mixed media" class. This is not film and video tape, but a melange of charcoal, pastels, watercolor and more. If sculpture is included, the artist learns how to work with clay, wood and stone using fingers, knife, hammer and chisel. We learn the various properties and problems of each tool. How charcoal and pastels flake off the paper. What to do about it? Carry a can of fixitive to make them adhere. We learn time-honored techniques for using these implements and, at some point, imitate (consciously or unconsciously) the "Old Masters." How did Van Gogh make a Van Gogh?

Having honed these skills with the passage of time (and many crumpled sheets of paper), the artist knows automatically how a given medium will behave. Indeed, we no longer think about it, but use the medium to express our own individual style and creativity. Our hands have become trained to the point that the line is an extension of our mind. Use of tools is no longer a process of learning or thinking, but is automatic. It is knowing what they will do.

Computers seem far removed from the intimate personal experience between an artist and his creation. You no longer "wear" your art. It's easy to tell if someone has been working in oils. Just the smell will give it away. There is no grubby contact sense from handling floppy disks, using a graphics tablet or dealing with bits or bytes. Perhaps a pink tinge to the eyes or a glazed look from information overload, but on the whole, your outfit will look the same after a session with the computer as when you started. How can you make a computer graphic "yours" — bring your own personal style and flair to something created with the help of a keyboard? It can be done. But you must become as familiar with the computer as an art medium as you are with the traditional tools. There are several ways to go about this familiarization process. What follows are some of the ways I've found.

Large system computer graphics are wonderful. They have speed, resolution, memory and lots of colors. They also have a few drawbacks. Access to the system is one. You usually can't get your hands on one unless you work for a company that owns one, or happen

to proudly possess one yourself. Obviously, the more powerful a system is, the more expensive it is. A DEC PDP 11/34 is hardly ever found in an artist's garret. But an Apple™ or an ATARI® will fit perfectly in the available space, and for an affordable price.

Once you choose your system, investigate it. If, as with the Apple™, orange can't be put next to green, or black1 has to be used with white1, find out why and remember why. You want your art to look right and be an effective means of communication.

Understand the shortcomings of your system. Sometimes, what seems like a glitch or bug may actually become a benefit. The color-fringing problem of the Apple's™ whites are an example. Instead of getting a true white, it is usually a mixture of green and violet or orange and blue. Upon investigation, however, this helps the resolution and gets rid of the most obvious aliasing or "jaggies."

Does your computer have "pages" and "planes?" In what language are you working? You may not have to learn a programming language or how to program, but you will have to learn certain commands in order to save your work for future perusal.

Choose your tools, and use the right tool for the job. Learn the terminology. It is extremely important to know what you are saying and to say what you mean. RAM may look similar to ROM, but they're really quite different. Get used to saying pixel for dot; find out what CAD/CAM means.

Practice, practice, practice. Learn your computer and your software. Train your hands and your mind.

If you are rendering illustrations from print to the computer, you must know how to work with the problems of resolution. What lines or objects can be deleted (so the picture will fit on the screen) and still retain the essence or the idea. You may have to work within a partial screen, the top and/or bottom may be needed for print or for text.

It is easy to feel limited by (or even ashamed of) your "small computer." Imagination and creativity know no bounds, even if they are swung through a small micro. Let your talent shine through. If the software tools or utilities you want do not exist, help create them. Translate your favorite art world medium into the computer.

Get to know some programmers. They don't all wear white socks and hide out in dungeons full of strange machines. Many of them are quite literate and very interested in your problems. They need ideas for their software and are usually willing to help when you find yourself in a corner. I'm aware of several artist/programmer couples or teams, and if the relationship doesn't always run smoothly, it's interesting! If you feel that words such as "algorithm" or "concatenation" are rather arcane on the part of the programmer, counter with "negative space" or "reverse ground" and watch the fur fly.

Start a network. Develop your own sources of information. Check out conventions, clubs, classes and conferences. Read computer magazines. Computer graphics is a hot topic for which there is increasing coverage. Get on mailing lists. If you're like me, you'll soon be looking for another apartment in which to store all the back issues and product information brochures.

Keep up with developments in the field. If you can't travel the country following conventions, start a user/graphics group in your city. This is a handy way of learning about new software and applications. Realize that your ideas and thoughts are just as valid as the other members' and they want to hear them. Talk to people – manufacturers, developers, artists and engineers. There is usually someone in every city who knows what you need to learn and is willing to help you.

When you have your computer and know how to use it, you'll need to find work. To find it, the following factors or definitions should be taken into account:

Artistic Considerations– The artist's style must be translated to the computer. The artist must learn new terms, shortcuts and ways to "trick the viewers' eye." Color must be used effectively and tastefully. To establish it as an art, computer graphic artists must create their work with the same care and concern they have always used. The public will soon become educated to expect computer art that lives up to traditional art standards. Just because art has been created on a computer will no longer be an excuse for poor images.

Idea – the theme of the picture or graphic; supplied by the artist or client, or by the artist for the client. Do not expect the client to always agree with your idea.

Medium or Method — a software tool used by the artist to translate ideas or inspiration to screen or disk via computer.

Routine– how the idea is displayed on screen in the final product. It can be a method of animation, or a way of changing from an image displayed on Page One to another displayed on Page Two.

Speed, Resolution, Available Memory – factors which must be taken into consideration before work is begun. Often a piece of software will not hold a series of "full graphics." Ways must then be investigated to "picture pack," or get graphics in less space. There may be a fight here with the client or programmer. Be forewarned.

Do you know how the image is going to be used? Towards what age group? Is it an illustration for a game? Is the art a game? Work with the programmer, the engineer and the client. Know what can be done. You can't do *TRON* graphics on a 48K Apple II + ™ — yet.

A piece of educational software has different goals than a game, or art for art's sake. Would low resolution graphics get the job done more effectively? How can animation techniques be used? What portion of the available memory is allotted for graphics? This is perhaps the most important factor in the first steps from an idea to reality. It often dictates how a graphic will be created, what software can be used and what will be done to it.

To carry through a common theme, you may have to design a text font to go with the graphic. If a particular trait common to professional typography is needed, you may even have to help a programmer design a character generator.

One very positive factor in computer graphics is that you are no longer bound by centuries-old thinking, customs and habits. You are breaking new ground with every stroke, be it key or stylus. There are no canvas-bound limitations, once you know your (and your system's) capabilities.

Don't feel that you have to learn everything at once. Take computer graphics one step at a time. It's an exciting field which will benefit from individual style and aptitude. Life may get expensive, and full of floppy disks, but it's an investment in the future, using future technology.

GENERAL INFORMATION

Concerning SoftSide line listings, SWAT & Magnetic Media

Follow these procedures unless otherwise instructed by the documentation in the magazine. Back issues may differ in some details.

SWAT TABLES

At the conclusion of each line listing of a ***SoftSide*** program, we include a *SWAT* (*Strategic Weapon Against Typos*) Table. *SWAT* was published in issue #30 of ***SoftSide*** and is available as a free reprint. Please send a self-addressed, stamped envelope to ***SoftSide*** Publications, Inc., Dept. *SWAT*, 6 South Street, Milford, NH 03055.

APPLE™

Disks are in 13-sector format, created under DOS 3.2.1. If your system is set up for 16-sector disks (DOS 3.3), first boot your BASICS disk or BRUN BOOT13 from the System Master Diskette, then insert the ***SoftSide*** disk. A cover/menu program will run automatically.

Tapes LOAD in the normal manner. Advance the tape to the beginning of the lead-in tone; stop the tape; insert the plug into the EAR jack; type LOAD; start the tape; and press RETURN. Side two of the tape is a duplicate of side one, unless one or more Integer BASIC programs are included, in which case side two contains the Integer programs.

ATARI®

Line Listings use the following conventions in representing unprintable characters, unless otherwise noted:

Characters (including blank spaces) which are underlined should be typed in inverse video.

When graphics or control characters are to be included in a string (between quotation marks), it will be noted in a nearby REMark. In such cases, graphics characters are represented by the corresponding lower-case letter, and control characters are represented by the corresponding unshifted key symbol. For example: The lower-case letter s represents a control-down-arrow, entered by first pressing and releasing the ESC key, then holding down the CTRL key and pressing the = key. (See Appendix F, and the back cover, of the *ATARI® BASIC Reference Manual.)*

The one exception to the above practice is that a clear-screen character (ESC CTRL-↰) is represented in listings by a right-hand brace, which looks like this: }

A shifted = is represented in the listings by a vertical line with a small gap in it: ¦

SWAT — Before appending *SWAT* to a program in memory, the program to be SWATed must first be LISTed to disk or cassette (using LIST "D:FILENAME" for disk or LIST "C:" for tape). Next, turn the computer off, then on again, to clear the system and ENTER the program back into memory (using ENTER "D:filename" for disk or ENTER "C:" for tape). Because of the unique method in which *ATARI® BASIC* stores variables in a program, the variable table must always be in the same order to produce accurate *SWAT* codes. LISTing and ENTERing the program is the only known way to rebuild the variable table in a specific order so that *SWAT* codes can match.

Disks do not contain DOS.SYS files, and are therefore not bootable by themselves. First boot a disk which contains any version of DOS, then insert the ***SoftSide*** disk and RUN "D:COVER" (*Adventure of the Month* — RUN "D:INTRO").

Tapes CLOAD in the normal manner. If you have difficulty, try this procedure:
(1) Type POKE 54018,54 and press RETURN.
(2) Turn up the volume on your TV.
(3) Type CLOAD and press RETURN once.
(4) Press the PLAY button and listen.
(5) When you hear a steady lead-in tone, press RETURN again.
Side two of the tape is a duplicate of side one.

TRS-80®

Disks are available in Model I or Model III format. They contain the DOS PLUS operating system, and a cover program which automatically runs upon booting. Back issues prior to May, 1982, are available only in Model I format, and may be converted using the TRSDOS CONVERT utility on a two-drive Model III. Older back issues (with Model I TRSDOS) require you to enter BASIC and then type RUN "COVER".

Tapes CLOAD in the normal manner on Model I's, and at low speed (500 baud) on Model III's. The first program is a cover/menu program. Side two of the tape is a duplicate of side one.

NOTES ABOUT MAGNETIC MEDIA

SoftSide disks and tapes are duplicated by reliable, professional duplication services; bad copies are very rare. However, the trip through the mail occasionally wreaks havoc with sensitive magnetic media. If, after a reasonable number of tries and a careful check and cleaning of your equipment, you are not able to load a program from a tape or disk, please return it to us with an exact description of the problem. If we cannot duplicate the problem on our systems, we will advise you when we send the replacement copy.

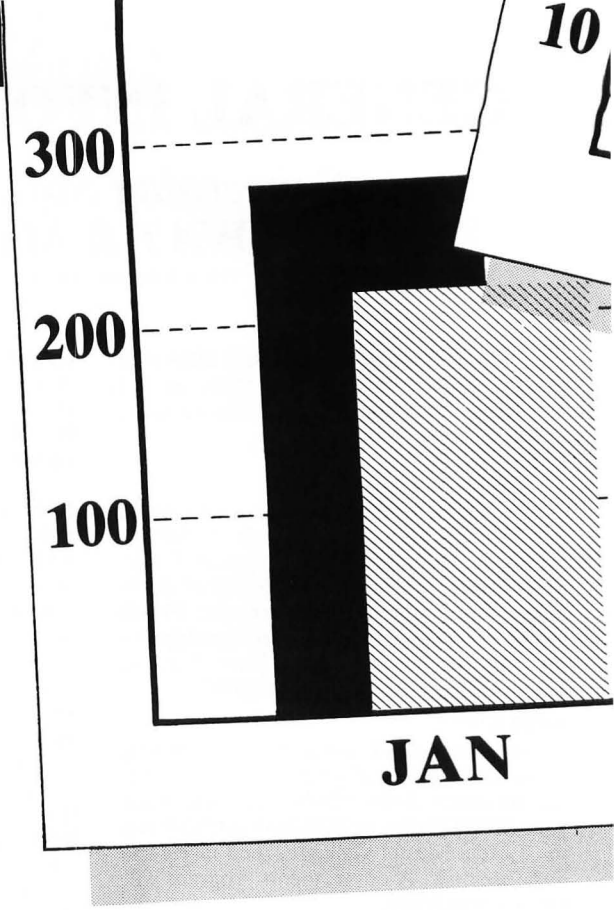
We use no copy-protection on our media. We urge you to make a backup copy of every disk or tape as soon as you receive it (and at the same time resist the urge to give copies to friends). Our replacement policy does not extend beyond 30 days.

APPLE™/SIDE page 28

TRS-80®/SIDE page 48

ATARI®/SIDE page 74

Hi-Res Character Generator



by Kerry Shetline

***Hi-Res Character Generator* is a utility program for the Apple II™. System requirements: 48K RAM, Applesoft, and one disk drive. It is included as the bonus program on Issue #33 Apple™ DV.**

There are many reasons why you might want to print text on the hi-res screen. You could label graphs, for instance, or, by controlling individual points on the screen, create special symbols or lowercase letters. For the programmer using BASIC, the only method generally available is to create a shape table with all the characters and use a subroutine that goes through a string, one character at a time, plotting shapes as it goes. This is usually slow and cumbersome. When I ran into a situation where I wanted to do hi-res printing, such an unappealing method didn't quite suit my tastes. I decided it was time for a little Machine Language.

The result was the *Shetline High Resolution Character Generator,* or *SHRCG* if you like abbreviations. This routine is 244 (decimal) bytes long, and is relocatable. The routine requires a character set to be loaded into memory, which occupies 1K. Once the routine is loaded and hooked up properly, all the output normally sent to the text screen will go to hi-res page 2. In fact, the graphics screen will, in almost every way, behave like the text screen. HTAB, VTAB, TAB, and SPC all function normally. The POKEs used to set the window on the text screen will set the same window in hi-res. The mild-mannered PRINT statement will send characters directly to the graphics display. Even the INVERSE statement works.

SHRCG also provides some new capabilities. Because the character set is in RAM, it can be changed to suit different needs. The character set provided with *SHRCG*, CHARSET, is like the normal Apple™ font. It does, however, include lower case letters, braces, the tilde, the backslash, and the grave accent. The FONT EDIT program provided will allow you to easily create your own character sets. Printing characters can even be assigned to control codes, with the exception of backspace, linefeed, and carriage return. You thus have access to as many as 125 different symbols.

Output can go to the screen in three different ways. The normal method causes a character being printed on the screen to replace anything that may occupy the print location. The output, however, can also be set to overprint the existing background, or to "exclusive-OR" with the background. This means that the dots which form a character will appear white when printed on a black area of the screen, or black when printed on a white area.

How to use *SHRCG*

First, BLOAD the file *SHRCG* into any convenient area in memory. Then BLOAD the character set, also wherever you would like. Remember that memory from $4000-5FFF (hi-res page 2) is off limits.

Now, set the print mode by doing one of the following POKEs:

```
POKE 25,0   : Set overprint mode
POKE 25,128 : Set normal mode
POKE 25,255 : Set "exclusive-OR"
              mode
```

Next, set up the pointer to the character set. By referencing character sets through a pointer, *SHRCG* can support the use of multiple fonts. Just doing a couple of POKEs might switch you from Cyrillic to English, for example.

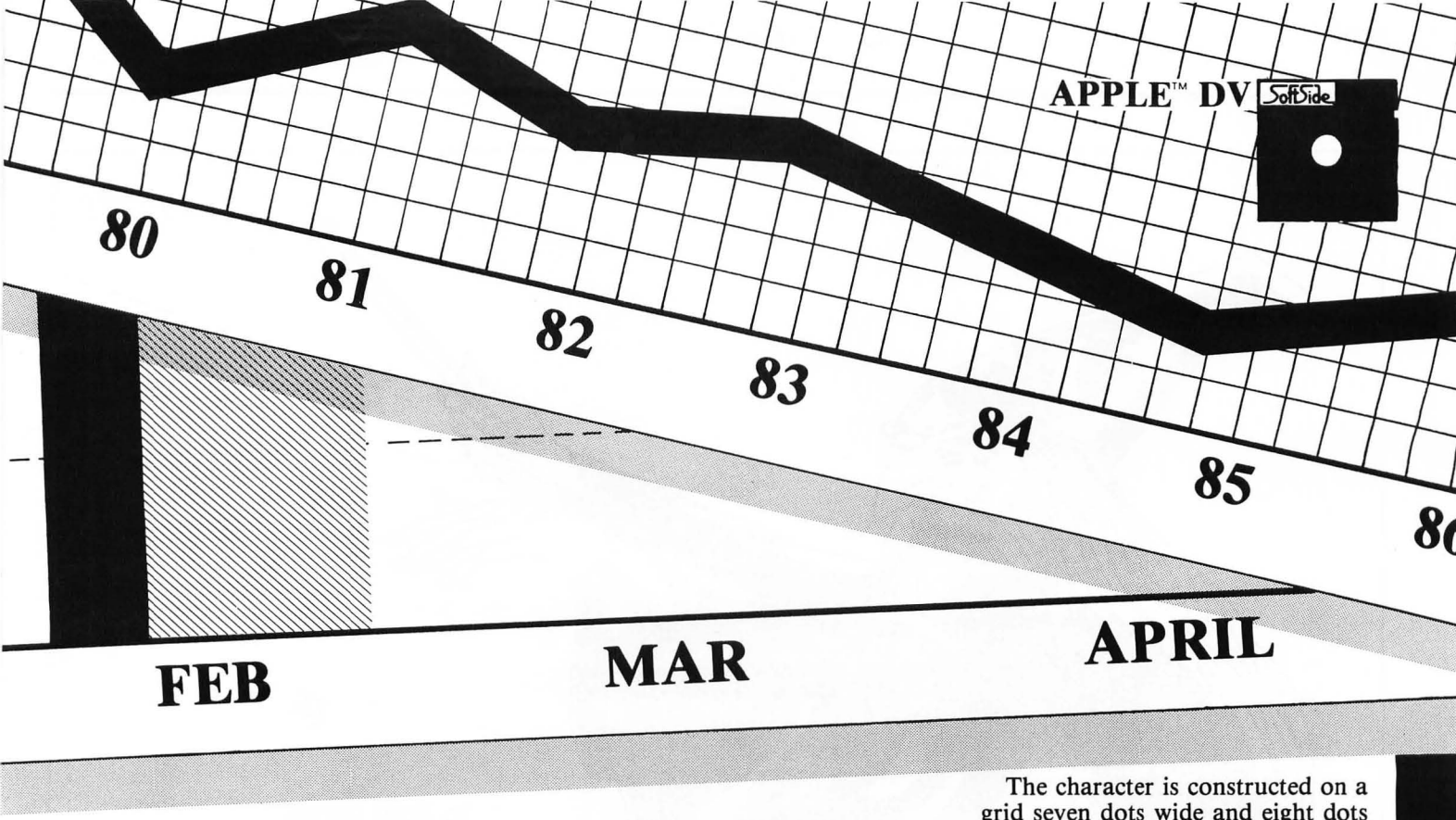
The pointer is at locations $FE and $FF. Set these two bytes to contain the low byte and the high byte of the address of the character set. In a program, the pointer would be set like this:

```
POKE 254,INT(address/256):
POKE 255,address-INT(address/
256)*256
```

You must now tell the Apple™ to send its output to *SHRCG*. Do this by changing the two memory locations, $36 and $37, that tell the Apple™ where to find the routine that will be used for output. These locations are usually called the "output hooks." They must be set to the low and high bytes of the address of *SHRCG*. This is just like setting the character set pointer:

```
POKE 54,INT(address/256):
POKE 55,address-INT(address/
256)*256
```

At this point, you would be finished if it weren't for DOS. DOS uses both the output hooks and another pair of locations, known as the "input hooks." Changing only the output hooks leaves DOS half connected, which can cause a few problems. The solution is to completely disconnect DOS. Do this by IN#0. Be sure to use the BASIC format for IN#0 rather than the DOS format. After that, CALL 1002 will get everything up and running and keep DOS happy.



Okay, this sounds a bit complicated. Let me clarify it with an example. Here is a typical routine to load and hook up *SHRCG*:

```
10  HGR2 : PRINT  CHR$ (4)"BLOAD SH
    RCG,A$2000": PRINT  CHR$ (4)"B
    LOAD CHARSET,A$2100": POKE 25,
    128: POKE 254,0: POKE 255,33:
    POKE 54,0: POKE 55,32: IN# 0:
    CALL 1002
```

The routine may be disconnected accordingly:

```
200  PRINT  CHR$ (4)"PR#0": TEXT
```

There are a few things that *SHRCG* won't do. The HOME statement will move the cursor, but won't clear the screen. The combination HOME: HGR will do the trick. If you set a screen window and wish to clear only the contents of that window, use successive PRINTs and scroll the window clean.

Another thing about windows — don't set a window that is one line tall. It won't work. Also, avoid the FLASH statement. The output is a kind of mangled inverse print.

Don't be alarmed if the bell won't sound. CHR$(7) produces a printed symbol like most of the other control codes.

Special Character Sets

You can create special character sets with the FONT EDIT program. To start the program, simply type "RUN FONT EDIT". The program will start with a prompt which asks if you want to load a font from disk. If you do not choose to load from disk, the character set will be entirely blank. Otherwise, you may load an old character set and work on that. If you wish to modify the character set provided, use the file name CHARSET.

You will then be shown a side-by-side comparison of your characters and the standard characters. After you hit a key to proceed, you will see a box on the screen, and the prompt "CHARACTER?". You may now do one of three things:

1) Type the character you wish to modify, and proceed to the edit mode.

2) Press ESC and jump to the options menu (described later).

3) Press CTRL-A, then enter the ASCII code of the character you wish to edit. If you enter -1, you will jump to the options menu.

If you have chosen to edit a character, you will now see a blinking cursor inside the box on the screen. The box will contain the chosen character if it was previously defined, or will be blank if it was not.

The character is constructed on a grid seven dots wide and eight dots high. However, when you are creating characters for text purposes, allow one dot of clearance on the left and right, and one dot of space either above or below. Points on the grid are reversed by pressing the space bar. Black points will become white, or white points will become black.

Use the I, J, K, and M keys to move the cursor. This will move the cursor up, left, right, and down, respectively. If you suddenly decide that you shouldn't be editing the character currently chosen, just press ESC to return to the character selection prompt. Press ESC a second time to go to the options menu.

While you are editing, you may press "C" to clear the character, or "R" to restore the character to what it was before the editing began. To complete the work on a character, press RETURN. This will put you in the options menu.

From the options menu you may:

1) Press E to edit another character.

2) Press S to display the entire character set.

3) Press Q to exit the program.

Before the program ends, you will be asked if you want to save the font you have just created. If you choose to do so, enter the file name you wish to use. If you have been editing the standard font, you should probably save the new version with a name other than CHARSET.

by Brent Iverson

***Shape Wizard* is a graphics utility for an Apple™ with Applesoft, 48K RAM, and one disk drive. It will also be provided on Apple™ CV.**

Constructing shape tables by hand is a boring, tedious, repetitive chore. And what do computers excel at? Boring, tedious, repetitive chores, of course. With this in mind, I set out to write a program that would take the drudgery out of creating shape tables. The result was *Shape Wizard*. This program lets you construct shapes quickly and easily, then save them to disk, if you so desire.

When you use *Shape Wizard*, it asks you in what scale you would like to construct your shape. The choices are: actual size, 3 times actual size, or 5 times actual size. If you choose 3 or 5 times actual size, the program asks you if you would like a grid of dots on the screen.

Now you are ready to start constructing your shape. You control the cursor with the keyboard, as shown below.

Diagram 1

SHAPE WIZARD

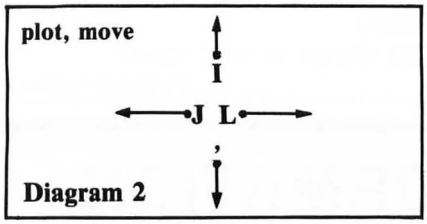


Diagram 2

Notice that your cursor will either "plot" or "no-plot" at its current location, and only *then* move in the direction indicated by the arrow. Also note that when you are constructing an actual size shape, your cursor is indistinguishable from any other filled square, as each square occupies only one dot. See the diagram below for a picture of the cursor and a filled square at each scale.

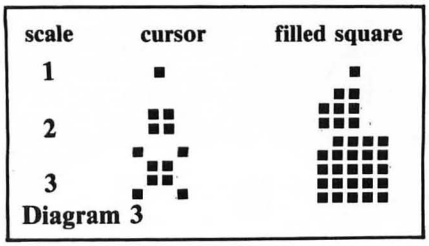


Diagram 3

If, while constructing a shape, you plot on a square, then later move back and plot on it again, the first plot is negated. This brings up a slight difference between the DRAW and XDRAW commands for drawing a shape. On a black background, with HCOLOR set white, there is no difference if there are no overlapping plots. If, however, you have overlapped plots, DRAW causes them to appear white, even if they were negated while you were constructing the shape; XDRAW causes them to appear just as *Shape Wizard* originally displayed them. To be safe, simply avoid overlapped plotting.

If you attempt to move your cursor off the screen or to make two consecutive "no plot, move up" moves (by pressing W twice), the program buzzes and says, "Illegal move!" The reason for this is documented on page 92 of the *Applesoft Reference Manual*. When your shape is compiled, these two moves *may* be interpreted as the end of your shape table. (Also see pages 92-97 of the manual for a clear, concise discussion of the mechanics involved in the construction of shape tables.)

When you are finished constructing your shape table on the screen, just press C. The program then compiles the movement codes, which it was storing as you made the shape, into the final shape; XDRAWs the final shape in the middle of the screen; and asks you if you want to save your shape on disk. Finally, it asks you if you want to construct another shape.

An Example of Using the Program

The first thing you see after the title page is this prompt:

```
What scale would you like to
plot in?

   1 = 1 * Size
   2 = 2 * Size
   3 = 3 * Size

      What scale: ■
```
Diagram 4

If you try to enter anything but 1, 2, or 3, the program displays a zero beneath the cursor, and waits for you to press something else. This time, press 3, which gives you the greatest amount of detail. Now answer "Y" when the program asks you if you want a grid. Drawing the grid takes about ten seconds. When the grid is complete, an X appears in the center of the screen. This is your cursor, and shows you your current location. At the bottom of the screen, this information appears: LIMIT: 24576, CURRENT ADDRESS: 16384. This tells you that you have slightly over 8000 bytes of memory left in which the program can store movement codes. The limit of 24576 keeps all stored data within the second hi-res page, an arbitrary limit. If you ever want to construct a shape with more than 8000 moves, you should be able to move the limit up 5000 to 10,000 bytes with no trouble. You can do this by changing the initial value of LIMIT in line 280. If you do this, you may also want to insert this line: 1005 BB=FRE(0). This makes sure that no string data comes down into your data area. This line does, however, slow down the program, as it causes garbage collection (string gathering).

Let's start by making a simple shape: a square. Make the square by entering the following commands: L L L I I I I J J J J , , , , L. Now press C to tell the program that the shape is ready for compilation into something that the computer can understand. The program asks if

you are really ready to compile, so answer "Y".

The bottom of the screen clears, and displays information about the progress of the compilation. For our simple shape, compilation takes only a few seconds; however, if your shape is exceptionally large, it does take the program a little while to get done. A shape of more than 8000 moves took the program well over five minutes to compile.

When the program is finished compiling, it will clear the screen and XDRAW your shape in the center of the screen.

It then asks you if you want to save the shape on disk. If you type "Y", it prompts you for a filename under which to save the shape, then saves the shape as a binary file. When it is done, it asks you if you want to construct another shape, and if you do, starts all over again. If not, it clears the screen, and ends.

Simple, isn't it? Soon, you'll be making large, complicated shapes quickly and easily, as your friends stare on in amazement. Chris Carroll's *Hi-Res Shape Combiner/Splitter* from *Nibble* (vol. 2, no. 5) makes an excellent companion to this program. Good luck, and have fun making shapes.

Variables

A$: Used for input from user.
AN$: Temporary string.
BA: Base address to store data.
BB: Temporary variable.
C1: Previous movement code.
CC: Current movement code.
CH: Change in position of cursor on hi-res screen with each movement.
CO: Current column of data. Used while making shape on screen.
CP: Current address into which to poke compiled results.
CU: Hi-res cursor shape.
EE: Temporary variable.
ER: Error code.
GD: If 1, then draw grid; if 0, then don't.
I: Temporary variable.
L: Numeric equivalent of L$.
L$: Left half of A$.
LI: Upper limit of data storage.
LM: Address at which to stop compiling.
ME: Current address into which to poke movement codes.
N$: Filename for saving shape.
NU: Temporary variable
PP: Used in compile routine. Current row of data being compiled.
PP%: Used to detect need to force string-gathering with the FRE(0) command.
Q: Numeric output for the routine at line 5000.
Q$: String input for the routine at line 5000.
R: Numeric equivalent of R$.
R$: Right half of A$.
RO: Current row of data. Used while making shape on screen.
RR: Current column of data being compiled.
SC: Scale in which to construct shape.
SQ: Shape of solid square.

continued on page 34

SoftSide DV, the magazine of the future, is here!

If your computer could pick a magazine, wouldn't it prefer one in its own language? Now there's one available. **SoftSide DV** is an enhancement of the **SoftSide** you have in your hands.

SoftSide DV contains not only the complete programs listed in every month's issue of **SoftSide**, but additional programs of every conceivable type, as well — multiple and Machine Language programs, modified languages, ongoing modular programs and software so extensive, it would take an entire issue of **SoftSide** just to print the code. Only the documentation for these programs will appear in **SoftSide** Magazine, **NOT** the code.

Feel as though you're missing something? You are! But, you needn't miss out on another issue. **SoftSide DV** is now available for Apple™, ATARI® and the TRS-80® . The cost to you — $125 for 12 magazines and 12 disks, packed with some of the best software available, all delivered to your home in the next year. For orders outside the USA, please add $36. For your convenience, we offer an installment payment plan for VISA and MasterCard holders: You pay only $32.50 per month for four months (a total of $130, which includes a $5 billing charge). Please use the special DV-CV bind-in card in this issue to order.

Computerists are offered the rare opportunity of marching into a new frontier. Advance to the front of the parade by subscribing to **SoftSide DV**, the magazine of the future, available today!

```
    SS SS SS SS SS SS SS SS SS SS SS
    SS                           SS
    SS      APPLESOFT BASIC      SS
    SS       'SHAPE WIZARD'      SS
    SS   AUTHOR: BRENT IVERSON   SS
    SS     COPYRIGHT (C) 1982    SS
    SS SOFTSIDE PUBLICATIONS, INC SS
    SS                           SS
    SS SS SS SS SS SS SS SS SS SS SS

10  ONERR  GOTO 20000
```

Make title and ask user how he wants the screen to look.

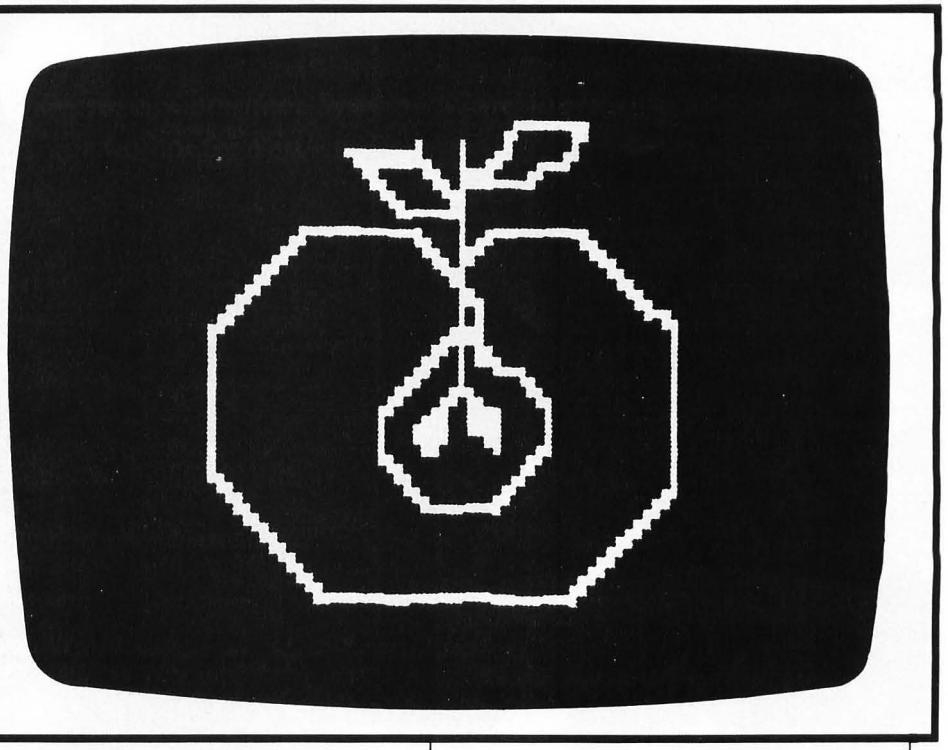
```
100  TEXT : HOME : INVERSE : POKE  -
     16368,0
110  PRINT "SHAPE WIZARD" SPC( 12)"
     BY BRENT IVERSON"
120  NORMAL
130  GOSUB 10000
140  VTAB 4: HTAB 1
150  POKE  - 16368,0
160  PRINT " WHAT SCALE WOULD YOU L
     IKE TO PLOT IN?"
170  PRINT :Q = 14
180  PRINT  SPC( Q)"  = 1 * SIZE"
190  PRINT  SPC( Q)"  = 3 * SIZE"
200  PRINT  SPC( Q)"  = 5 * SIZE"
210  INVERSE : FOR EE = 6 TO 8: VTAB
     EE: HTAB Q + 1: PRINT EE - 5;:
      NEXT : NORMAL
220  VTAB 10: HTAB 15: PRINT "WHAT
     SCALE:";: GET A$:SC =  VAL (A$
     ): PRINT SC
230  IF SC < 1 OR SC > 3 THEN 220
240 GD = 0: IF SC < 2 THEN 270
250  PRINT : PRINT "SHOULD THERE BE
      A GRID ";: GOSUB 4000
260  GET A$: PRINT A$: IF A$ = "Y" THEN
     GD = 1
```

Initialize variables.

```
270 CU = SC:SQ = SC + 3:CH =  VAL (
     MID$ ("135",SC,1))
280 BASE = 16384:LIMIT = 24576:ROW =
     1:COL = 3:X = 140:Y = 80
290  POKE BASE,1: POKE BASE + 1,0: POKE
     BASE + 2,4: FOR I = BASE + 3 TO
     BASE + 10: POKE I,0: NEXT
```

Set up the screen and the text window.

```
300  HGR : HCOLOR= 3
310  IF GD = 0 THEN 380
```



```
320  IF SC = 2 THEN SX = 2:SY = 2
330  IF SC = 3 THEN SX = 0:SY = 0
340  FOR YY = SY TO 159 STEP CH
350  FOR XX = SX TO 279 STEP CH
360  HPLOT XX,YY
370  NEXT : NEXT
380  VTAB 21: HTAB 1: PRINT "LIMIT:
     ";LIMIT;: HTAB 19: PRINT "CURR
     ENT ADDRESS:";BASE;
390  XDRAW CU AT X,Y
```

Main Loop.

```
1000  VTAB 21: HTAB 40: GET A$
1010 XO = X:YO = Y:C1 = CC
1020  IF A$ = "W" THEN Y = Y - CH:C
      C = 0: GOTO 1120
1030  IF A$ = "A" THEN X = X - CH:C
      C = 11: GOTO 1120
1040  IF A$ = "D" THEN X = X + CH:C
      C = 1: GOTO 1120
1050  IF A$ = "X" THEN Y = Y + CH:C
      C = 10: GOTO 1120
1060  IF A$ = "I" THEN Y = Y - CH:C
      C = 100: GOTO 1120
1070  IF A$ = "J" THEN X = X - CH:C
      C = 111: GOTO 1120
1080  IF A$ = "L" THEN X = X + CH:C
      C = 101: GOTO 1120
1090  IF A$ = "," THEN Y = Y + CH:C
      C = 110: GOTO 1120
1100  IF A$ = "C" THEN  VTAB 23: HTAB
      3: PRINT "ARE YOU READY TO COM
      PILE ";: GOSUB 4000: GET AN$: HTAB
      1: NORMAL : PRINT  SPC( 39);: IF
      AN$ = "Y" THEN 3000
1110  GOTO 1000
```

Check for running out of space or exceeding the screen boundaries.

```
1120 NU = BASE + 3 * ROW + COL: VTAB
     21: HTAB 35: PRINT NU;: IF NU >
      = LIM THEN  GOTO 3000
1130  IF (X < 0 OR X > 279) OR (Y <
     0 OR Y > 159) OR (C1 = CC AND
     CC = 0) THEN X = XO:Y = YO:CC =
     C1: VTAB 22: HTAB 13: PRINT "<
     ILLEGAL MOVE>";: FOR BB = 1 TO
     50:PP =  PEEK ( - 16336): NEXT
     : HTAB 13: PRINT  SPC( 14): GOTO
     1000
```

Find out what "row" and "column" we're on.

```
1140  IF COL = 3 OR (COL = 2 AND CC
      <  > 0) THEN  GOSUB 2000:COL =
     COL - 1: GOTO 1170
1150  IF (COL = 1 AND (CC = 0 OR CC
      > = 100)) OR (COL = 2 AND CC
      = 0) THEN ROW = ROW + 1:COL =
     3: GOSUB 2000:COL = COL - 1: GOTO
     1170
1160  IF COL = 1 AND (CC > 0 AND CC
      < 100) THEN  GOSUB 2000:ROW =
     ROW + 1:COL = 3
```

Update the screen.

```
1170  IF CC >  = 100 THEN  XDRAW SQ
      AT XO,YO
```

```
1180  XDRAW CU AT XO,YO
1190  XDRAW CU AT X,Y
1200  GOTO 1000
```

Put the movement code into memory.

```
2000 MEM = BASE + 3 * ROW + COL: POKE
     MEM,CC
2010  IF COL = 3 THEN  FOR WW = MEM
      + 1 TO MEM + 10: POKE WW,0: NEXT
2020  RETURN
```

Compile the movement codes into the *final shape* table.

```
3000 ZERO$ = "0000":CP = BASE + 4
3010 BB =  FRE (0)
3020 LM = BASE + 3 * ROW + COL + 3
3030  HTAB 1
3040  HOME : VTAB 21: HTAB 14: INVERSE
     : PRINT "<COMPILING>": NORMAL

3050  VTAB 23: HTAB 1: PRINT "BYTES
      TO COMPUTE:";LM - BA;: HTAB 2
     5: PRINT "SO FAR:0"
3060  FOR PP = BASE + 4 TO LM STEP
     3
3070 A$ = ""
3080  FOR RR = 0 TO 2
3090 T$ =  STR$ ( PEEK (PP + RR))
3100  IF  LEN (T$) < 3 THEN T$ =  LEFT$
     (ZERO$,3 -  LEN (T$)) + T$
3110  IF RR = 0 THEN T$ =  RIGHT$ (
     T$,2)
3120 A$ = A$ + T$
3130  NEXT
3140 L$ =  LEFT$ (A$,4):R$ =  RIGHT$
     (A$,4)
3150 Q$ = L$: GOSUB 5000:L = Q
3160 Q$ = R$: GOSUB 5000:R = Q
3170  POKE CP,((16 * L) + R)
3180 CP = CP + 1
3190  VTAB 23: HTAB 32: PRINT PP -
     BA
3200 PP% = PP / 50: IF PP% = PP / 5
     0 THEN BB =  FRE (0)
3210  NEXT
```

Display the final shape and ask if it should be saved.

```
3220  POKE 232,0: POKE 233,64
 3230  HGR
3240  XDRAW 1 AT 140,80: POKE  - 16
     368,0
3250  HOME : VTAB 21: PRINT "SHAPE
      ENTERED FROM ";BASE;" TO ";CP;
      "."
3260  PRINT "SAVE TO DISK? ";: GOSUB
      4000: GET A$: IF A$ <  > "Y" THEN
      3290
3270  VTAB 23: HTAB 1: INPUT "ENTER
       NAME TO SAVE AS: ";N$
3280  PRINT  CHR$ (4);"BSAVE ";N$;"
      ,A";BASE;",L";CP - BASE
3290  HOME : VTAB 21: PRINT "MAKE A
      NOTHER SHAPE ";: GOSUB 4000: GET
      A$: IF A$ <  > "Y" THEN  TEXT
      : HOME : POKE 216,0: END
3300  GOTO 100
```

Subroutine for "Y/N" prompt.

```
4000  NORMAL : PRINT "(";: INVERSE
      : PRINT "Y";: NORMAL : PRINT "
OR ";: INVERSE : PRINT "N";: NORMAL :
PRINT "):";: RETURN
```

Find the number that the four-character string Q$ "represents" in hex.

```
5000 Q =  VAL ( RIGHT$ (Q$,1)) + 2 *
      VAL ( MID$ (Q$,3,1)) + 4 *  VAL
     ( MID$ (Q$,2,1)) + 8 *  VAL ( LEFT$
     (Q$,1))
5010  RETURN
```

POKE in the shape table that this program uses.

```
10000  POKE 232,0: POKE 233,3
10010  RESTORE
10020  FOR I = 768 TO 837
10030  READ V: POKE I,V
10040  NEXT
10050  SCALE= 1: ROT= 0: HCOLOR= 3
10060  RETURN
10070  DATA  6,0,14,0,19,0,26,0,37,
      0,42,0
10080  DATA  51,0,5,0,101,101,0,17,
      53,39,0,110,0,111,17,77,23,55,
      117,223
10090  DATA  7,0,101,101,0,5,0,0,0,
      0,41,62,55,45,5,0,11,10,10,41
10100  DATA  45,62,63,55,45,45,62,6
      3,55,45,45,4,0,48,45,45,0,0
```

Error control routine.

```
20000 ER =  PEEK (222)
20010  HOME
20020  VTAB 21: HTAB 1
20030  FOR BB = 1 TO 50:TT =  PEEK
      ( - 16336): NEXT
20040  IF ER = 4 THEN  PRINT "  YO
      UR DISKETTE IS WRITE-PROTECTED
      !": GOTO 20120
20050  IF ER = 8 THEN  PRINT  SPC(
      16);"I/O ERROR!": GOTO 20120
20060  IF ER = 9 THEN  PRINT  SPC(
      15);"DISK FULL!": GOTO 20120
20070  IF ER = 11 THEN  PRINT  SPC(
      11);"ILLEGAL FILE NAME!": PRINT
       SPC( 6);"<PRESS ANY KEY TO CO
      NTINUE>";: POKE  - 16368,0: GET
      AN$: HOME : GOTO 3270
20080 LI =  PEEK (218) +  PEEK (219
      ) * 256: POKE 216,0: TEXT
20090  IF ER = 16 THEN  PRINT "SYNT
      AX ERROR IN LINE ";LI: END
20100  IF ER = 255 THEN  PRINT "USE
      R INTERRUPT IN LINE ";LI: END

20110  PRINT "UNEXPECTED ERROR (";E
      R;") IN LINE ";LI: END
20120  PRINT "PRESS ";: INVERSE : PRINT
      "RETURN";: NORMAL : PRINT " WH
      EN READY TO ATTEMPT": PRINT "A
      NOTHER SAVE, OR ANY OTHER KEY
      IF": PRINT "YOU DON'T WANT TO
      SAVE YOUR SHAPE.";
20130  POKE  - 16368,0: GET A$
20140  IF A$ <  > CHR$ (13) THEN 3
      290
20150  HOME : VTAB 21: PRINT  SPC(
      7);"<ATTEMPTING ANOTHER SAVE>"

20160  GOTO 3280
```

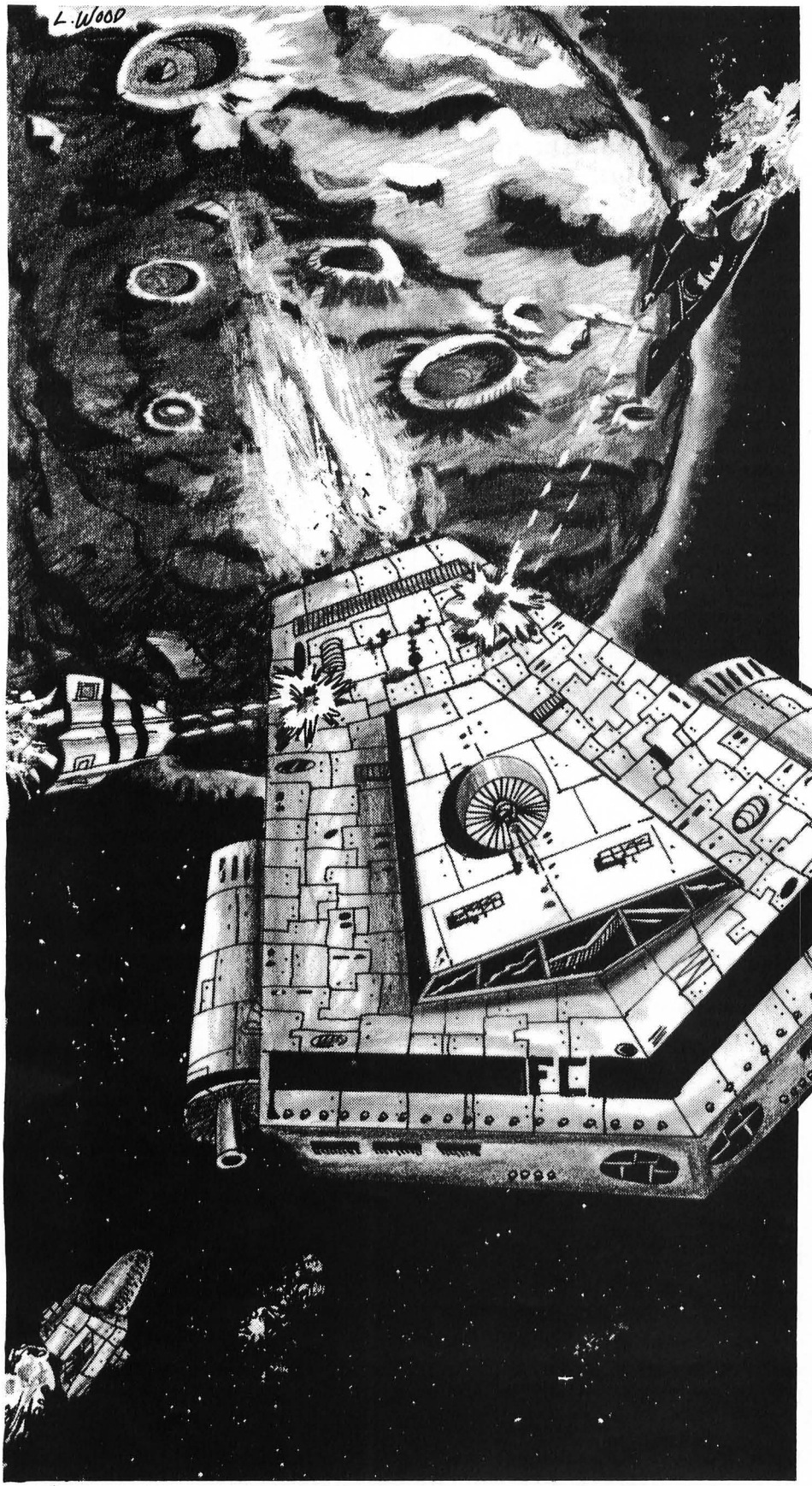
APPLE™ SWAT TABLE FOR:
SHAPE WIZARD

LINES	SWAT CODE	LENGTH
10 - 200	JB	241
210 - 320	HW	370
330 - 1040	JM	272
1050 - 1150	XE	556
1160 - 3030	ZG	249
3040 - 3150	FY	271
3160 - 3270	ZH	298
3280 - 10050	ZQ	306
10060 - 20060	BZ	411
20070 - 20160	XU	486

DEFENSE

by Greg Schroeder

Apple™ translation by Jordan Drachman.



***Defense* is a 1 or 2 player arcade type game for an Apple™ with Applesoft and 32K RAM. This version is the Translation Contest winner for issue 33.**

Prepare for battle as you are the pilot of a starship and evil aliens have invaded the moon. Your mission — destroy the attacking aliens before they launch their fleets to conquer the Earth.

The aliens attack in waves, with up to three aliens on the screen at a time. There are two types of alien: Drones (worth 20 to 80 points), and Smiling Blobs (100 to 300 points). After each wave, you will receive 500 points for each ship you have remaining. After four waves, you will receive an extra ship.

You may use either the keyboard or a joystick to control your ship.

After thrusting several times, your ship will begin to gain speed.

Program Notes

Defense was originally written for the ATARI® computers, and utilized a redefined character set. Since the Apple™does not have this ability, I had to find a way to translate this program using high-resolution graphics. I decided to use block shapes. I first learned about block shapes from the article "Understanding Hi-Res Graphics" by Roy Spurlock (*Apple Orchard*, Fall 1980), which was my source for the shapes I used for letters and numbers in *Defense.*

Here is a brief description of block shapes. The Apple's™ hi-res screen does not map linearly into memory. In other words, the first memory location in one line of the hi-res screen does *not* immediately follow the last memory location in the previous line. This is inconvenient for most Machine Language programs, but not for block shapes. The memory is arranged so that the hi-res screen is divided into 24 horizontal blocks of eight hi-res lines. The memory for any given line in a block is exactly 1024 bytes (1K) from the memory for the previous line. This makes it easy to form shapes by lighting up different bits in each line.

The ampersand (&) command calls a Machine Language routine that is set up in line 10020. The command &S,X,Y tells

the routine the number of the shape to draw (S), and the x and y coordinates (X,Y) of the spot where it should draw the shape. Memory locations 1013 to 1015 (decimal) must contain a Machine Language jump to the beginning of this routine so that the ampersand command will execute it. Locations 234 and 235 (decimal) contain the starting location of the shape table.

The Machine Language routine has another function as well. The command &0,L,S will make the shapes in line L move S spaces to the left. Line 105 uses this command to move the moon's surface.

Variables

A(i,j): Invisible positions of aliens along moon surface.

A,D,I,I$,J,N,X,X1,Y1: Miscellaneous variables.

AL(i): Number of aliens in each player's current attack wave.

A$: Moon surface picture string.

B(i), C(i): X and Y positions of aliens on screen.

D(i): Y movement of alien on screen.

E(i): Type of alien on screen.

F(1), F(2): Number of aliens in each player's wave. Used to reset AL(i).

F(3), F(4): Number of attack waves destroyed by each player.

GS: Ground speed.

PL: Number of current player.

PP: Total number of players.

R(i): Number of aliens along moon's surface that appear on the screen.

S0, S1, S: Joystick and keyboard values.

SC(i): Each player's score.

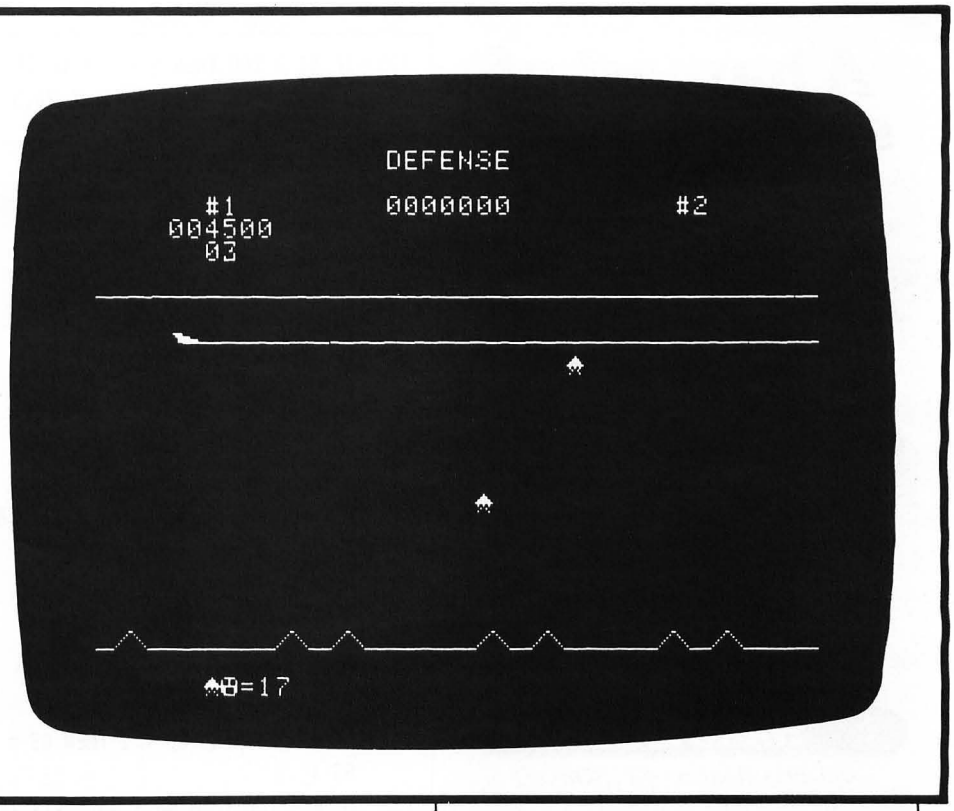
SH(i): Remaining ships for each player.

TT: Counter for alien movement on screen.

XC: Counter for alien movement along moon surface.

Y: Y-coordinate of player's ship (X defaults to 4).

Y1: Temporary storage for Y.




```
 SS SS SS SS SS SS SS SS SS SS SS
 SS                           SS
 SS      APPLESOFT BASIC      SS
 SS         'DEFENSE'         SS
 SS   AUTHOR: GREG SCHROEDER  SS
 SS  TRANSL: JORDAN DRACHMAN  SS
 SS     COPYRIGHT (C) 1982    SS
 SS SOFTSIDE PUBLICATIONS, INC SS
 SS                           SS
 SS SS SS SS SS SS SS SS SS SS SS
```

Program initialization.

```
5  LOMEM: 24576

10  GOSUB 10000

15  DIM A(2,41),B(4),C(4),D(4),E(4)
    ,F(4),SH(2),SC(2),R(4),AL(2):S
    C(1) = 0:SC(2) = 0

20  FOR I = 0 TO 4:B(I) =  - 1:C(I)
     = 0:D(I) = 0:E(I) = 0:R(I) =
    0: NEXT I

25 I = SC(1) * (SC(1) > SC(2)) + SC
    (2) * (SC(2) > SC(1)):HS = HS *
    (HS > I) + I * (I > HS)

30  FOR I = 1 TO 2: FOR J = 1 TO 20
    :A(I,J) =  INT ( RND (1) * 16)
    : NEXT J,I

40 SH(1) = 3:SH(2) = 3:SC(1) = 0:SC
    (2) = 0:AL(1) = 20:AL(2) = 20:
    F(1) = 20:F(2) = 20:F(3) = 0:F
    (4) = 0:XC = 0

50  GOSUB 6000: GOTO 100
```

Subroutine to print numbers to the hi-res screen.

```
60  IF N > = 10 ^ D THEN A = N: FOR
    J = 1 TO D:A =  INT (A / 10): NEXT
    J:N = N - A * (10 ^ D): IF N <
     > 0 THEN N = N + 1

70  FOR J = 1 TO D:A = N - 10 *  INT
    (N / 10) + 15: & A,X1,Y1:X1 =
    X1 - 1:N =  INT (N / 10):A = 0
    : NEXT J: RETURN
```

Main program loop.

```
100 TT = TT + 1: IF TT = 4 THEN TT =
    1

105  IF GS > 1 THEN  & 0,21, INT (6
    S)

106  IF KB = 1 THEN 148

110 S0 =  PDL (0):S1 =  PDL (1): IF
    S0 > = 55 AND S0 < = 200 AND
    S1 > = 55 AND S1 < = 200 THEN
    145

120 Y1 = Y: IF S1 < 55 AND Y > 7 THEN
    Y = Y - 1
```

```
125  IF S1 > 200 THEN Y = Y + 1: IF
     Y = 21 THEN  & 26,4,Y1: & 26,5
     ,Y1: GOTO 3000
130  & 26,4,Y1: & 26,5,Y1: & 1,4,Y:
      & 2,5,Y
135  IF S0 > 200 AND GS < 2 THEN GS
      = GS + .05
140  IF S0 < 55 THEN GS = GS - .05:
      IF GS < 0 THEN GS = 0
145  IF  PEEK ( - 16287) > 127 THEN
     160
147  GOTO 200
148 S =  PEEK ( - 16384): POKE  - 1
     6368,0: IF S < 128 THEN 154
149 Y1 = Y: IF S = 193 AND Y > 7 THEN
     Y = Y - 1
150  IF S = 218 THEN Y = Y + 1: IF
     Y = 21 THEN  & 26,4,Y1: & 26,5
     ,Y1: GOTO 3000
151  & 26,4,Y1: & 26,5,Y1: & 1,4,Y:
      & 2,5,Y
152  IF S = 149 AND GS < 2 THEN GS =
     GS + .1
153  IF S = 136 THEN GS = GS - .1: IF
     GS < 0 THEN GS = 0
154  IF S = 160 THEN 160
155  GOTO 200
```

Fire laser and check to see if all aliens are dead.

```
160  POKE 6,48: POKE 7,255: POKE 8,
     250: CALL 975: FOR I = 6 TO 39
     : & 7,I,Y: NEXT I: FOR I = 1 TO
     3: IF B(I) > 5 AND  INT (C(I))
      = Y THEN 4000
162  NEXT I
165  FOR I = 6 TO 39: & 26,I,Y: NEXT
     I
170  IF AL(PL) < 1 THEN 4100
```

Move aliens and check for collisions.

```
200 XC = XC + 1: IF XC = F(PL) + 1 THEN
     XC = 1
210  IF A(PL,XC) >  - 5 THEN A(PL,X
     C) = A(PL,XC) - GS: IF A(PL,XC
     ) <  = 0 THEN 1000
220  IF B(TT) =  - 1 THEN 300
230  & 26, INT (B(TT)), INT (C(TT))
     :B(TT) = B(TT) - GS - .5: IF B
     (TT) < 0 THEN B(TT) =  - 1:A(P
     L,R(TT)) = 15: GOTO 300
240  IF E(TT) = 2 THEN 2000
250 C(TT) = C(TT) + D(TT): IF C(TT)
      < 7 OR C(TT) > 20 THEN C(TT) =
     C(TT) - D(TT):D(TT) =  - D(TT)

255  IF ( INT (B(TT)) = 4 OR  INT (
     B(TT)) = 5) AND ( INT (C(TT)) =
     Y) THEN 3000
260  IF E(TT) = 1 THEN  & 3, INT (B
     (TT)), INT (C(TT)): GOTO 300
270  & 4, INT (B(TT)), INT (C(TT))
300  GOTO 100
```

Place a new alien somewhere on the screen.

```
1000  FOR I = 1 TO 3: IF B(I) =  -
     1 THEN 1010
1005  NEXT I:A(PL,XC) = 15: GOTO 22
     0
1010 C(I) =  INT ( RND (1) * 14) +
     7:D(I) =  RND (1) + .1 - 1.5 *
     ( RND (1) < .5)
1020 E(I) = 2 +  SGN ( INT ( RND (1
     ) * 3) - 2):R(I) = XC:A(PL,XC)
      =  - 10:B(I) = 39: IF E(I) =
     2 THEN B(I) = 20
1025  POKE 6,80: POKE 7,48: POKE 8,
     80: CALL 975: GOTO 220
```

Special routine to move the "smiling blob" alien.

```
2000  IF C(TT) < Y THEN D(TT) = 1: GOTO
     250
2010  IF C(TT) >  = Y THEN D(TT) =
      - 1: GOTO 250
```

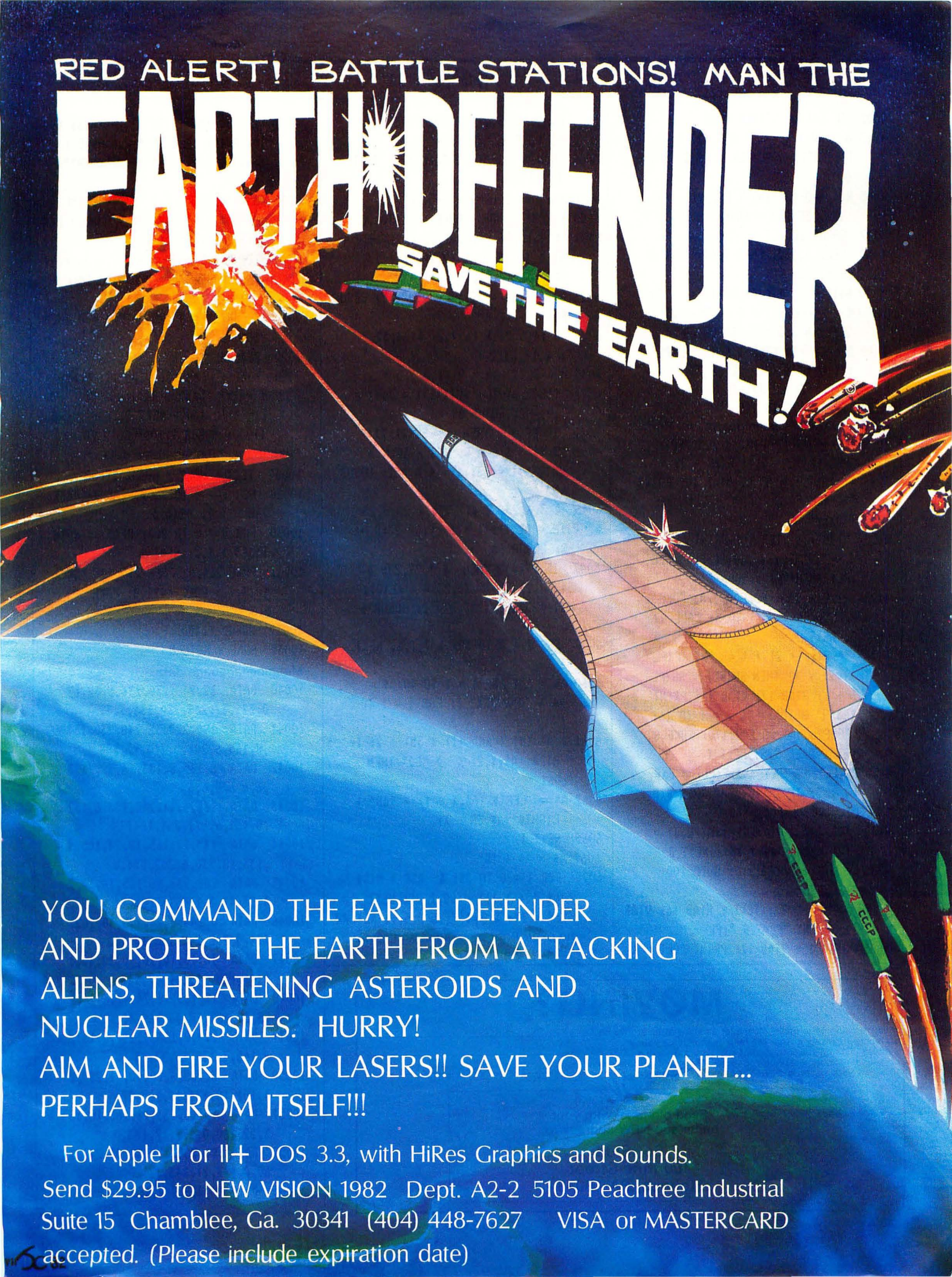
Player has been hit. Blow up ship and initialize for new turn.

```
3000  FOR I = 1 TO 3: IF B(I) > 0 THEN
     A(PL,R(I)) = 15:B(I) =  - 1: NEXT
     I
3010  & 1,4,Y: & 2,5,Y
3020  FOR I = 1 TO 5: FOR J = 1 TO
     2
3025  IF J = 2 THEN 3040
3030  & 26,4,Y: & 26,5,Y
3035  & 7,3,Y: & 7,6,Y: & 5,6,Y - 1
     : & 5,3,Y + 1: & 6,3,Y - 1: &
     6,6,Y + 1: GOTO 3050
3040  & 1,4,Y: & 2,5,Y
3045  & 26,3,Y: & 26,6,Y: & 26,6,Y -
     1: & 26,3,Y + 1: & 26,3,Y - 1:
      & 26,6,Y + 1
3050  POKE 6,69: POKE 7,J * 72: POKE
     8,160: CALL 975: NEXT J,I
3055  & 26,4,Y: & 26,5,Y
3060  HGR
3062 GS = 0:SH(PL) = SH(PL) - 1
3063  IF SH(PL) = 0 THEN  GOSUB 700
     0: FOR I = 1 TO 5000: NEXT I
3065  IF SH(PL - 1 + 2 * (PL = 1)) >
     0 THEN PL = PL + 1: IF PL > PP
      THEN PL = 1
3070  GOSUB 5000: FOR I = 1 TO 1000
     : NEXT I: GOSUB 6000:XC = 0:GS
      = 0: GOTO 100
```

Alien has been hit. Add his point value to the score and adjust the counters.

```
4000 X1 = B(I) + (B(I) <  > 39):Y1 =
      INT (C(I)):D = 2 + (E(I) = 2)
```

continued on page 40


RED ALERT! BATTLE STATIONS! MAN THE
EARTH DEFENDER
SAVE THE EARTH!
YOU COMMAND THE EARTH DEFENDER
AND PROTECT THE EARTH FROM ATTACKING
ALIENS, THREATENING ASTEROIDS AND
NUCLEAR MISSILES. HURRY!
AIM AND FIRE YOUR LASERS!! SAVE YOUR PLANET...
PERHAPS FROM ITSELF!!!
For Apple II or II+ DOS 3.3, with HiRes Graphics and Sounds.
Send $29.95 to NEW VISION 1982 Dept. A2-2 5105 Peachtree Industrial
Suite 15 Chamblee, Ga. 30341 (404) 448-7627 VISA or MASTERCARD
accepted. (Please include expiration date)

```
4002  IF E(I) = 2 THEN 4005
4003 N = 20 * ( INT ( RND (1) * 4) +
     1):SC(PL) = SC(PL) + N: GOTO 4
     007
4005 N = 100 * ( INT ( RND (1) * 3)
      + 1):SC(PL) = SC(PL) + N
4007  GOSUB 60
4010  POKE 6,112: POKE 7,64: POKE 8
     ,80: CALL 975
4030 B(I) =  - 1:A(PL,R(I)) =  - 5:
     N = SC(PL):X1 = 9 + (PL = 2) *
     26:Y1 = 3:D = 6: GOSUB 60
4040 AL(PL) = AL(PL) - 1:N = AL(PL)
     :X1 = 10 + (PL = 2) * 23:Y1 =
     23:D = 2: GOSUB 60
4050  NEXT I: GOTO 165
```

Attack wave has been destroyed. Print message and initialize for next wave.

```
4100  HGR :F(PL + 2) = F(PL + 2) +
     1: TEXT : HTAB 8: VTAB 10: PRINT
     "ATTACK WAVE ";F(PL + 2);" DES
     TROYED"
4120  HTAB 13: VTAB 14: PRINT "BONU
     S - ";SH(PL) * 500:SC(PL) = SC
     (PL) + SH(PL) * 500
4123  POKE 6,255: POKE 7,170: POKE
     8,255: FOR I = 1 TO 5: CALL 97
     5: NEXT I
4124  IF F(PL + 2) / 4 =  INT (F(PL
      + 2) / 4) THEN SH(PL) = SH(PL
     ) + 1: HTAB 14: VTAB 16: PRINT
     "EXTRA SHIP": FOR I = 1 TO 16:
      POKE 6,50: POKE 7,170: POKE 8
     ,255: CALL 975: NEXT I
4125 AL(PL) = F(PL) + 5:F(PL) = F(P
     L) + 5: IF F(PL) > 40 THEN F(P
     L) = 40:AL(PL) = 40
4130  FOR J = 1 TO AL(PL):A(PL,J) =
      INT ( RND (1) * 16): NEXT J
4135  GOSUB 6000:GS = 0: GOTO 100
```

"Get Ready" routine.

```
5000  HGR : HOME : TEXT : HTAB 16: VTAB
     11: PRINT "PLAYER ";PL
5005  HTAB 15: VTAB 12: PRINT "GET
     READY!": POKE 6,255: POKE 7,17
     0: POKE 8,255
5006  FOR I = 1 TO 5: CALL 975: NEXT
     I: RETURN
```

Initialize and draw the screen.

```
6000 : HOME : HGR : POKE  - 16302,0
6020  & 9,16,0: & 10,17,0: & 11,18,
     0: & 10,19,0: & 12,20,0: & 13,
     21,0: & 10,22,0
6025 N = HS:X1 = 22:Y1 = 2:D = 7: GOSUB
     60
6026  & 14,6,2: & 16,7,2: & 14,32,2
     : & 17,33,2
6030 N = SC(1):X1 = 9:Y1 = 3:D = 6:
      GOSUB 60:N = SH(1):X1 = 7:Y1 =
     4:D = 2: GOSUB 60
6040  IF PP = 2 THEN N = SC(2):X1 =
     35:Y1 = 3:D = 6: GOSUB 60:N =
     SH(2):X1 = 33:Y1 = 4:D = 2: GOSUB
     60
6045  & 3,6,23: & 4,7,23: & 25,8,23
     :N = AL(1):X1 = 10:Y1 = 23:D =
     2: GOSUB 60
6047  IF PP = 2 THEN  & 3,29,23: &
     4,30,23: & 25,31,23:N = AL(2):
     X1 = 33:Y1 = 23:D = 2: GOSUB 6
     0
6050  FOR I = 0 TO 39: & 7,I,6: NEXT
     I
6060 A$ = "": FOR I = 1 TO 40:A$ =
     A$ + "D": NEXT I: FOR I = 1 TO
     7
6061 X =  INT ( RND (1) * 35) + 2: IF
      MID$ (A$,X,2) <  > "DD" THEN
     6061
6062 A$ =  LEFT$ (A$,X - 1) + "AB" +
      RIGHT$ (A$,38 - X)
6065  NEXT I
6067 A$ = A$ + "D"
6070  FOR I = 0 TO 39: &  ASC ( MID$
     (A$,I + 1,1)) - 60,I,21: NEXT
     I:Y = 14: & 1,4,Y: & 2,5,Y: RETURN
```

Game over. Print messages and restart if both players are done.

```
7000  HGR : HOME : TEXT : HTAB 16: VTAB
     10: PRINT "PLAYER ";PL: HTAB 1
     5: VTAB 12: PRINT "GAME OVER!"

7005 I$ = "SCORE = " +  STR$ (SC(PL
     )): VTAB 14: HTAB 20 -  INT ( LEN
     (I$) / 2): PRINT I$
7007  FOR I = 100 TO 1 STEP  - 1: POKE
     6,50: POKE 7,I: POKE 8,I: CALL
     975: NEXT I
7010  IF SH(1) = 0 AND SH(PP) = 0 THEN
     7050
7020  RETURN
7050  HOME : HTAB 15: VTAB 12: FLASH
     : PRINT "GAME OVER": NORMAL : FOR
     I = 1 TO 3000: NEXT I: POP : GOSUB
     11000: GOSUB 11100: GOTO 20
```

Poke in machine-language data.

```
10000  TEXT : HOME : GOSUB 11000
10020  FOR I = 0 TO 230: READ J: POKE
     768 + I,J: NEXT I
10030  FOR I = 0 TO 207: READ J: POKE
     16384 + I,J: NEXT I
10031  POKE 234,0: POKE 235,64: POKE
     1013,76: POKE 1014,0: POKE 101
     5,3
10032  GOSUB 11100
10035  HOME : RETURN
```

Machine-language data.

```
10040  DATA  32,248,230,134,6,32,19
     0,222,32,248,230,134,7,32
10041  DATA  190,222,32,248,230,134
     ,8,165,6,201,0,240,97,198
10042  DATA  6,165,8,133,16,32,73,3
     ,6,6,6,6,6,6
10043  DATA  165,6,41,248,133,6,162
     ,8,164,6,177,234,164,7
10044  DATA  145,17,165,18,24,105,4
     ,133,18,230,6,202,224,0
10045  DATA  208,234,96,165,16,74,7
     4,74,41,3,170,169,0,224
10046  DATA  0,240,8,24,105,40,202,
     224,0,208,248,133,17,165
10047  DATA  16,10,10,10,10,10,10,1
     0,41,128,24,101,17,133
10048  DATA  17,165,16,74,41,3,24,1
     05,32,133,18,96,169,40
10049  DATA  229,8,133,9,169,0,133,
     6,165,7,133,16,32,73
10050  DATA  3,165,6,10,10,41,252,2
     4,101,18,133,18,165,17
10051  DATA  101,8,133,21,165,18,13
     3,22,160,0,177,17,153,0
10052  DATA  80,200,196,8,208,246,1
     60,0,177,21,145,17,200,196
10053  DATA  9,208,247,162,0,189,0,
```

```
     80,145,17,200,232,192,40
10054 DATA 208,245,230,6,165,6,20
     1,8,208,184,96,164,6,166
10055 DATA 7,173,48,192,202,208,2
     50,136,240,10,166,8,202,208
10056 DATA 253,230,8,76,209,3,96
10060 DATA 0,60,120,112,112,0,0,0
     ,0,0,3,31,127,0,0,0
10061 DATA 0,8,28,62,127,42,20,34
     ,0,62,73,73,127,99,99,62
10062 DATA 0,64,32,16,8,4,2,1,0,1
     ,2,4,8,16,32,64
10063 DATA 0,0,0,0,127,0,0,0,0,0,
     0,0,0,0,0,127
10064 DATA 0,30,34,34,34,34,34,30
     ,0,62,2,2,30,2,2,62
10065 DATA 0,62,2,2,30,2,2,2,0,34
     ,34,38,42,50,34,34
10066 DATA 0,28,34,2,28,32,35,28,
     0,20,20,62,20,20,62,20
10067 DATA 0,28,34,50,42,38,34,28
     ,0,8,12,8,8,8,8,28
10068 DATA 0,28,34,32,24,4,2,62,0
     ,62,32,16,24,32,34,62
10069 DATA 0,16,24,20,18,62,16,16
     ,0,62,2,30,32,32,34,28
10070 DATA 0,56,4,2,30,34,34,28,0
     ,62,32,16,8,4,4,4
10071 DATA 0,28,34,34,28,34,34,28
     ,0,28,34,34,60,32,16,14
10072 DATA 0,0,0,62,0,62,0,0,0,0,
     0,0,0,0,0,0
```

Title display routine.

```
11000 TEXT : HOME : PRINT TAB( 16
     );"DEFENSE": PRINT : PRINT : PRINT
     : PRINT "    EVIL ALIENS FROM
     BEYOND EARTH HAVE INVADED THE
     MOON.": PRINT : PRINT
11020 PRINT "    YOUR MISSION IS T
     O DESTROY AS MANY ATTACKING W
     AVES OF THOSE ALIENS BEFORE"
11030 PRINT "THEY LAUNCH THEIR FLE
     ETS TO CONQUER THE EARTH.": PRINT
     : PRINT
11050 RETURN
```

Input subroutine.

```
11100 VTAB 21: PRINT TAB( 11);"JO
     YSTICK OR KEYBOARD?";: GET I$:
      PRINT : IF I$ < > "J" AND I$
     < > "K" THEN 11100
11110 KB = 0: IF I$ = "K" THEN KB =
     1
11115 PRINT
11120 VTAB 23: PRINT TAB( 11);"PU
     SH 1 OR 2 TO START";: GET I$: PRINT
     : IF I$ < > "1" AND I$ < > "
     2" THEN 11120
11130 PP = VAL (I$):PL = 1: GOSUB
     5000: RETURN
```

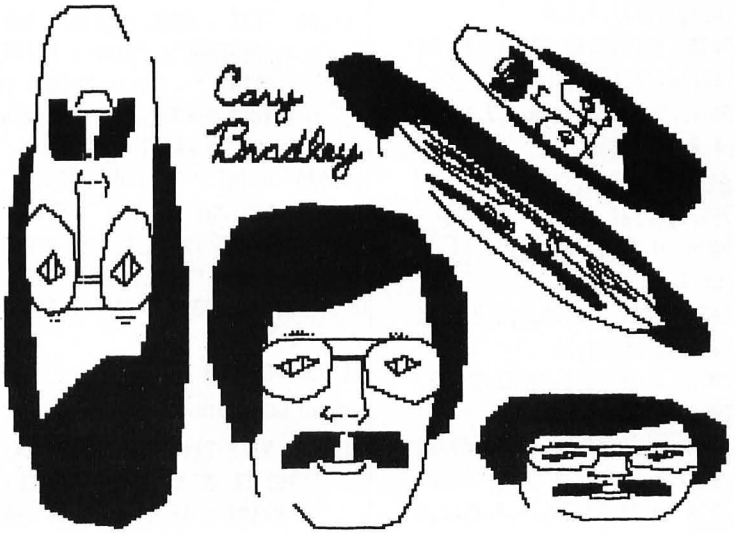
APPLE™ SWAT TABLE FOR: **DEFENSE**

LINES	SWAT CODE	LENGTH
5 - 100	RA	513
105 - 149	JM	355
150 - 210	LP	374
220 - 1020	XY	478
1025 - 3050	PF	373
3055 - 4010	CX	358
4030 - 4130	DH	506
4135 - 6047	XZ	514
6050 - 7020	CW	393
7050 - 10044	OW	452
10045 - 10054	RO	520
10055 - 10068	YQ	514
10069 - 11100	EW	502
11110 - 11130	BF	121

GPS — Graphics Processing System

Reviewed by Cary W. Bradley

By Richard Blum, Stoneware, Inc., 50 Belvedere Street, San Rafael, CA 94901. System requirements: 48K Apple II Plus™, one disk drive, monitor, game paddles or joystick. Optional hardware: Apple™ Graphics Tablet or Symtec Light Pen, second disk drive, 16K RAM card or Language Card, graphics printer or plotter (see text). Suggested retail price: Professional Version — $179.00, Standard Version — $69.00.



The number of ways you can make use of the Apple II's™ graphics features is virtually limitless. There is no better evidence of this than the profusion of software packages that enable you to approach Apple™ graphics from every imaginable standpoint. Although you may never need to create shape tables or animation routines, you will probably want to use the hi-res screen to display a picture. Even in this limited context, you'll have a wide choice of software packages to do the job in one way or another.

Stoneware's *GPS — Graphics Processing System* — is one of the newest entries into the market for Apple™ drawing software, and will be a formidable competitor for others in the field. *GPS* is based on the concept that the elements of a picture should be subject to manipulation much like letters, words, sentences and paragraphs in a word processor. *GPS* implements this idea with finesse; I used it to create the best pictures I have ever made with my Apple™.

GPS was originally intended for people using the Apple™ for professional design work; illustrators, designers, architects, and so on. It was programmed to support a variety of drawing tools: game paddles, joystick, Apple™ Graphics Tablet and Symtec Light Pen. For output, *GPS* supports the Apple Silentype™ printer, other (unnamed in the manual) graphics printers, and Houston Instruments HIPLOT DMP 3, 4, 6 and 7 graphics plotters. A modified version, to be released in September, will support the Hewlett-Packard 7470A and Strobe 100 graphics plotters, the IDS 460 and 560, and Epson MX80 printers.

For those who could use *GPS* but don't have all that specialized hardware, Stoneware has a Standard Version of *GPS*, which supports only game paddles or joystick, and the Silentype™. The cost difference from the Professional Version is substantial, and you still get more drawing capability than you ever bargained for. The self-portrait I drew to accompany this article was done with nothing more complicated than game paddles; I'll explain later how it was done.

Describing *GPS* is not easy. It's one of those cases where, in order to understand the parts, you have to understand the whole, and vice versa. This same problem is evident in the system documentation. In the "Forward" [sic] to the manual, it is suggested that you have the choice of reading the manual carefully or using the "Quick Reference Chart" to "experiment your way through *GPS*." Try one or the other, preferably a thorough reading of the manual, but don't try, as I did, to figure it out for yourself. There is no prompting, other than menus, and much of what goes on while you are using *GPS* is not evident from what happens on the screen. After a few minutes of frantic knob-turning and button-pushing, it's easy to get the erroneous impression that the program doesn't do anything. Nothing could be further from the truth. *GPS* is extremely powerful, but it will take you some time and effort to learn to use it.

A picture created with *GPS* is made up of certain elements, organized according to a hierarchy. You can manipulate and modify the elements singly or in chunks, depending on how you have defined the hierarchy. When you draw something, it is implicitly defined as an "object." That makes sense. Then, you can tell *GPS* that you wish to associate several objects, done by creating a "group." The collection of all objects and groups you have created is called the "picture." This system of image organization is what gives *GPS* its power. It is a shame that this simple explanation is not the first part of the tutorial section of the manual.

All of the manipulations and modifications you can do with *GPS* are performed on either an object, a group, or a picture, at your option. Having created objects and groups (and, hence, a picture) you can

move, enlarge, shrink, rotate, erase, duplicate or color any part of the picture you desire. The analogy to word processing is accurate; while the whole picture is in the Apple's™ memory, you can change, relocate or eliminate any object or group of objects without affecting anything else in the picture.

GPS is driven by menus which appear at the bottom of the video screen. A cursor is used to indicate the action you want *GPS* to take. The exact form of the cursor depends on the hardware you are using. For this discussion, I will assume that you will be using game paddles, in which case the cursor is a single, hi-res dot. The only times you use the keyboard are when you specify hardware immediately after booting, and when you supply file names for disk operations. Menu options are selected by placing the cursor over the appropriate word at the bottom of the screen and pressing the button (for graphics tablet users, this means pressing down on the pen). When a menu is visible, the cursor disappears when it is in the menu area. If it sounds confusing to be moving a cursor you can't see, it is, at first, but you'll find that positioning the cursor in the menu area becomes simple with very little practice.

The *GPS* main menu gives you eight options; Draw, Erase, Modify, Duplicate, Display, Group, Information and Cancel. Various submenus appear after you select one of the main menu options. The combination gives you a total of over 50 options from which to select.

The Draw option gives you two methods for creating an image: freehand and line drawing. The freehand option is best with a graphics tablet or light pen. With paddles, it works like an Etch-a-Sketch, with one paddle controlling vertical motion, and the other, horizontal. But, drawing only happens when one of the paddle buttons is held down, which makes it a real test of manual dexterity. If you insist, you can probably rig up a rubber band or clamp of some kind to hold down the button, but line drawing is more rewarding. You simply position the cursor where you want the line to begin, push the button, move the cursor to where you want the line to end and push again. Instant line. You can continue connecting lines from successive endpoints, or begin new lines, as you wish.

While in the Draw mode, you can erase a portion of the image you're working on, or change its color. You can also choose Cancel, which erases what you've just drawn and lets you start over. When you finish the image, select "End" and you've established, in *GPS* terminology, an object. That object then stays as it is when you reenter Draw mode to create other objects. "Cancel" only erases the object you're currently working on.

From the main menu, select "Group," and you can tell *GPS* to associate any combination of objects for manipulation as a single unit. Even when an object is a member of a group, you still have the ability to work with it independently of the other objects in the group.

Back at the main menu again, choose "Modify," and you'll see what is so exciting about *GPS*. The Modify sub-menu is loaded with methods for changing anything you've drawn to suit your fancy. Modifications can be made, at your option, to any object or group, or the entire picture at once. You can change the position, size or color of anything. You can rotate anything, to any degree you wish, about any point you choose. You can also change the proportions of any image, increasing or decreasing height, width or both. The infinite variety afforded by the Modify menu will let you make your picture look exactly as you want it.

The Display feature allows you to zoom in on any part of your picture, at either 4 or 16 times its original size. All other *GPS* options can be used while the zoom is in effect. With Display, you can also look at any object or group by itself, or clear the screen for undistracted drawing, while your picture is retained in memory.

Erasing of any object, group or picture can be accomplished at just about any point in the program. Color can also be added or changed at several points along the way. You can choose a color for any outline, or to fill an object. The background of your hi-res screen can also be filled with any color. Unlike some "coloring book" type programs, *GPS* doesn't give you 947,000 weird colors to choose from; you have the basic eight hi-res colors, and combinations of any two of them. That's plenty of variety for any practical application.

Objects, groups or pictures can be saved on a normally formatted disk for later use. Two formats are offered; *GPS* format, which permits further manipulation of objects and groups after reloading from disk, and Apple™ format, which is the standard 34-sector image of the hi-res area of RAM. This is sufficient for interfacing with any software which can handle this common type of file.

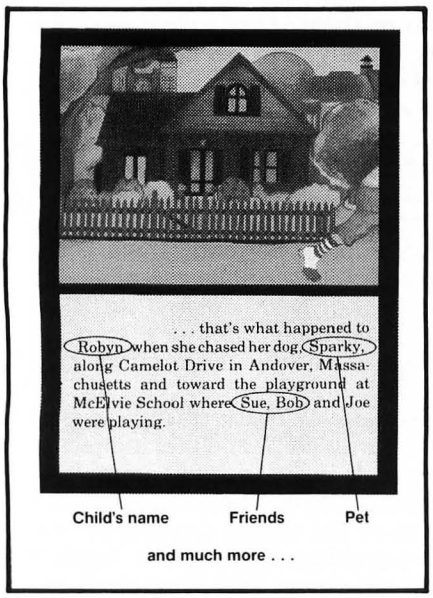
A Special Features disk is provided with *GPS*, and it is completely copyable using normal copy procedures, unlike the program disk. Two of the special features are not very exciting; they provide Apple™ text characters and large block letters which can be used in a *GPS* picture. The text characters are difficult to work with, and the block letters are crude, but don't let that turn you off to this otherwise marvelous package.

The special feature that is most useful is an option to create grids, handled as "objects" by the *GPS* program. The grid was the key to my self-portrait. Using a 20X20 grid, I was able to transfer a line drawing traced from a photograph to *GPS*. I used the 4X zoom feature to enlarge a portion of the screen, and the line-drawing option to draw angular approximations of various parts of myself. I put them all together as a group, and when I went back to the full hi-res screen, the smaller image was very good. I then duplicated the group several times and put it through a variety of size, rotation and proportion modifications to produce the final image. The signature was made similarly. Not bad for me and my little old game paddles.

For applications where hi-res pictures are needed, *GPS* is an excellent choice for both the professional and the home user. Where dedicated graphics hardware is used, *GPS* can help get the most out of it. Even with a minimum of equipment, you can still produce first-rate pictures. Now, aren't you ATARI® people jealous? No need to be — a version of *GPS* is in the works for you.

Personalized Children's Book System

...AN OPPORTUNITY TO ESTABLISH YOUR OWN PERSONALIZED BOOK CENTER THAT PRINTS & BINDS A HARDBOUND CHILDREN'S BOOK IN JUST MINUTES.



PROFIT WITH YOUR APPLE
There are many software packages for your Apple computer. But only I Discover Books offers a unique software package that enables you to earn money with your Apple by a combination of a business and an educational apple-cation.

PERSONALIZED BOOKS CREATED WITH YOUR APPLE
Each of the three I Discover Books are exciting adventure tales created just for that special child and is personalized with the child's name, address, birthdate, friends and pet. Each book is printed on beautifully illustrated, full color pages and is bound in a durable hard cover. Best of all, using the I Discover software and binder and your Apple II, you can create a book on site in just 5 *minutes*!

MULTIPLE MARKETS
Because the I Discover system requires an area of only approximately 5 feet by 5 feet, and is as fully portable as your Apple itself, the system can be moved to take advantage of seasonal fairs, exhibitions, special sales, events or holidays such as Christmas. Every loving parent, grandparent, aunt and uncle — in fact, everyone who has a special friend aged 1 to 12 is a potential purchaser of an I Discover Book.

ORDER YOUR I DISCOVER SYSTEM NOW TO TAKE ADVANTAGE OF THE CHRISTMAS SEASON.

For Apple II owners with one disk drive in 48K

It's New — It's Unique

Join the growing network of personalized children's book centers. Our computer system enables you to produce a personalized book in just minutes. Software & Marketing manual only $395.00.

Solid Marketing Plan

We will help you develop a full marketing plan, including:

- **Direct Mail**
- **Point of Purchase Displays**
- **On Site Locations**
- **Premium Markets**
- **Space and TV Ads**
- **Educational Markets**

Profit Potential

The suggested retail price for I Discover Books is $9.95. We supply you with all the necessary covers and blank pages... your cost is approximately $2.50 per book... a profit of nearly $7.50 per book. Currently, we have three exciting adventure stories available, including; *Adventures on the Riddle Planet, The Mystery of Scented Mountain* and *The Holiday Dragon.*

I'M SOLD — SEND THE SOFTWARE

☐ Please send the complete I Discover Personalized Book System and Manual along with 15 sample books. I understand if I am not completely satisfied, I may return the software for a full refund. I am enclosing $395.00.

Name _____ Printer Model _____

Street _____

City _____ State _____ Zip _____

I'M NOT SOLD YET

☐ Please send me an information packet on the I DISCOVER SYSTEM (No charge)

☐ Please send me a sample book for the child named below in addition to the information packet. (Please enclose $5.00 for processing. Reg. Retail $9.95)

Child's first name _____ Last _____ ☐ Boy ☐ Girl

Street _____ City _____ State _____ Zip _____

School or park _____ Age today _____ Birthday __/__/__ Pet's name _____

Kind of pet _____ Three friends 1. _____ 2. _____ 3. _____

How do want your name to appear in the book (ex.: Mom, Dad, Aunt Jane, etc.) _____

Your name _____ St. _____ City _____ State _____ Zip _____

Send to: I DISCOVER BOOKS • P.O. Box 170, Andover, MA 01810 OR Call 617-689-2496

The Complete Graphics System & Special Effects

Reviewed by Ame Choate Flynn

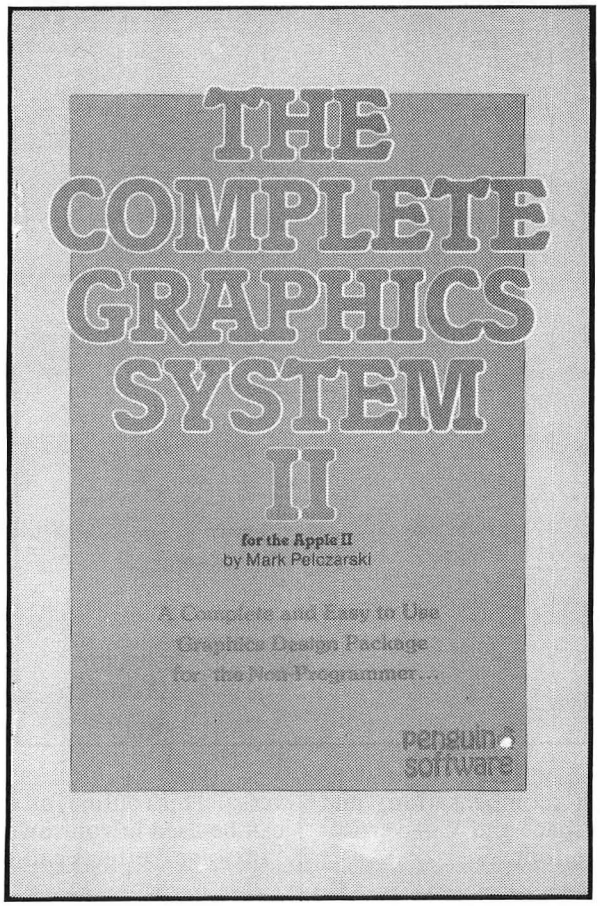
By Mark Pelczarski from Penguin Software, 830 4th Avenue, Geneva, IL 60134. System requirements: 48K Apple II™ with paddles and disk drive, Applesoft firmware or the language system. A version including an Apple™ graphics tablet is also available. Retail price is $109.90, or $189.90 for the graphics tablet version.

The Complete Graphics System/Special Effects. It's a mouthful. It's essential. I'm hooked on it. I would guess that 90 percent of the work I create with my Apple™ computer has at least one element from this software package. It is such an all-purpose tool that I still do not feel fully conversant with each and every module it contains. I have watched it grow over the past year, become more sophisticated, more friendly to the artist user. On more than one occasion, I'm sure I've driven its author crazy!

The advertisement said 108 colors! I was *so* sick of the Apple™ basic eight that I called, spent a lot of money, and had it sent UPS Blue Label. I then entered into a very close relationship with a piece of software and a Penguin.

I knew what I wanted to do with my Apple™. Create, draw, design and develop entertainment or educational packages. Do illustrations for children of all ages. I wanted them to have the same integrity as those paintings and drawings I made with more traditional media and methods. I wanted to start a micro-based computer graphics company and keep it going — make it respectable at SIGGRAPH — and eventually, make it easier for other artists to use a computer.

After more than a year, and many calls to Illinois, I think we've got it. If you can afford only one Computer Graphics package, this is the one to buy. It is the mainstay of my MicroGraphics courses at the New School in New York. The documentation is not written only for assembly-level programmers, but is easy for an artist to muddle through and learn by doing. It covers all the aspects of Apple™ graphics that come to mind, and throws in a few twists, too. I'll step through the various modules and modifications and try to explain how they can be used with each other.



Drawing And Painting

Parts of *The Complete Graphics System/Special Effects, (CGS/SE),* can be considered as creative modules. In the Complete Graphics System, the Drawing Module would be your starting point. With it, you can draw with lines, as in "connect the dots," or choose from eight brushes to paint with various shapes. Throw in the Brush Module from Special Effects and, voila, 96 more brush shapes!

When someone exclaims "You're an artist! I can't even draw a straight line!", I'm tempted to tell the truth. I can't draw a straight line except with a T-square and triangle. The text feedback at the bottom of the screen in this program lets me know if the line I want to make will be straight. Just match up the horizontal or vertical values, press the button or push on the stylus and it's straight. It works as well for circles and ellipses, which I managed to forget after I passed geometry in junior high school. There are plenty of times when I have to duplicate a well-known logo and thank Penguin Software that these options are included.

With the Brush Module, you can put color next to color; you can go "out of bounds" as far as lines are concerned. With Autofill, you can fill any enclosed, bounded area with any of the 108 colors.

The Fill (Manual) option has a confusing title. With this option, you set a point and your "fill" will radiate around it. It gives great starburst and "emanating ray" effects, but the name tends to confuse my students. The Manual Fill can only be used with the Apple™ colors.

Which brings us to 108 colors! Realize that 0 is a color, and the palette only goes up to 107. This is a common, non-computerist's pitfall — realizing that you have to start from zero. The Palette (which is displayed by pressing "P") is shown on Page Two. These are not true colors, but rather mixtures or patterns of the standard eight. However, most come close enough to what you need that it doesn't matter. Art, after all, is an illusion.

When you reach a point in your graphic masterpiece that you want to get picky about individual areas or pixels, the

Magnificent Mode in Special Effects lets you get "up close and personal." Using this option will put the cap on the Apple™ color incompatibility mystery. It will teach you what happens when the high bit is turned on or off and which colors come in what columns. Read the documentation and use the module if you are even thinking about teaching Apple™ Graphics. It's "theory in action." Then try to explain it to a non-computerized friend.

Text

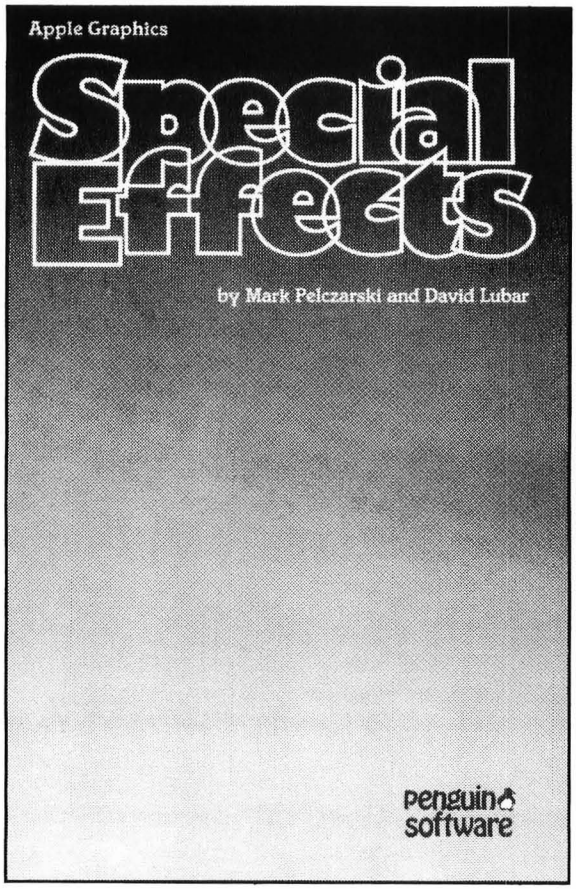
You now have a finished graphic. How do you sign or title it? There is a Text Module in *CGS*. With it, you can generate text or type on the screen with large or small characters. If you don't like the available fonts, create one of your own with the Font Editor. If you adore a character set from DOS TOOLKIT or HIGHER TEXT, the Conversion Utility in Special Effects will let you use them.

You can type in three modes — Destructive, Non-Destructive or Reverse. With the use of large fonts you can have colored text, again in 108 colors.

Be forewarned that the Palm Trees and the Penguin included in the font can only be reproduced under program control. My students used the Editor to duplicate them in their own fonts.

Mr. Pelczarski has been introduced to some of the sophistications of traditional typography. An "i" should not float in space, an "A" and a "V" should be tucked together like this: AV. He has included this "kerning" function in his latest versions and it does make a difference. Hitting the Escape key and the ← or → key will let you move a letter forward or backward one line.

One of my clients complained that the small text was unreadable and that use of the large font gave a page the contents of a telegram. I designed a mid-sized font with the editor. Instead of the usual 14 by 16 dot matrix, it was 13 by 14. The character set was legible and functional. The character-tightening (compensation) and line space functions (H — horizontal space and V — vertical space) enabled us to use this new size font.



Converting Pictures and Fonts

The Text or Font Converter of *Special Effects* has been mentioned. It is part of a larger utility titled The Conversion Module. It not only converts fonts, but allows the artist to take other software-created pictures and use them with the Penguin programs. This is a matter of proper page placement and a .PIC on the end of the title. It saves much hair-pulling and yelling on the part of the artist who wants to use all those colors on a differently formatted graphic.

Shrink

Now we have the graphic, but have become a tad bored with it. How can we liven it up?

We can shrink it. Reduce it in size. There is a Shrink Utility in the *CGS* which gives you the picture or image in one-quarter size. The colors may not reproduce, but if you are doing a logo, this is an invaluable utility. It lets you place the image in any or all of the four quadrants and the intensity (black and white) can be controlled. While you're at it, you can reduce a reduction. Find out how small your picture can get before it loses all comprehensibility.

Tricky Graphics

The *Special Effects* Graphics Tricks module lets you play with your graphics. You can get mirror images and reverses using Picture Flips. I changed the Cumberland Gap into the Donner Pass by changing its direction and adding snow. (It was actually a bit more complex, but I needed somewhere to start.) Once left is mirrored on right, and top on bottom, the starting image is abstracted and duplicated in all four directions.

More tricks can be carried out by going to the Color Tricks mode. Here, complex color changes are made. Swap green for orange, or blue for violet. Using the tricks in combination will give you a total of 32 possible changes. You can go back and forth between Picture Flips and Color Tricks for more variations.

A Move routine is included to move rectangular parts of your picture — either from the same page or from the other page. This so intrigued some of my students that they created an ice cream cone with scoops of ice cream slowly descending from the top of the graphic and landing on the cone.

Spiral Transfers move an entire picture from page 1 to 2 or vice versa. This routine, as well as most others in *Special Effects,* can be used in your own programs. By increasing the speed, "dissolve" effects can be achieved.

Packing and Stringing

Alright — I now have 52 wonderful pictures that I want to have on one disk. I know that about 13 is the limit. How do I do it?

The Packing Module packs standard pictures (34 sectors) and stores them in less space. The more complex the picture,

the more space it will take. But it will always use less than 34 sectors.

Once packed, a string of packed pictures can be created. This can contain from three to ten pictures. If you "preplan" or storyboard your string, adding or deleting discrete elements, this can be played back for animation. You can look at your string with the Viewstring Module, or again, use the Programmers Notes to put it in your own programs.

A Third Dimension

Three-dimensional graphics can be designed and manipulated using the 3-D Graphics Module and the Panel Utility.

These modules enable you to use the Apple's™ 3-D capability. Two-dimensional surfaces or panels are drawn with the Panel Utility and stored with 3-dimensional coordinates - x, y and z. When a panel is finished, it can be taken to the 3-D Module and joined with other panels to create a complex 3-D figure.

The 3-D Graphics Module takes a figure created with the Panel Utility and lets you view and manipulate it. Rotations, moves, scaling and distortions can be performed. You can also edit individual points of a figure while you are viewing it. Three-dimensional images are impressive and complex, but by working your way patiently through the utility with the documentation, you should be able to understand and create them.

Shapes

The Apple™ computer can store graphic objects called "Shape Tables." In *CGS/SE* these shapes are used as brushes. The Shape Module helps you create your own custom-made brushes for use in the Drawing Module. With the "V" option, you will be able to scale and rotate the shape table using paddles.

It is impossible to go into depth on this software package. It would take another article. I've hit the high points, and hopefully, I've intrigued budding artists, or artists who want to use the computer. Mark Pelczarski encourages feedback from users and is constantly updating the software to make it more comprehensive. Manufacturer support such as this is a primary factor when buying any software package.

Artists (non-programmers) like me either have to have a full-time programmer on staff, or Mark Pelczarski's software and helpful hints. Use your imagination! Just because a programmer wrote the software doesn't mean he's tried it out. A character-generator program can become a method of animation. A way of reading out pictures becomes animation in its own right. *How* you create the picture is the animation.

Versions of *The Complete Graphics System/Special Effects* are written for both paddles and joystick, or the Graphics Tablet. Both programs are available on one disk, which saves wear and tear on wrists. These are also unlocked, unprotected disks. Buyers are on the "honor code" not to spread them around, but they can also make their own back-up copies. My copies get a lot of use and it is nice to know that if one wears out, there's another waiting in the wings.

ILIST by Joe Iwanski

***ILIST* is a Machine Language utility which provides listing and cross reference enhancements for TRS-80® Model I/III BASIC programs. It requires 48K and one disk drive. A printer is assumed, but not required. It is included as the bonus program on this issue's TRS-80® DV.**

Initial Display — After *ILIST* loads, it will present an initial display screen. This screen will be used throughout *ILIST* execution.

In The Box — There is a rectangular "box" drawn at the right of the display. All the data you enter will be displayed within this box. All instructions and prompts will also be contained there.

At the bottom of the box, you will see two lines of information:

<<BREAK>> To Set New Options
<<ENTER>> Program File Name

The <<BREAK>> line describes what will happen if you press the BREAK key. Depending on where *ILIST* happens to be, this will be either 1) Set New Options 2) Cancel this Run 3) Return to DOS.

The <<ENTER>> line describes what *ILIST* expects you to enter from the keyboard. Whenever there is a line in this position, you must enter some information. During *ILIST* execution, the <<ENTER>> line is inactive and will appear as <<*****>>. The <<BREAK>> line is always active.

Processing a Program — All BASIC programs processed by *ILIST* are listed from standard BASIC "SAVE" files. If you try to list an ASCII-format file, *ILIST* will give you an error message. Specify the first program to be processed by entering its BASIC filespec.

Note On Keyboard Entry — A marker will always point to the area where your input characters are displayed. Use the left arrow key to backspace, and ENTER to terminate. If you hit BREAK, the <<BREAK>> action will be taken.

All input fields have a maximum length specified by a line of dashes following the pointing marker. If you enter the maximum number of characters, entry is terminated, and you don't have to hit ENTER.

If you enter a file name which doesn't exist, or with bad syntax, or if the file is not a BASIC program, you will get an error message above the <<BREAK>> line. The program will prompt you to re-enter correctly.

Titles and Remarks — The next prompt will ask for TITLE information. See the section, "Special REMs," below for a method to have this processed automatically. Enter the TITLE information you want printed at the top of each page.

The next prompt will be for a REMARK. This information will also be printed in the Page Heading, below the TITLE. It is designed to carry such information as: Version 1.2, TEST RUN 12, etc.. You will always get a chance to enter a REMARK.

You may hit ENTER in response to either the TITLE or REMARK prompt, and the Page Heading will be blank.

What Happens Next? — You will note that *ILIST* has constructed another display to the left of the "box" which looks like:

Program	Status
YOUR PROG	HOLD

ILIST will then ask you to enter another program file name. If you wish, you may enter up to eight programs to be listed, and *ILIST* will "batch" them for processing.

To erase the previous entry, enter a dollar sign when you are asked to enter the program file name. This will erase the last entry.

To terminate a "batch" list, press ENTER. This tells *ILIST* to process the batch and list the program(s).

Display Information — As you are processing a "batch," the screen will inform you of what's happening. An arrow will appear in the Status Area, pointing to the program now being processed. Its status will change from HOLD to MAIN, SORT, REF, and DONE as *ILIST* runs. You may not notice the SORT message, as it is quite fast.

Page and line numbers are displayed as they are processed, and a count of the line and variable references is displayed in "USED:". The maximum number of line and variable cross references is contained in "XREF Max". Even large programs do not often require over 1000, so 2981 should be more than enough.

Breaks And Errors — If you end a run by hitting BREAK while a program is being printed, the following will happen:

1) *ILIST* will print a special line on your listing: **** WARNING ! B-R-E-A-K Terminated Run. This warns you that this listing may be incomplete.

2) The Status Field for a program terminated this way will contain: < BK >

3) *ILIST* will generate a form feed, and process the next program in the batch. If there are no more programs, *ILIST* will restart and wait for new instructions.

If an error occurs during processing, *ILIST* will end the program, similar to the way BREAK does, with the following differences:

1) The message printed on the listing will be: ***WARNING! E-R-R-O-R Terminated this Run.

2) The Status Field will contain: E nn, where nn is the Error Code.

The Error Codes 1-50 are DOS error codes. Codes 90-92 are special *ILIST* error codes described at the end of these instructions.

Printout Explanation — In addition to the TITLE and REMARK, The Page Header printed on every page includes the following:

a) Page Number
b) Date — From DOS
c) Time — From DOS
d) Program Filespec

"Program Filespec" will be preceded by an asterisk if the Partial Lines Option was selected for that program. At the conclusion of the program listing, you will get a statistics trailer which contains:

a) Program Size, in bytes
b) Number of BASIC lines
c) Checksum (a single byte checksum of the entire program)

Following the Cross Reference Listing, you will get the following Statistic Printout:

a) Lines Referenced
b) Number of Variables in Program

Options — There are nine possible options which can be selected for each program in a batch:

1) MAIN LISTING (default = YES) — If this option is turned OFF, the program listing will not be output. All you will get is a page header and the statistics line at the end of the program.
2) CROSS REFERENCE (default = YES) — If this option is turned OFF, you will get neither the Cross Reference List nor the Cross Reference Statistics line.
3) DOUBLE SPACE (default = YES) — YES selects double spacing, NO selects Single Spacing. See REMs section below for other spacing options.
4) DECOMPRESS (default = YES) — If NO, the program will be listed exactly as it appears on the disk SAVE file. If this option is YES, *ILIST* will insert spaces to make the listing more readable.
5) DISK OUTPUT (default = NO) — If this option is YES, the program will be listed to a disk file. During the input phase, *ILIST* will ask you to enter the filespec for the disk output file. If you make an error in the name, *ILIST* will detect it and ask you to re-enter. If the name already exists, *ILIST* will *not* give you a warning — the existing file will be overwritten. If writing the output file generates a DISK FULL error, that run will be terminated with an error condition.
6) PRINT OUT (default = YES) — If YES, the printer will be an output device for all output specified. If NO, the printer will not be used.

If the YES Option has been taken, *ILIST* will warn you of printer unavailable conditions, with one exception. EPSON printers return no status

information to DOS when they are turned off. Consequently, *ILIST* will dump characters to the dead printer in ignorant bliss.

7) EPSON (default = YES) — If this option is set to NO, *ILIST* will print no special characters. All pagination is done by explicit line counting. If this option is set YES, *ILIST* will take advantage of some Epson printer features:

a) The TITLE will be printed in compressed expanded mode, giving a highlight to the page header.

b) *ILIST* will use Epson vertical tabbing to eject pages, giving a faster, smoother operation, and helping to insure that all new pages after the first are correctly aligned.

If you are printing a program with string packing, any characters in strings between 128 and 191 will be printed as graphic characters.

8) NORMAL MODE (default = YES) — If this option is YES, normal Epson printing mode is used. If NO, double-strike mode is used.

9) PARTIAL LINE NUMBERS — Enabled by entering an asterisk immediately before the input filespec name. If partial listing is selected, *ILIST* will, during the input phase, prompt for a starting and an ending line number.

If you enter no line numbers, *ILIST* will use 0 and 65529 as the starting and ending line number defaults.

Important Note: Even if you select a partial listing, *ILIST* will still cross reference the *entire* program, if that option was active.

All of the above options are reset to their defaults at the beginning of *ILIST*'s execution, and at the end of each batch. Of course, you can PATCH your copy of *ILIST* to have any standard option set you want.

All of these options, with the *exception* of Partial Line Listing, may be set during the batch input phase by pressing the BREAK key, i.e., when " <<BREAK>> To Set Options" is present on the display. Each setting of the option list is effective for all subsequent programs entered in the batch, but you may change the option set for each individual program in the batch, if you wish.

If you press BREAK, part way through a program specification, that particular program will be erased from the batch list. You could, therefore, use BREAK to cancel the particular program you have been entering. Setting options, however, will *not* erase or affect the programs that have already been entered in the batch. If the program has made it to the Status Display to the left of the box, it is *safe*.

When you press BREAK, the Option Screen will be presented. In the Option Screen, there are 6 keys that will "take." All other keys will do nothing.

ENTER will take you back to the input phase, so you can continue building your batch.

BREAK will take you to DOS.

Y or N will cause a Y(es) or N(o) to be set for the option marked by the position of the diamond shaped cursor.

Up or Down Arrows will move the diamond shaped cursor.

The Option Display contains two columns for each option. The left column (Std) is the default option from *ILIST*. The right column (Cur) is the current setting of the option. Y or N entries will affect only the Cur column. Std is here only for your reference.

Remember, all options are independent of each other, and can be entered in any combination, even if this makes the run nonsensical. For example, you can specify no printout and no disk output. *ILIST* will dutifully process, but you will get no output except the screen displays.

Special REMs — *ILIST* recognizes three kinds of special REM's which can be used in your programs.

1) TITLE REM — If the first statement in your program is a REM immediately followed by a plus sign, the first fifty bytes of this REM will be used as the program TITLE, and you will not be prompted for this information. You will still be prompted for the REMARK.

On all subsequent occurrences of a REM+, the first fifty bytes of the REM will be used as a SUBTITLE header, replacing the comment in REMARK, and cause a page eject. These can be used to divide your program up into sections.

2) SINGLE SPACE REM — A REM immediately followed by a "1" will force single spacing for all following lines.

3) DOUBLE SPACE REM — A REM immediately followed by a "2" will force double spacing for all following lines.

The Cross Reference List is not affected by these space REMs. It will be printed with the spacing selected in the option set.

The +,1, or 2 must *immediately* follow the REM to be effective. No spaces, please.

NOTES ON FORMAT

Multiple Statement Lines — Each statement is printed on a separate line, and the separating colons are not printed. You should also note that if spaces occur immediately after a colon, they will not be printed, in order to maintain the left margin.

FOR-NEXT Indentation — FOR-NEXT loops are printed with indentations, to help you read nested loops. This will occur up to a maximum indentation level of seven deep. Each NEXT causes the indentation to move back towards the left margin one space. Statements like NEXT A,B,Y are handled properly.

IF...ELSE Line Splitting — All IF...ELSE statements are split, with each ELSE starting a new line.

Apostrophe REMs — An Apostrophe-type REM ('), will cause an automatic extra line feed.

Note: Apostrophe REMs *cannot* be used for the special formatting REMs described above.

Left Margin — The left margin of the listing is indented somewhat to allow for three-hole punching.

Complete Cross Reference — All valid line references and variables are correctly tracked. To *ILIST*, GOTO 120, GOSUB 12, and RUN 100 are valid line references, but A = 120 is not. Also, ALPHA and AL are both accounted for as < AL >, but ALPHA$, AL%, and ALPHA# are tracked as separate variables, i.e., < AL$ >, < AL% >, < AL# >.

Wild Characters In Program — If you happen to use linefeeds (Down Arrows) to give your listings some character, *ILIST* will interpret these correctly, and give you a new line at the proper margin. Any other codes less than 20H will be treated and printed as spaces.

All codes higher than 7FH will be treated as BASIC tokens, unless those codes happen to be in a string literal. If they occur in a string literal (probably because of string packing), they will always be treated as spaces if the Epson option is NO. If the Epson option is YES, then the codes between 80H and BFH will be printed as Epson graphic characters, which happen to be the same as the TRS-80® graphics characters for those codes. If "tokens" should appear which have a value of over 250, these will print as: $UT251, $UT252, $UT253, $UT254, $UT255, that is, undefined tokens.

Exceeding Memory — *ILIST* uses all available free memory to store cross references. The number of cross reference slots will be displayed on the screen in "XREF Max". If you should happen to exceed this number while processing, *ILIST* will insert "#####" in the "Used:" slot on the display, and will print a warning message in the Cross Reference listing to let you know this happened.

With 48K, you should have over 3500 slots. With 32K, you should have more than 1000. It will be extremely difficult to exceed 3500, and 1000 will be sufficient for all but the largest programs.

***ILIST* Error Codes** — If a printout ends in an error, the DOS error message will be displayed. In addition, *ILIST* displays three special error codes as follows:

E 90 - Input file had bad filespec

E 91 - Output disk file had bad filespec

E 92 - Input File was not a BASIC file

SHOWS FOR YOU IN '82

THE NATIONAL COMPUTER SHOWS are the largest public computer expositions in the country. They feature hardware and software for business, industry, government, education, home, and personal use.

Under one roof you'll be able to test the new computers, desk top computers, data and word processing equipment, and a huge array of computer peripherals, computer services and computer supplies.

At each show you'll see all the major brands, all the major manufacturers, the big distributors and the local dealers and retailers. Everything together and for sale at super show prices.

Don't miss the National Computer Show coming to a city near you. Admission is $5 per person per day.

TICKET INFORMATION:
Send $5 (payable to National Computer Shows) for each day-long ticket along with the name of the Show you plan to attend to the address below. Tickets can also be purchased at the show.

The National Computer Shows,
822 Boylston St., Chestnut Hill, MA 02167
Telephone: 617-739-2000

BOSTON
THE NORTHEAST PERSONAL COMPUTER SHOW
Friday-Monday October 8-11 (Columbus Day Weekend)
Hynes Auditorium/Prudential Center
Show Hours: Friday 11 am to 9 pm,
Saturday & Sunday 12 noon to 9 pm,
Monday 11 am to 6 pm

WASHINGTON, D.C.
THE MID-ATLANTIC COMPUTER SHOW
Thursday-Sunday October 28-31
D.C. Armory/Starplex
Show Hours: 11 am to 6 pm daily

BOSTON
THE NORTHEAST BUSINESS COMPUTER SHOW
Thursday-Sunday November 11-14
Hynes Auditorium/Prudential Center
Show Hours: 11 am to 6 pm daily

ATLANTA
THE SOUTHEAST COMPUTER SHOW
Thursday-Sunday December 9-12
Atlanta Civic Center
Show Hours: 11 am to 6 pm daily

THE NATIONAL COMPUTER SHOWS

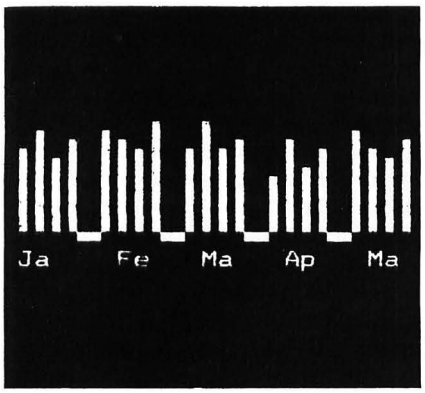
822 Boylston St., Chestnut Hill, MA 02167

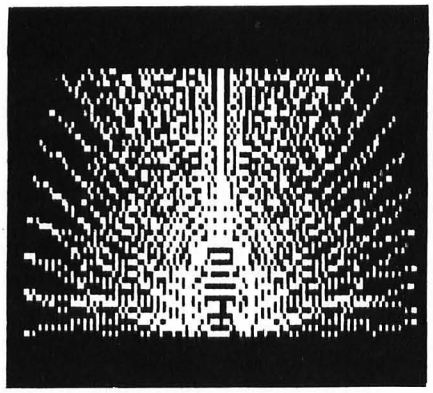
by Darwin Collins

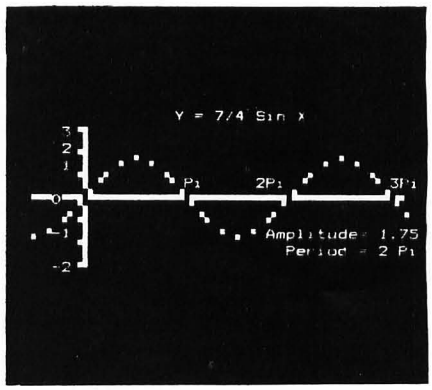
***Graphic Writer* is a graphics utility for a TRS-80® Model I or III with 16K (tape) or 32K (disk) RAM.**

Creating graphic displays on the TRS-80® can be a hair-pulling chore. *Graphic Writer* will allow you to design and edit complicated patterns with an ease greater than pencil and paper. After you have completed your creations, you'll be able to print them out on your line printer. Take some time to familiarize yourself with the various features of the program. In no time, you'll be creating graphic patterns for multitudes of uses.

When you run *Graphic Writer*, the title will appear, the screen will







GRAPHIC WRITER

clear, and questions will be asked about the vertical (height) and horizontal (width) parameters of the sketch (text) you want to type. The first question will ask the height of the material. Enter the number of lines you want to type, and press ENTER.

The second question will ask the width of the material. Follow the same procedure as with the height. The width of the material is the number of characters from the left to the right edge.

The directory will then appear. This directory will be the junction to reach all 14 sections of the program.

<C>haracter Writer

The arrows on the keyboard are directional controllers. When any of these keys are pressed, the cursor on the screen will move in that direction, if possible.

If the cursor is at the bottom of the screen and the height of the material is larger than the line number (The line number is after "Y>" on the bottom of the screen.), the down-arrow will move all screen contents one line upward and the next line will be displayed.

The same line movement occurs when the up-arrow is pressed and the cursor is at the top of the screen. If the line number is not 1, all lines will move downward, exposing the preceding line. When the screen width is larger than 64, a similar operation can be performed at the left and right edges. The character number is printed after "X>". This is the number of characters from the left-edge of the material.

Everything that is typed is recorded in memory so that when a line moves off the screen, it is not erased. The ENTER key will move the cursor to the left-edge of the screen and advance one line downward. If the cursor is at the bottom of the screen, it will only move to the left edge.

The @ key will transfer the special controllers between line and character modes. The current mode is shown on the bottom of the screen. The special controllers are used by pressing the SHIFT and the left or right arrow keys together.

If the mode on the bottom of the screen displays "(Shift < -)DEL (Shift ->)INSERT..." then whenever the SHIFT and left-arrow keys are pressed at the same time, the characters to the right of the cursor are moved one space to the left. The opposite happens whenever the SHIFT and right-arrow keys are pressed. There is a space inserted at the cursor and everything is moved to the right, including the character that was under the cursor.

The other mode is "(Shift <-)DEL Line (Shift ->)ADD Line...". When the SHIFT and left-arrow keys are pressed at the same time, the entire line is erased and everything below it moves up one line. When the SHIFT and right-arrow keys are pressed together, there is a line inserted at the cursor and all lines are moved downward one line.

Note: Anything moved outside the limits (height or width) is erased. For example, if there is a sentence on the last line of the material and there is a line inserted in the middle, the bottom line will move past the lower limit of the sketch and be erased. With this feature, a user can erase characters at the end of the sketch without having to move the cursor to it.

<D>irectory of Sketches — Disk Only

With this command, the filenames of all sketches on the disk

Bugs, Worms, and other Undesirables

In the ATARI® version of *Escape* from the Dungeon of the Gods, the printed listing omitted the underlining that indicates inverse text. The corrected lines are given here.

```
1 N0=0:N1=1:N2=2:N3=3:N4=4:N5=5:L1=970
:L2=1830:L3=960:L4=5030:DIM OK$(2):OK$
="OK"
610 GRAPHICS N0:? "You are in ";:P$=A$
:GOSUB N4:P$=B$:GOSUB N4:PRINT :PRINT
"Items you can see: "
618 PRINT :PRINT "Exits: ";:P$=D$:GOSU
B N4
630 PRINT :PRINT "Command ";:INPUT V0$
:GOSUB 6
1050 GRAPHICS N0:PRINT "Player's Inven
tory"
1800 GRAPHICS N0:PRINT "Player's Statu
s:"
1820 GRAPHICS N0:PRINT "Game over":P2=
P2-30:GOTO 2010
2130 PRINT "You are dead!!!"
2260 GRAPHICS N0:PRINT :PRINT " Escape
 from the Dungeon of the Gods":POSITIO
N 15,N2:PRINT "by Ray Sato"
2330 ? :? "      Press any key to cont
inue";
2390 ? :? "      Press any key to begi
n";
```

An error occurred in line 1152 of the ATARI® version of *Operation: Sabotage*. The second word in P$ should be XILDYZI.

The ATARI® version of *Gambler* (January, 1982) has a minor omission in line 1244. The correct line should read as follows.

```
1244 ? N$(IP*10-9,IP*10):? "How much w
ill you bet ? ";:J=IP:IF IP=N THEN B=5
4+INT(RND(0)*4):GOTO 1260
```


are listed to the screen. The sketches' filenames are kept in a sequential file called DIR-WRTR/DIR. To erase the entire contents of the directory, delete this data file.

<E>rase the Screen

If you press the E while in the directory, the program will erase all that has been typed in, including any graphics. Before it will do this, the program will ask if you are sure. If your answer is "Yes", it will erase the current file and ask you for new line widths and lengths. If "No", it will return to the directory without erasing anything.

<G>raphic Writer

This screen mode is similar to the Character Writer section, with the following differences:

1) There are no keys for writing letters or symbols.

2) The ENTER key will not move the cursor to the left-edge of the screen.

3) The BREAK key is not used.

4) The cursor is smaller.

5) The special controllers are not used.

The directional arrows are used to move the cursor, though the movement will be smaller than in Character Writer, for the cursor controls a smaller area. If a number (0-9) is pressed, the bottom of the screen will ask for a direction or the spacebar to cancel. If one of the arrow keys is pressed, the cursor will move the desired number of character spaces. Next to the x,y position indicator on the bottom of the screen are two vertical white bars with another cursor between them. This shows the position of the cursor within the current character space (2 by 3 pixels) it occupies on the screen.

The current controller mode of the cursor is shown on the bottom of the screen to the right of the two vertical bars. Q and W change the controller mode. In "Position Only" mode, the cursor will move freely about the screen, not drawing or changing anything already drawn. In "Set" mode, the cursor will leave a white pixel behind, when moved. When the cursor is moved in "Reset" mode, it will leave a black pixel behind. If the mode is in "Position Only", Q changes it to "Reset", to "Set" when pressed again, then back to "Position Only". Pressing W reverses the order of the mode changes.

There are three special drawing functions: <C>ircle, <S>quare, and <A>Line. To execute any of these commands, press the appropriate key. There will be a message indicating the type of function you are using on the bottom of the screen and a blinking marker will occupy the cursor's location.

There are three command keys to be used with the special drawing functions: J, K, and L. The L key will draw the desired shape without terminating the function. The K key will draw the shape and terminate the function. The J key will terminate the function without drawing.

The Line function will draw a line from the marker to the cursor. The Square function uses the marker and the cursor as opposite corners to draw a square. The Circle function will use the marker as the center of the circle and the cursor as its edge. The distance between them will be the circle's radius. When the edge of the circle being drawn is off the edge of the screen, the edge will be folded back on itself. With this feature, it is possible to draw curved lines other than circles. As in the regular drawing modes, a white line will be drawn when the controller mode is either "Position Only" or "Set", and a black line if it is "Reset".

As an additional feature, when D is pressed, the entire character block currently occupied by the cursor will be filled, according to the controller mode.

<H>ardcopy

The program will ask you if the parameters you've already set are OK. If so, the printer will print the sketch using those parameters. If not, the questions will ask for the left margin, top margin, if there are any graphics, and the type of printer used. If the printer is an Epson, there will be further questions about the type of print style you prefer.

continued on page 54

FIRST ISSUE FREE!

Get your first issue of COMPUTER GAMING WORLD free when you pay for a year subscription! For only $12.50 (U.S.) you get over a year's worth of reviews, strategy and tactics, playing aids, scenarios, contests, interviews, industry news, and more. CGW keeps you on the cutting edge of the computer gaming hobby and industry. Whether you have an Apple II, Atari, TRS-80, or other popular microcomputer, you will find many hours of entertaining and informative reading in the pages of CGW. Clip out and mail the coupon today and get seven bi-monthly issues at 40% off the newsstand price.

COMPUTER GAMING WORLD (Dept. S)
1919 E. Sycamore #203
Anaheim, CA 92805

Yes! Enter my subscription for a year of CGW and send my first issue free (a total of seven issues). U.S. - $12.50 Canada - $17.50; Foreign Air $26.50.

Please check one:

☐ Check or money order enclosed

☐ Master Card

☐ Visa

Name _____

Address _____

City/State _____ Zip _____

Name of Cardholder _____

MC#/VISA# _____

Exp. Date _____

Signature _____

This offer expires November 15, 1982.

When all questions are answered, there will be a sentence on the bottom of the screen asking you to adjust the paper and press the spacebar to begin printing. If you discover any mistakes in the answers you've given, or if you wish to exit the printing process, press any key other than the spacebar.

Note: The printer defaults at the beginning of the program to a left margin of 12, a top margin of 2, and to look for graphics. Also, the line spacing is at 1. These are the only parameters set at this time. If the printer is an Epson, make sure you set the parameters for it if you're using graphics.

<I>mitate a Character — Model III Only

This section allows the user to redefine the output of the keys on his keyboard. At the beginning of its execution, three rows of characters will be shown at the top of the screen. The first two rows are the alphabet and punctuation. The last row is for special characters that only the Model III uses. There will be a cursor blinking in the middle of these rows. Position the cursor over the desired character with the arrow keys and press ENTER.

The cursor will jump to the double set of characters at the bottom of the screen. Each of the columns is divided into two parts, on the left is the character on your keyboard and on the right is the character it will print if the key is pressed. Position the cursor over the keyboard character you wish to redefine and press ENTER. The desired character is now represented by the keyboard character you specified.

The redefined character will show on the right hand side of the keyboard character. Press the spacebar to return to the Directory or Y to redefine another character.

<K>ill a Sketch from Directory

Input the filename of the sketch you wish to kill and it will be deleted from the directory of *Graphic Writer*. To literally kill a sketch on the disk, you must do it manually from DOS using the command "KILL filespec".

<L>oad a Sketch

Type in the filename of the sketch that is saved. It will then be loaded using the same length and width parameters with which it was saved.

<M>ove a Line

A Sub-Menu will appear with 3 choices:

1) <A>dd or Delete Lines
2) <S>witch two Lines
3) <O>verlay a Line

<A>dd or Delete Lines gives you the option of adding or deleting lines from the bottom of the material. The program will ask the number of lines to add or delete. If lines are to be deleted, place a negative sign "-" before the number typed. If lines are to be added, type only the number. After this command is finished the program will return to the sub-menu.

<S>witch two lines will ask for the number of one of the lines to be exchanged, then the second line number. The exchange will occur and the program will return to the sub-menu.

<O>verlay a line will place a copy of one line over another. The first question will ask the line number of the source and the second, the destination — the line to be overlayed.

Note: The line numbers for line movement can be found at the bottom of the screen after "Y " in the Character Writer or Graphic Writer Sections. If an incorrect number is typed in response to the questions for line numbers, or if it is outside the boundaries of the material, the program will return to the sub-menu.

<N>ame a Sketch to Directory — Disk Only

Type the name of the sketch to be added. Its name will then be added into the directory of *Graphic Writer*. This command is generally used only when the programmer transfers sketches from one disk to another, with a DOS command, as this process does not automatically add the filespec to *Graphic Writer's* directory.

<O>pen ASCII Programs — Disk Only

To use this section, make sure the program you wish to load has been saved with the ",A" option. Type the filename of the program. It will be loaded into a sketch width of 64, so any lines longer than 64 characters will wrap around to the next line. All line numbers at the beginning of the statement lines must have a space between the number and the statements.

<R>estore ASCII Programs — Disk Only

The program to be restored must be loaded with the "<O>pen ASCII Programs" option. In order to retain proper line numbering you must put your line numbers at the beginning of the sketch lines and make sure there is a space between them and their corresponding statements. These procedures were described in the previous command.

<S>ave Sketch

Select and type the name under which you wish the sketch to be saved. The name will be saved with the file (and the disk version will add the name to the directory of *Graphic Writer*). Not more than 60 names can be saved under the *Graphic Writer* directory.

<Q>uit

Type Q if you wish to end the program. If you type it by mistake, type "CONT" and press ENTER.

Variables

Special Functions:
I$(0)-I$(14): A string that has its memory position moved to match each line on the screen.

String variables defined with DEFSTR:
A, B: Used for the cursor in Character Write and Question Cursor in other Sections.
A:and A1: Input Data for sequential files.
B: Cursor character — Underline Mark in Graphic Write.
C: Utility input string.
S(1)-S(90): Line for Sketches: one subscript for each line.

Standard String Variables:
R$: Utility variable. Used a lot...

also in filenames.
P$: Beginning of the question for printer output.
P1$: Graphics/No Graphics flag.

Numerical Variables:
D: The ASC value for input string C in Graphic Writer. Also, number of lines to be added or deleted in Move a Line.
D(R): R's domain is 122, ASCII Code representing Image Character for a keyboard entry in Character Writer.
E: Flag for Special Controller mode in Character Writer.
G: On/Off flag for Draw functions in Graphic Writer.
I: Screen position to print the Cursor.
K: Direction to move Cursor in Graphic Writer.
L: Set/Reset flag in Graphic Writer.
P: Sketch Line Number of the screen for placing characters into their proper subscript string.
P1!: COS distance in circle draw mode.
P2!: SIN distance in circle draw mode.
PE: Number to be added in Printing Graphics if using an Epson printer in Hardcopy Section.
PL: Printer left margin for Sketch.
PS: Printer line spacing.
PT: Printer output top margin.
R-R4: Utility variables.
RA!: Radius of circle in Circle Draw mode.
T: X sub-position of cursor in Graphic Writer.
T1-T4: Cursor position values for the Draw functions in Graphic Writer.
T1: Horizontal cursor value.
T2: Vertical cursor value.
T3: Horizontal cursor value at beginning of Activation.
T4: Vertical cursor value at beginning of Activation.
U: Y sub-position of cursor in Graphic Writer.
W: ASCII Value for character PEEKed in Character Writer, Flag for Cursor Mode in Graphic Writer.
W1: PEEKed ASCII Value for cursor in Graphic Writer and Imitate a Character.
X: X position of cursor in Character Writer.
XL: Left margin on sketch in Character Writer.
XM: Maximum sketch width in characters.
Y: Y position of cursor in Character Writer.
YL: Top margin on sketch in Character Writer.
YM: Maximum number of line numbers for Sketch.

Special Pokes:
Poke 16444,1: Repeat on Arrow keys and Spacebar Only.
Poke 16409,1: Sets Model III keyboard characters to upper case.

```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS        TRS-80 BASIC         SS
SS     'GRAPHIC WRITER'        SS
SS   AUTHOR: DARWIN COLLINS    SS
SS    COPYRIGHT (C) 1982       SS
SS SOFTSIDE PUBLICATIONS, INC  SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```

Initialization

```
10 CLS:CLEAR50:PRINT@272,STRING$(32,131);:PRINT@720,STRING$(32,1
76);:FORR=4TO11:PRINT@R%64+15,CHR$(191);:PRINT@R%64+48,CHR$(191)
;:NEXT
20 PRINT@152,"Graphic  Writer";:PRINT@857,"Graphic Version":PRIN
T@1000,"By Dar Collins";
30 PRINT@475,"Graphic";:PRINT@539,"Writer";
40 CLEAR:IFMEM>32700THENCLEAR31000ELSECLEARMEM-1500
50 DEFINTD-Z:DEFSTRA-C,S:DIMS(90):DIMD(122):DIMI$(14):P$="Should
 the Print be ":ONERRORGOTO2070:PRINTCHR$(21)CHR$(22);
60 PL=12:PS=1:P1$="Y":PE=0:PT=2:FORR=32TO122:D(R)=R:NEXT
70 FORR=0TO14:I$(R)=" ":POKEVARPTR(I$(R)),64:POKEVARPTR(I$(R))+2
,INT((R%64)/256)+60:POKEVARPTR(I$(R))+1,R%64-INT((R%64)/256)%256
:NEXT
```

Sketch Size Init.

```
80 CLS:FORR=1TOYM:S(R)="":NEXT:XM=0:PRINT@200,"What is the Line
Width of the Screen
(130 max., 64 default)";:INPUTXM:IFXM>130ORXM<5THENXM=64:PRINTTA
B(20)"Width set to 64"
90 YM=0:PRINTTAB(8)"What is the Height of the Screen
(90 max., 15 Default)";:INPUTYM:IFYM>90ORYM<1THENYM=15:PRINTTAB(
20)"Height set to 15"
100 X=0:Y=0:YT=1:XL=1:T=0:U=0:FORR=1TOYM:S(R)=STRING$(XM,32):NEX
T:R$=""
```

Directory.

```
110 A=CHR$(158):B=CHR$(173):L=0:POKE16409,1:CLS:PRINT@256,"Width
  Max.:"XM:PRINT"Height Max.:"YM:PRINT@320,"What is the choice?";
:PRINT@384,"<C> haracter Write";:PRINT@448,"<D> irectory of Sketc
hes";:PRINT@512,"<E> rase the Screen";
120 PRINT@576,"<G> raphic Write";:PRINT@640,"<H> ardcopy
"TAB(20)"<I> mitate a Character
"TAB(20)"<K> ill Sketch on Disk
"TAB(20)"<L> oad Sketch from Disk"
130 PRINTTAB(20)"<M> ove a line of a Sketch
"TAB(20)"<N> ame a Sketch to Directory
"TAB(20)"<O> pen Ascii Programs
"TAB(20)"<R> eStore Ascii Programs
"TAB(20)"<S> ave Sketch on Disk
"TAB(20)"<Q> uit,exit the Program
"TAB(13)"** <Press the Appropiate Key>
**";
140 I=40:GOSUB160:R=INSTR(" CMHGSLQEIDNKOR",C):IFR=0THEN140
150 ONRGOTO290,290,480,610,790,1390,1440,1500,280,1505,1660,1710
,1800,1890,1970
```

Hall of Subroutines.

Letter Inkey$.

```
160 POKE16444,1:PRINT@I,B;:FORR=-5TO5:C=INKEY$:IFC<>""THENRETURN
ELSEIFR=0THENPRINT@I,A;:NEXTELSENEXT:GOTO160
```

Any key Inkey$.

```
170 C=INKEY$:IFC=""THEN170ELSERETURN
```

TRS-80®

Slow Inkey$ Response.

```
180 FORR=1TO1250:IFINKEY$<>""THENRETURNELSENEXT:RETURN
```

Y/N Inkey$.

```
190 GOSUB170:IFC="Y"THENPRINT" Yes":RETURNELSEPRINT" No":RETURN
```

Print Sketch to Screen.

```
200 FORR2=0TO14:LSETI$(R2)=MID$(S(R2+YT),XL):NEXT:RETURN
```

Move Cursor Right.

```
210 IFX+1<XMANDX<63THENX=X+1:GOSUB230:RETURNELSEGOSUB390:RETURN
```

Move Cursor Left.

```
220 IFX-1>=0THENX=X-1ELSEGOSUB380:RETURN
```

Print X position.

```
230 PRINT@962,X+XL;:RETURN
```

Print Y position.

```
240 PRINT@969,Y+YT;:RETURN
```

Set/Reset Inkey$.

```
250 POKE16444,1:SET(T1,T2):GOSUB1250:FORR=-5TO5:C=INKEY$:IFC<>""
THENGOSUB1240ELSEIFR=0THENRESET(T1,T2):GOSUB1240:NEXTELSENEXT:GO
TO250
```

Set/Reset Function for Cursor Movement.

```
260 IFW=0THENRETURNELSEIFW=1THENSET(T1,T2):RETURNELSERESET(T1,T2
):RETURN
```

Screen to Sketch Memory Routine.

```
270 IFL=0THENRETURNELSER2=-((YM-YT)=>14)*14-((YM-YT)<14)*(YM-YT)
:FORR3=0TOR2:MID$(S(R3+YT),XL,64)=I$(R3):NEXT:L=0:RETURN
```

Erase Sketch from Memory

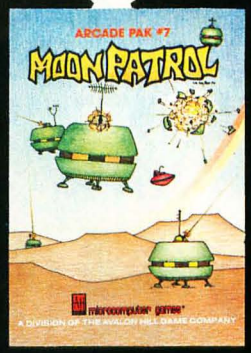
```
280 CLS:PRINT@13,"Section to Erase all Material on Screen":PRINT
@130,"Are You sure (Y/N) ?";:I=150:GOSUB160:IFC="Y"THEN80ELSE110
```

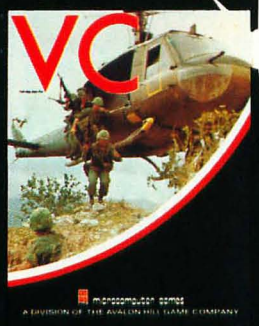
Character Write Section

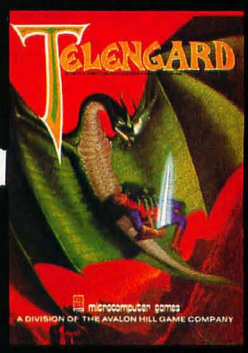
```
290 CLS:GOSUB200:L=0:E=0
300 PRINT@960,"X>     Y>    (SHIFT <-)DELETE (SHIFT ->)INSERT (C
LEAR)DIRECTORY";:GOSUB230:GOSUB240:B=CHR$(95)
310 P=Y+YT:I=X+(Y*64):W=PEEK(I+15360):A=CHR$(W):GOSUB160:D=ASC(C
):IFD<>91ANDD>31ANDD<>64THENPRINT@I,CHR$(D(D));:L=1:GOSUB210:GOT
O310
320 PRINT@I,CHR$(W);:IFD=31ORD=24ORD=25ORD=13ORD=64THEN340
330 IFD=91THENGOSUB360:GOSUB240:GOTO310:ELSEOND-7GOSUB220,210,40
0:GOSUB240:GOSUB230:GOTO310
```

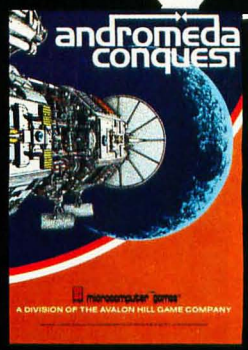
continued on page 60

RISING STARS









What some have called "the best kept secret in the game software industry". Unleashed here are six new software games blending advanced graphics with precise elements of playability, challenge and intense fun for every gamer's tastes. Beyond these six new games are 26 more for you to consider . . . all we believe are the best software values for your money!

LEGIONNAIRE—Real-time simulation of tactical combat in Caesar's time. YOU as Caesar command up to ten legions. Finest **full-color graphics** plus playability way beyond the state of the art! Cassette . . . $35.00

V.C.—Faithfully recreates unconventional conflict in Viet Nam. YOU command chopper and artillery units, and face task of protecting civilian population where the enemy hides among the people. Cassette . . . $20.00
Diskette . . . $25.00

G.F.S. SORCERESS—Sci-Fi adventure game. YOU are Joe Justin trying to clear yourself of a false charge of mutiny and get back to the Galactic Federation Starship "Sorceress". Beautiful full-color manuals provide useful clues. Cassette . . . $30.00 Diskette . . . $35.00

ANDROMEDA CONQUEST—Vast scale space strategy game of galactic colonizing and conquest among unique star systems with strange life forms and alien technologies that provide exciting exploration and battle. Cassette . . . $18.00 Diskette . . . $23.00

MOON PATROL—Arcade Pak™ game of lunar invasion. Beats any quarter-gobbling game around! Four levels of increasing difficulty present new attackers to battle. Fast, furious and fun! Cassette . . . $25.00

TELENGARD—Dungeon adventure in a mysterious underworld with 50 levels of ever-more-complex mazes to explore. Real time fantasy and role-playing game. Using wits, magic and fast thinking, gamers fight monsters and reap valuable rewards. Cassette . . . $23.00
Diskette . . . $28.00

* Trademarks for Apple Computer, Warner Communications, Tandy Corp., International Business Machines and Commodore International Ltd.

Available at finer computer stores everywhere!

AH QUALITY

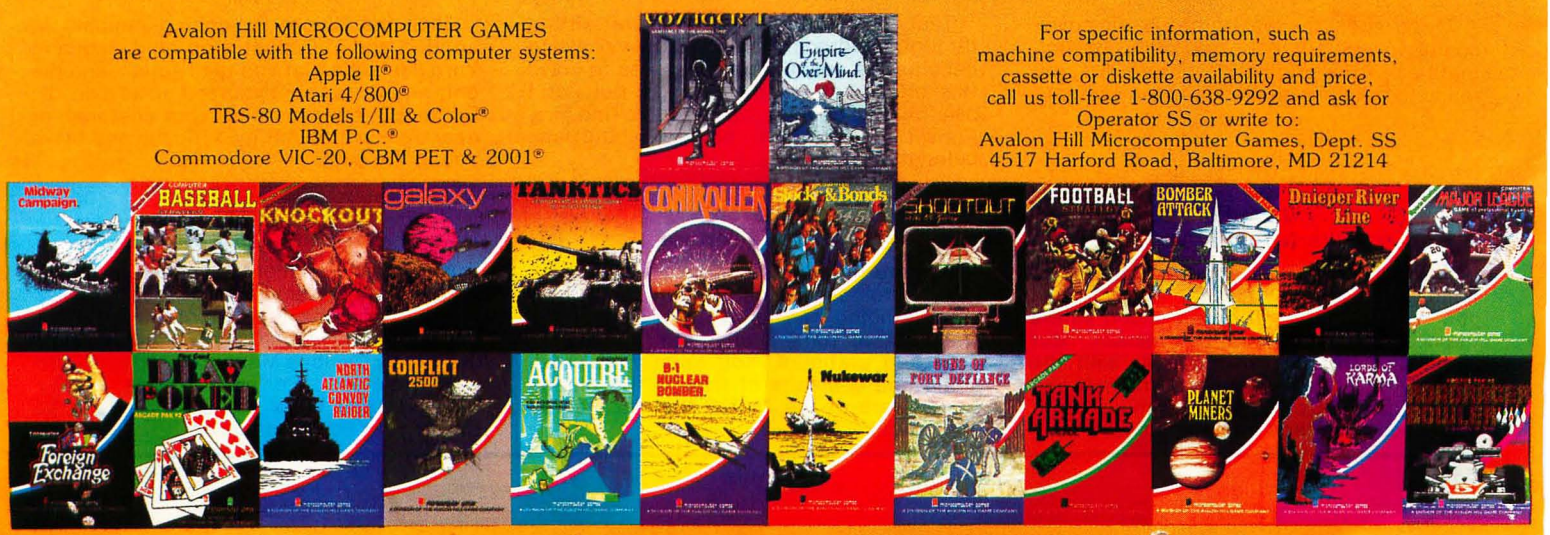
If your favorite dealer fails to have the games you want, call us toll free
1-800-638-9292

microcomputer games

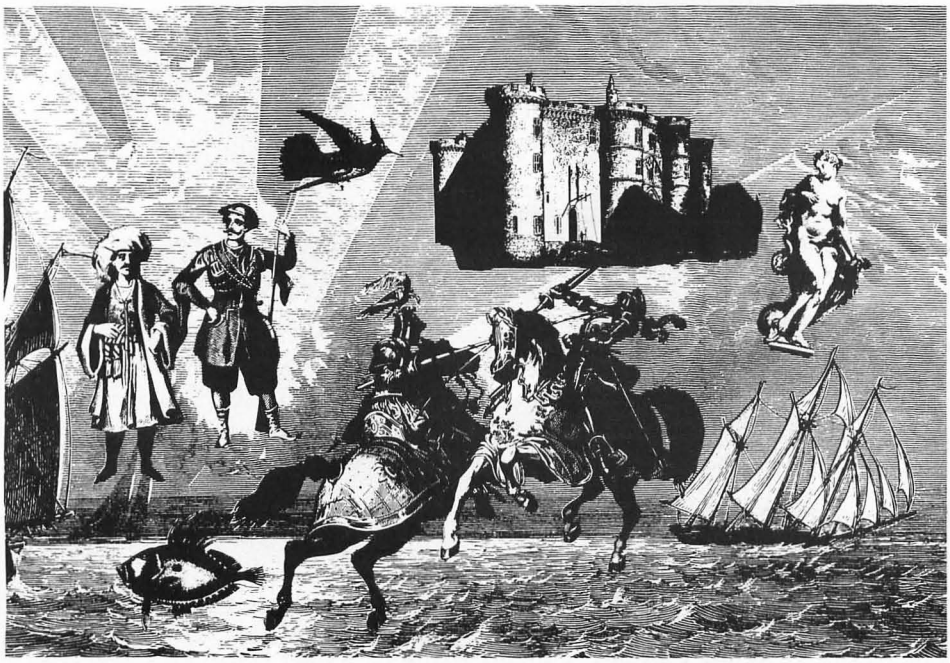
* REGISTERED TRADEMARK OF MICROCOMPUTER GAMES, INC. DIVISION OF
The AVALON HILL Game Company

Avalon Hill MICROCOMPUTER GAMES are compatible with the following computer systems:
Apple II®
Atari 4/800®
TRS-80 Models I/III & Color®
IBM P.C.®
Commodore VIC-20, CBM PET & 2001®

For specific information, such as machine compatibility, memory requirements, cassette or diskette availability and price, call us toll-free 1-800-638-9292 and ask for Operator SS or write to:
Avalon Hill Microcomputer Games, Dept. SS
4517 Harford Road, Baltimore, MD 21214



The Adventure is



August #15
The Mouse That Ate Chicago

We've got four man-eating mice for you, threatening to munch away on the city of Chicago. Each has its own specific weakness, and it's your civic duty to determine what it is and destroy them!

July #14
Robin Hood

There are rich merchants to rob and great deeds to do. But have care, bold Robin! The sheriff of Nottingham longs to see thy neck in a noose. Thou wilt surely need all thy wit and cunning to succeed.

June #13
Arrow One Adventure

You are Adam Trent, a trouble-shooter for the Federation of Space. You descend to an alien planet and make a horrifying discovery, which impels you into a desperate and dangerous quest. This unique science fiction adventure will test your skills and ingenuity.

May 1982
Titanic Adventure

You are the Captain of the Titanic on her maiden voyage. Suddenly a large white object comes into view through the window. Can you avoid the historic collision? If not, can you save the lives of your passengers and crew?

April 1982
Witches' Brew Adventure

You find yourself in an enchanted forest. You must find your way to the castle and rescue the Princess who is chained inside its dungeon. A tightly-woven blend of fantasy, horror, and science fiction, this complex adventure will challenge your wits and ingenuity.

March 1982
James Brand Adventure

The President's life is in danger. As James Brand, you must save his life and destroy the evil Dr. Death. Your life is constantly on the line; each move you make could be your last. "Your assignment, Mr. Brand...."

February 1982
Klondike Adventure

Snow, ice, and bitter cold surround you. Your search for fame and fortune in the northern country will lead you through many perils, but you may also see some familiar faces along the way. This breezy adventure will keep you occupied inside while the winter winds blow outdoors.

January 1982
Windsloe Mansion Adventure

A famous prisoner lies in the dungeon of an old mansion. An underground passage connects the mansion with the Blair house, whose owners will help you to rescue the prisoner. Can you overcome the human and the supernatural creatures who inhabit Windsloe Mansion?

December 1981
Black Hole Adventure

The crew of an interstellar craft discovers the long-lost Deep-Space Probe One, the Cygnus, at the edge of the vortex surrounding an immense black hole. See if you can foil the plans of Dr. Hans Reinhardt.

November 1981
Around the World in Eighty Days

Try to repeat the feat of the classic novel, complete with a balloon and other exciting features of the original adventure. Are you ready to take the challenge? Bon voyage!

October 1981
Crime Adventure

Test your skills as a detective, sifting through hundreds of clues. You may have to become the new Sherlock Holmes to solve this one! Look for the strange, but don't overlook the obvious, as you try to find Mrs. Fenwick and return her to where she belongs.

September 1981
Jack The Ripper Adventure

Jack the Ripper is running rampant in London and you must stop him! Scotland Yard demands that you take action, and the only answer is to set yourself up as a decoy. Be careful how you plan your costume, or dear Jack will laugh hysterically and leave you in the dust!

August 1981
Treasure Island Adventure

You are a hardy adventurer in search of *fame*, fortune, and whatever else you can get. You find yourself on an island where there is rumor of pirate's treasure. But watch out for the evil magician and the underground torture chamber! You may end up in a spot where all roads coming into it are paved with good intentions. . .

July 1981
Alien Adventure

You are the sole survivor of a crew on a mission to deliver a cargo of oil to Earth. A crash landing has left you stranded on a small planet, harshly alien but rich in lead, gold and platinum. You must find provisions and a means of leaving the planet. But beware of the THING that massacred your crew!

June 1981
Arabian Adventure

As Sinbad, the mightiest sailor in ancient Arabia, your mission is to rescue Princess Jasmine from the clutches of the Wizard of Darkness. You will cross the Seven Seas to the deadly Cyclops Mountain, and do battle with skeletons, a one-eyed beast, a hairy tarantula and more monsters who try to thwart your noble pursuit.

Waiting for You...

ADVENTURE OF THE MONTH
MENAGERIE ADVENTURE

Have you ever wondered what it is like to live in a zoo cage? Well, here's your big chance to find out. You find yourself aboard a strange spaceship, a huge floating zoo. You've been snatched from Earth and you're bound for who knows where. Escape is possible, but you will have to be clever. Don't take too long, though. You've seen the cage they are building for you and it is almost ready....

How would you like to go back in time to 19th century London to match wits with Jack the Ripper? Out into space to brave the swirling vortex of a black hole? Into the depths of the ocean, or on a quest to rescue a beautiful princess from the clutches of evil monsters?

You never know where **SoftSide Magazine's Adventure of the Month Club** might take you. But you can be sure that each month you will experience new delights and new challenges as you receive an original adventure on tape or disk, ready to load into your computer.

And now it's even easier for you to join **Adventure of the Month.** A Trial Membership (3 months 3 different Adventures), costs only $29 for Disks, $19 for Cassettes.

Or, choose to become a Charter Member (24 months/24 different Adventures), and you'll save almost half on the individual price of the Adventures you'll receive.

To enter your subscription membership, just fill out the coupon below or the handy postage-free bind-in card found elsewhere in this issue.

Either way, join today! Because you never know where **Adventure of the Month** will take you...

See page 32 for ordering information

Adventure of the Month

6 South Street, Milford NH 03055

Yes, I'm ready to start! Enter my membership subscription to Adventure of the Month.

NEW MEMBERSHIPS NOW AVAILABLE

■ **Check the type of Membership you want:**

☐ Trial **Disk** Membership (3 mo.) $29
☐ Trial **Cassette** Membership (3 mo.) $19

Disks	**Cassettes**
☐ 6-Mo. Member $49	☐ 6-Mo. Member $29
☐ 12-Mo. Member $89	☐ 12-Mo. Member $49
☐ 24 mo. Charter $169	☐ 24 mo. Charter $89

Please send me the ☐ APPLE™ ☐ ATARI® ☐ TRS-80®

■ **Super Disk Three Adventures — ($26 each):**

☐ Arabian, Alien, Treasure Island
☐ Black Hole, Windsloe Mansion, Klondike
☐ Jack the Ripper, Crime, Around the World
☐ James Brand, Witches' Brew, Titanic
☐ Arrow One, Robin Hood, The Mouse That Ate Chicago

Prices for USA only. For foriegn orders, see page 32. Please include exact postage on all foreign orders.

■ **Individual adventures**

(please specify) _____

_____

☐ **Cassette** — $7 each ☐ **Disk** — $10 each

Name _____

Address _____

City/State _____ Zip_____

☐ Check or Money Order (Please enclose this card and your payment in an envelope.)
☐ MasterCard ☐ VISA

Name of Cardholder _____

MC# and Interbank#/VISA# _____

Exp. Date_____Signature _____

Prices subject to change without notice. Apple™, *ATARI® and TRS-80® are registered* trademarks of The Apple Computer Company, Warner Communications and The Tandy Corporation respectively.

```
340 IFD=31THENGOSUB270:GOTO110ELSEIFD=64THEN470ELSEIFD=13THEN460
350 GOSUB270:IFE=0THENOND-23GOSUB420,440:LSETI$(Y)=MID$(S(P),XL,
64):GOTO310ELSEOND-23GOSUB430,450:GOSUB200:GOTO310
```

Cursor Movement Routine.

```
360 IFYT=1ANDY=0THENRETURNELSEIFY>0THENY=Y-1:RETURN
370 GOSUB270:YT=YT-1:GOSUB200:RETURN
380 IFXL=1ANDX=0THENRETURNELSEGOSUB270:XL=XL-1:GOSUB200:RETURN
390 IF(X=63ANDXL+63=XM)ORX+XL=XMTHENRETURNELSEGOSUB270:XL=XL+1:G
OSUB200:RETURN
400 IF(Y=14ANDYT+14=YM)ORY+YT=YMTHENRETURNELSEIFY<14THENY=Y+1:RE
TURN
410 GOSUB270:YT=YT+1:GOSUB200:RETURN
```

Delete Space Sub.

```
420 S(P)=LEFT$(S(P),X+XL-1)+MID$(S(P),X+XL+1)+" ":RETURN
```

Delete Line Sub.

```
430 FORR=PTOYM-1:S(R)=S(R+1):NEXT:S(R)=STRING$(XM,32):RETURN
```

Insert Space Sub.

```
440 S(P)=LEFT$(S(P),X+XL-1)+" "+MID$(S(P),X+XL,-X-XL+XM):RETURN
```

Insert Line Sub.

```
450 IFP<YM-1THENFORR=YMTOP+1STEP-1:S(R)=S(R-1):NEXT:S(R)=STRING$
(XM,32):RETURN:ELSES(YM)=STRING$(XM,32):RETURN
```

Return Key Sub.

```
460 X=0:GOSUB230:IFY+YT<YMANDY<14THENY=Y+1:GOSUB240:GOTO310ELSE3
10
```

Change Controller Mode.

```
470 IFE=1THENE=0:GOTO300ELSEPRINT@986," LINE (SHIFT ->)ADD LINE
(@)RETURN   ";:E=1:GOTO310
```

Move a Line Section

Menu.

```
480 CLS:PRINT@20,"$Section to Move a line$"
490 PRINT@205,"Which should be changed
"TAB(20)"<A> dd or Delete lines
"TAB(20)"<E> xtend or Limit a line width
"TAB(20)"<S> witch two lines
"TAB(20)"<O> verlay a line
"TAB(20)"<SPACEBAR> for Menu":PRINT@914,"<Press the Appropriate
Key>"
500 I=229:GOSUB160:CLS:FORR=1TO4:IFMID$("AESO",R,1)<>CTHENNEXT:G
OTO110ELSEIFR=1THEN550ELSEIFR=2THEN590
```

Exchange and Overlay Routine.

```
510 PRINT@128,"Enter number for First line";:INPUTR:IFR<1ORR>YMT
HEN480
520 PRINT"Enter number for ";:IFC="O"THEN540:ELSEPRINT"Second li
ne";:INPUTR1:IFR1<1ORR1>YMTHEN480
530 R$=S(R):S(R)=S(R1):S(R1)=R$:R$="":GOTO480
540 PRINT"line to be overlayed";:INPUTR1:IFR1<1ORR1>YMTHEN480ELS
ES(R1)=S(R):GOTO480
```

Add or Delete Lines Routine.

```
550 PRINT@260,"Positive number if adding lines, Negative if dele
ting lines":PRINT@849,"There are currently";YM;"lines":PRINT@210
,"How many lines to be added";
560 D=0:INPUTD:IFD=00RYM+D<10RYM+D>90THEN480ELSEIFD<-10ORD>10THE
NPRINT@464,"Limit is ten lines per segment":IFD<0THEND=-10ELSED=
10
570 IFD>0THENR$=STRING$(130,32):FORR=YM+1TOYM+D:S(R)=STRING$(XM,
32):NEXT:R$=""ELSEFORR=YMTOYM+D+1STEP-1:S(R)="":NEXT:X=0:Y=0:YT=
1
580 YM=YM+D:GOTO480
```

Extend or limit line Width.

```
590 X=0:PRINT@18,"Extend or limit line width"::PRINT@266,"Curren
t Line Width is"XM:PRINT@590,;:INPUT"Input New Line width ";R:IF
R<50RR>130ORR=XMTHEN480
600 IFR>XMTHENR$=STRING$(130,32):FORR1=1TOYM:S(R1)=S(R1)+STRING$
(R-XM,32):NEXT:XM=R:R$="":GOTO480ELSEXM=R:FORR1=1TOYM:S(R1)=LEFT
$(S(R1),R):NEXT:GOTO480
```

Print Sketch Section

```
610 CLS:R$="":PRINT@15,"$ Section to Print Sketch $";:PRINT@200,
"Are the Printer parameters set (Y/N) ?";:I=238:GOSUB160:IFC="Y"
THEN710
```

Print Initialization.

```
620 PRINT@128,"What is the Left margin (0-80)";CHR$(31);:INPUTPL
:IFPL<0ORPL>80THENPRINTTAB(15)"Left Margin is set at 12":PL=12
630 PRINT"What is Top Margin (0-30) ";:INPUTPT:IFPT<0ORPT>30THEN
PT=2:PRINTTAB(15)"Top margin is 2"
640 PRINT"   Is there any Graphics (Y/N) ?";:GOSUB190:IFC="Y"THE
NP1$="Y"ELSEP1$="N"
650 PE=0:PRINT"   Is the Printer, a Epson (Y/N) ?";:GOSUB190:IFC
<>"Y"THEN700
660 PE=32:PRINTP$"Compressed (Y/N) ?";:GOSUB190:IFC="Y"THENR$=CH
R$(15)ELSER$=CHR$(18)
670 PRINTP$"Emphasized (Y/N) ?";:GOSUB190:IFC="Y"THENR$=R$+CHR$(
27)+"E"ELSER$=R$+CHR$(27)+"F"
680 PRINTP$"Double-Strike (Y/N) ?";:GOSUB190:IFC="Y"THENR$=R$+CH
R$(27)+"G"ELSER$=R$+CHR$(27)+"H"
690 R$=R$+CHR$(27)+"8"
700 PS=0:PRINT:PRINT" Should the lines be Spaced (Y/N) ?";:GOSUB
190:IFC="Y"THENPRINT" How many lines to be inserted (0-9) ?";:GO
SUB170:PS=VAL(C):PRINTPS
710 PRINT@968,"Press <SPACEBAR> when top is aligned";:I=1004:GOS
UB160:IFC<>" "THEN110
```

Print Sketch.

```
720 IFPT>0THENFORR=1TOPT:LPRINT:NEXT
730 IFP1$="N"THEN770
```

Graphic Print Section.

```
740 R=1:LPRINTTAB(PL)R$;:GOSUB750:IFYM=1THEN110ELSEFORR=2TOYM:LP
RINTTAB(PL)"";:GOSUB750:IFINKEY$=""THENNEXT:GOTO110ELSE110
750 FORR1=1TOXM:P=ASC(MID$(S(R),R1,1)):IFP<128THENLPRINTCHR$(P);
:ELSEIFP>127ANDP<192THENLPRINTCHR$(P+PE);:ELSEIFP>191THENLPRINTC
HR$(P-32);
760 NEXT:LPRINT:GOSUB780:RETURN
```

No-Graphics Print Section.

```
770 LPRINTTAB(PL)R$S(1):GOSUB780:IFYM=1THEN110ELSEFORR=2TOYM:LPR
INTTAB(PL)S(R):GOSUB780:IFINKEY$=""THENNEXT:GOTO110ELSE110
780 IFPS>0THENFORR1=1TOPS:LPRINT:RETURNELSERETURN
```

Graphic Writer

```
790 CLS:W=0:GOSUB200:PRINT@960,"X>      Y>";:GOSUB1140:GOSUB230:G
OSUB240:PRINT@974,CHR$(170)" "CHR$(149);:G=0:L=0
800 P=Y+YT:I=X+(Y¥64):W1=PEEK(I+15360):IFW>0THENL=1
810 T1=X¥2+T:T2=Y¥3+U:GOSUB1110:PO=POINT(T1,T2):GOSUB250:PRINT@I
,CHR$(W1);:GOSUB260
```

Input Processing.

```
820 D=ASC(C):IFD=91ORD>7ANDD<11THEN1000
830 IFD=13ANDG=1THEN:Y=T4/3:X=T3/2:U=T4-(Y¥3):T=T3-(X¥2):GOTO800
840 D=D-48:IFD>0ANDD<10THEN940
850 IFC=" "THENL=1:IFW=2THENRESET(T1,T2):GOTO800:ELSESET(T1,T2):
GOTO800
860 IFASC(C)=31THENGOSUB270:GOTO110
870 IFC="W"THENW=W+1:IFW=3THENW=0
880 IFC="@"THENW=W-1:IFW=-1THENW=2
890 IFC="J"THENG=1:PRINT@990,STRING$(33,32);:PRINT@INT(T4/3)¥64+
INT(T3/2),C2;:G=0:GOTO810
900 IF(C="C"ORC="A"ORC="S")ANDG=0THENL=1:G=1:C1=C:C2=CHR$(W1):GO
TO1220
910 IFC="D"THENL=1:IFW<2THENPRINT@I,CHR$(191);:GOTO800:ELSEPRINT
@I,CHR$(128);:GOTO800
920 IF(C="L"ORC="K")ANDG=1THENIFC1="A"THEN1210:ELSEIFC1="C"THEN1
150ELSEIFC1="S"THEN1330
930 GOSUB1140:GOTO800
```

Hyper-Cursor Movement.

```
940 PRINT@980,"Press Direction or <Spacebar> to Abort     ";:GOS
UB250:PRINT@I,CHR$(W1);:GOSUB260
950 R=ASC(C$):IFR=32AND(R<8ORR>10)ANDR<>91THEN980
960 IFR=8THENFORR1=1TOD:GOSUB220:NEXTELSEIFR=9THENFORR1=1TOD:GOS
UB210:NEXT
970 IFR=10THENFORR1=1TOD:GOSUB400:GOSUB240:NEXTELSEIFR=91THENFOR
R1=1TOD:GOSUB360:GOSUB240:NEXT
980 PRINT@980,STRING$(40,32);:GOSUB1140:IFG=1THENPRINT@990,;:IFC
1="A"THENPRINT"<L>ine";:ELSEIFC1="S"THENPRINT"<S>quare";:ELSEIFC
1="C"THENPRINT"<C>ircle";
990 IFG=1THENPRINT" Drawing, <J> to abort";:GOTO800ELSE800
```

Cursor Movement.

```
1000 IFD=8ORD=9THENK=D¥2-17:T=T+K:GOSUB1030:GOSUB230:GOTO800
1010 IFD=10THENK=+1ELSEK=-1
1020 U=U+K:GOSUB1070:GOSUB240:GOTO800
```

continued on page 62

Horizontal Movement.

```
1030 IF(T<00RT>1)AND(X+K<00RX+K>630RX+K+1>XM)THENGOSUB1060:IFD=8
THENGOSUB220:RETURN:ELSEGOSUB210:RETURN
1040 IFT>-1ANDT<2THENRETURN:ELSEX=X+K
1050 IFK=-1THENT=1:RETURN:ELSET=0:RETURN
1060 IF(K=-1ANDXL<>1)ORX+XL=XMTHENT=1:RETURN:ELSET=0:RETURN
```

Vertical Movement.

```
1070 IF(U<00RU>2)AND(Y+K<00RY+K>140RY+K+1>YM)THENGOSUB1100:IFD=9
1THENGOSUB360:RETURN:ELSEGOSUB400:RETURN
1080 IFU>-1ANDU<3THENRETURNELSEY=Y+K
1090 IFK=-1THENU=2:RETURN:ELSEU=0:RETURN
1100 IFP<>1AND(K=-10RP=YM)THENU=2:RETURN:ELSEU=0:RETURN
```

Cursor Position inside box.

```
1110 IFT=0THENIFU=0THENR=129ELSEIFU=1THENR=132ELSER=144
1120 IFT=1THENIFU=0THENR=130ELSEIFU=1THENR=136ELSER=160
1130 PRINT@975,CHR$(R);:RETURN
```

Cursor Mode Status.

```
1140 IFW=0THENPRINT@980,"Pos. Only";:RETURNELSEIFW=1THENPRINT@98
0,"Set      ";:RETURNELSEPRINT@980,"Reset    ";:RETURN
```

Circle Drawer init. and execution.

Calculate for degree (1-90).

```
1150 PRINT@INT(T4/3)*64+INT(T3/2),C2;:RA!=SQR((T1-T3)[2+(T2-T4)[
2):R3=(RA!<10)*-10+(RA!>10)*-4+(RA!>48)*-2:FORR=1TO90STEPR3:RC!=
R*.01745329
1160 P1!=COS(RC!)*RA!:P2!=SIN(RC!)*RA!:R1=ABS(T3+P1!):R2=ABS(T4+
P2!):GOSUB1190:SET(R1,R2):R1=ABS(T3-P1!):R2=ABS(T4-P2!):GOSUB119
0:SET(R1,R2)
```

Draw Circle:four times for each 90:deg.

```
1170 R1=ABS(T3-P2!):R2=ABS(T4+P1!):GOSUB1190:SET(R1,R2):R1=ABS(T
3+P2!):R2=ABS(T4-P1!):GOSUB1190:SET(R1,R2):
1180 NEXTR:IFC<>"L"THENG=0:PRINT@990,STRING$(33,32);:GOTO800:ELS
E800
1190 IFR2>44THENR2=ABS(44-(R2-44))
1200 IFR1>127THENR1=ABS(127-(R1-127)):RETURNELSERETURN
```

Line Drawer init. and execution.

```
1210 IFT1<>T3ORT2<>T4THEN1260ELSEG=0:PRINT@990,STRING$(33,32);:G
OTO800
1220 T3=T1:T4=T2:PRINT@990,;:IFC1="A"THENPRINT"<L>ine";:ELSEIFC1
="S"THENPRINT"<S>quare";:ELSEIFC1="C"THENPRINT"<C>ircle";
1230 PRINT" Drawing, <J> to abort";:GOTO810
```

Marker and Cursor Flash Subroutine for all Draw routines.

```
1240 IFG=1THENRESET(T3,T4):RETURN:ELSERETURN
1250 IFG=1THENSET(T3,T4):RETURN:ELSERETURN
```

Line Draw Routine.

```
1260 IFW=2THENRESET(T1,T2):RESET(T3,T4):ELSESET(T1,T2):SET(T3,T4
)
1270 IFABS(T2-T4)>ABS(T1-T3)THEN1300
1280 FORX1=T1TOT3STEPSGN(T3-T1):Y1=T2+(T2-T4)/(T1-T3)*(X1-T1)
1290 GOSUB1320:NEXT:IFC<>"L"THENG=0:PRINT@990,STRING$(33,32);:GO
TO800ELSE800
1300 FORY1=T2TOT4STEPSGN(T4-T2):X1=T1+(Y1-T2)*(T3-T1)/(T4-T2)
1310 GOSUB1320:NEXTY1:IFC<>"L"THENG=0:PRINT@990,STRING$(33,32);:
GOTO800ELSE800
1320 IFW=2THENRESET(X1,Y1):RETURN:ELSESET(X1,Y1):RETURN
```

Square Drawer init. and execution.

```
1330 IFT1=T3ANDT2=T4THENG=0:PRINT@990,STRING$(33,32);:GOTO800
1340 FORR=T1TOT3STEPSGN(T3-T1):GOSUB1370:NEXT
1350 FORR=T2TOT4STEPSGN(T4-T2):GOSUB1380:NEXT
1360 IFC<>"L"THENG=0:PRINT@990,STRING$(33,32);:GOTO800ELSE800
1370 IFW=2THENRESET(R,T2):RESET(R,T4):RETURNELSESET(R,T2):SET(R,
T4):RETURN
1380 IFW=2THENRESET(T1,R):RESET(T3,R):RETURNELSESET(T1,R):SET(T3
,R):RETURN
```

Save Sketch on Disk

```
1390 CLS:PRINT@15,"*Section to Save Sketch to Disk*":PRINT@520,"
What is the Name of the Sketch to be Saved":PRINTTAB(10)"(Type S
TOP if incorrect command)"
1400 PRINTSTRING$(12,".");CHR$(29);:LINEINPUTR$
1410 IFR$="STOP"THEN110ELSEIFLEN(R$)<1ORLEN(R$)>12THENPRINT"The
Format of the Sketch is incorrect":GOSUB180:GOTO1390
```

Output to Disk, Name of Sketch.

```
1420 OPEN"O",1,R$:PRINT@918,"Lines to go"
```

Output to Disk, Sketch with a" > " in front of it.

```
1430 FORR=1TOYM:PRINT#1,">"+S(R):PRINT@930,YM-R:NEXT:CLOSE1:A1=R
$:GOTO1730
```

Load Sketch from Disk

```
1440 CLS:PRINT@15,"*Section to Load Sketch from Disk*"
1450 R$="":PRINT@452,"What is the Name of the Program (type STOP
, if mistake)":PRINTSTRING$(12,".");CHR$(29);:LINEINPUTR$
1460 IFR$="STOP"THEN110ELSEIFLEN(R$)<1ORLEN(R$)>12THENPRINT@712,
"Illegal Program name":FORR=1TO2000:NEXT:GOTO1440
1470 OPEN"I",1,R$:PRINT@118,"Line Num.";:R3=1
```

Input Sketch till < EOF > marker.

```
1480 IFEOF(1)THEN1490ELSELINEINPUT#1,A$:S(R3)=MID$(A$,2):PRINT@5
76,S(R3);:PRINT@114,R3;:R3=R3+1:GOTO1480
```

Initialize Sketch size and boundaries.

```
1490 Y=0:X=0:YT=1:XL=1:YM=R3-1:XM=LEN(S(1)):CLOSE1:GOTO110
```

Quit, exit the program

```
1500 CLS:PRINTCHR$(21)CHR$(22):PRINT"End of Graphic Writer":STOP
:PRINTCHR$(21)CHR$(22):GOTO110
```

continued on page 66

If You Sell Small Business or Personal Computer Systems, Word Processing, Software, Media & Supplies, or Computer Services...

Have We Got An Offer For You!

Why Prospect?
The answer is obvious. Without prospecting... No Business. But, there's a better way...COMPUTER SHOWCASE EXPO!

This proven series of regional, end-user expositions has brought thousands of prospects, with needs in mind and buying dollars in hand.

THE INTERFACE GROUP, the World's largest producers of Computer Shows, offers you instant availability to a wide variety of end-users in such professions as:

- **Doctors • Lawyers • Accountants • Retailers**
- **Corporate Managers • Many Other Professionals**

These face-to-face contacts are in such businesses as:

- **Banking • Health Care • Insurance • Education**
- **Government • Real Estate • Manufacturing**
- **Distribution**

If you sell computers, or computer systems, as a solution for a variety of Small Business problems, COMPUTER SHOWCASE EXPO offers you more prospects in a few days than you could possibly see in months!

For more information on how you can reach the market you need, call THE INTERFACE GROUP, toll free at **(800) 225-4620,**
(In Massachusetts, call 617-879-4502)

FALL 1982
New York
Sept. 23-25, 1982
San Francisco
Sept. 30-Oct. 2, 1982
South Florida
Oct. 28-30, 1982
Atlanta
Oct. 28-30, 1982
Chicago
Nov. 4-7, 1982
Los Angeles
Nov. 18-20, 1982

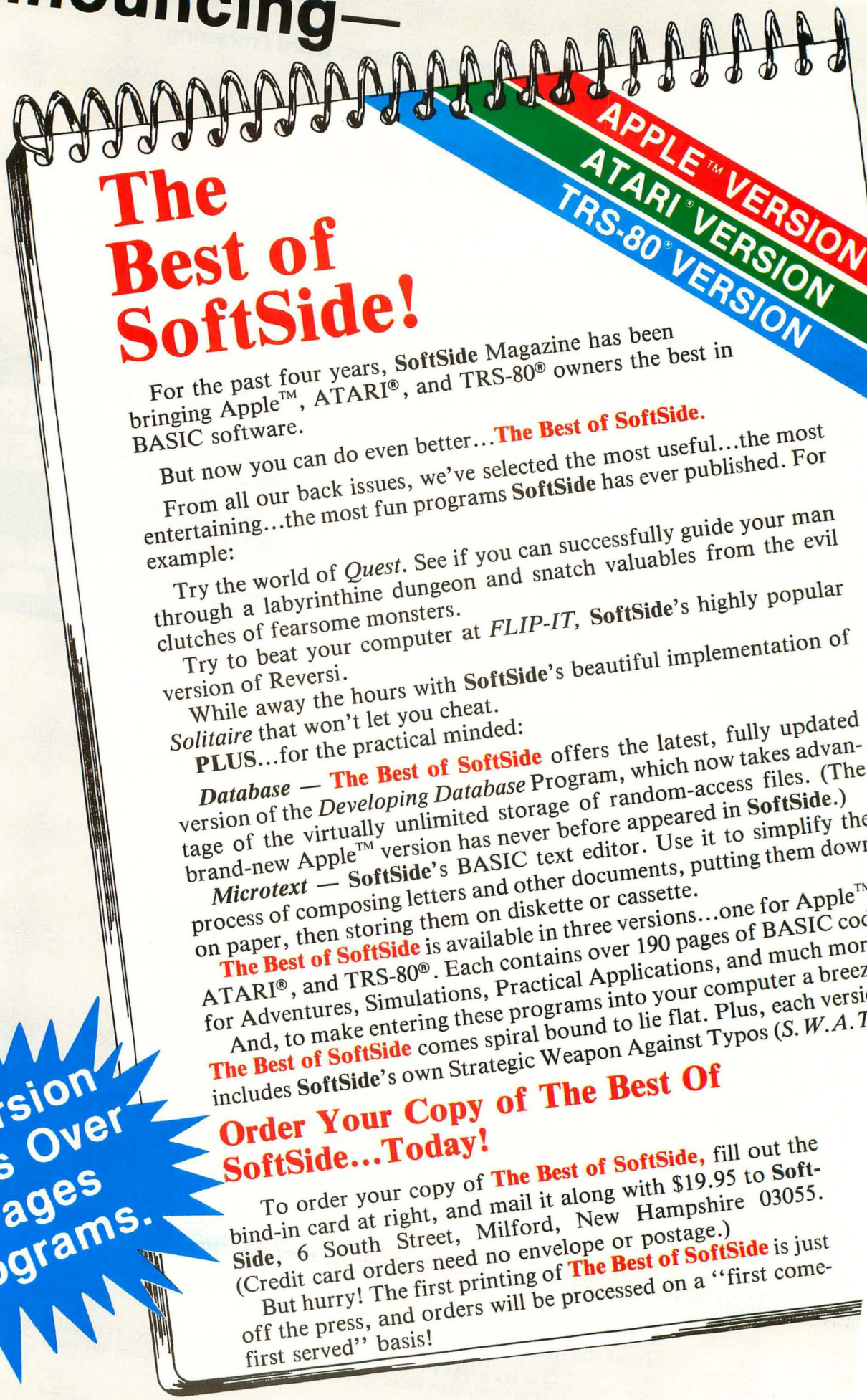
SPRING 1983
Phoenix
Feb. 3-6, 1983
Atlanta
March 24-27, 1983
Chicago
April 7-10, 1983
St. Louis
April 21-23, 1983
South Florida
April 28-May 1, 1983
Anaheim
May 5-8, 1983
Washington, DC
May 5-8, 1983
Boston
May 12-14, 1983
San Diego
May 12-15, 1983
Houston
May 19-22, 1983
Seattle
June 2-4, 1983
Kansas City
June 9-12, 1983
Milwaukee
June 16-19, 1983

FALL 1983
Detroit
Sept. 22-25, 1983
New York
Sept. 22-25, 1983
San Francisco
Sept. 29-Oct. 2, 1983
Philadelphia
Oct. 6-9, 1983
Portland, OR
Oct. 13-15, 1983
Pittsburgh
Oct. 20-23, 1983
South Florida
Oct. 27-29, 1983
Denver
Nov. 3-6, 1983
Los Angeles
Nov. 10-13, 1983
Chicago
Nov. 18-20, 1983

Another Exposition from THE INTERFACE GROUP
Producers of: THE COMPUTER SHOWCASE EXPOS (Nationwide), COMDEX/SPRING, COMDEX/FALL, COMDEX/EUROPE, INTERFACE, THE FEDERAL DP EXPO

160 Speen Street, Framingham, MA 01701 • (617) 879-4502, Toll Free (800) 225-4620, TELEX - 951176, TWX - 710-380-7645
Regional Office: Suite 121, 4700 North State Road 7, Fort Lauderdale, FL 33319 • (305) 484-6800

Announcing—



The Best of SoftSide!

For the past four years, **SoftSide** Magazine has been bringing Apple™, ATARI®, and TRS-80® owners the best in BASIC software.

But now you can do even better…**The Best of SoftSide.**

From all our back issues, we've selected the most useful…the most entertaining…the most fun programs **SoftSide** has ever published. For example:

Try the world of *Quest*. See if you can successfully guide your man through a labyrinthine dungeon and snatch valuables from the evil clutches of fearsome monsters.

Try to beat your computer at *FLIP-IT*, **SoftSide**'s highly popular version of Reversi.

While away the hours with **SoftSide**'s beautiful implementation of *Solitaire* that won't let you cheat.

PLUS…for the practical minded:

Database **— The Best of SoftSide** offers the latest, fully updated version of the *Developing Database* Program, which now takes advantage of the virtually unlimited storage of random-access files. (The brand-new Apple™ version has never before appeared in **SoftSide**.)

Microtext **— SoftSide**'s BASIC text editor. Use it to simplify the process of composing letters and other documents, putting them down on paper, then storing them on diskette or cassette.

The Best of SoftSide is available in three versions…one for Apple™, ATARI®, and TRS-80®. Each contains over 190 pages of BASIC code for Adventures, Simulations, Practical Applications, and much more.

And, to make entering these programs into your computer a breeze, **The Best of SoftSide** comes spiral bound to lie flat. Plus, each version includes **SoftSide**'s own Strategic Weapon Against Typos (*S.W.A.T.*).

Order Your Copy of The Best Of SoftSide…Today!

To order your copy of **The Best of SoftSide,** fill out the bind-in card at right, and mail it along with $19.95 to **SoftSide**, 6 South Street, Milford, New Hampshire 03055. (Credit card orders need no envelope or postage.)

But hurry! The first printing of **The Best of SoftSide** is just off the press, and orders will be processed on a "first come-first served" basis!

Each Version Contains Over 190 Pages Of Programs.

Win $500!

The First National Computer Owner Survey

50 Second Prizes of $10 each!

In order to keep developing and bringing you very special hardware, software and publications, we've been commissioned to find out what you, the customer, wants and needs.

In addition to contributing to the computer owners' data base, you have a chance to win $500... just for filling out this survey.

JUST TAKE A FEW MINUTES, ANSWER EVERY APPLICABLE QUESTION - YOU MUST TO BE ELIGIBLE - AND MAIL TO US NO LATER THAN OCTOBER 31, 1982. FOR 20¢ YOU COULD WIN HUNDREDS.

Entrants must be computer owners or users and answer every applicable question. A random drawing, eligibility approval and list of winners will be verified by a Notary Public. Winners will be notified by Dec. 31, 1982. Grand Prize winner gives IRV Brechner Enterprises the right to use name and photo in future surveys and advertising. No purchase necessary. Limit one entry per person. Entries must be postmarked no later than Oct. 31, 1982 and reach us by Nov. 15, 1982. Prizes include one cash award of $500, and 50 cash prizes of $10 each. All survey entries become property of IRV Brechner Enterprises; none will be returned. All prizes will be awarded by Nov. 31, 1982. All Federal, State and Local taxes are responsibility of the winner. This contest void where prohibited by law. For a prize winner list, send a self-addressed stamped envelope to IRV Brechner Enterprises, Box 264WOB, West Orange, N.J. 07052.

NAME _____ (1) ADDRESS _____ (2)

CITY/STATE/ZIP _____ (3) AGE____ (4) SEX____ (5) Circle: SINGLE (6) MARRIED (7) # CHILDREN ____ (9)

COMPUTERS(S) YOU OWN OR USE (Circle) (10) APPLE (11) ATARI (12) TRS-80 (13) IBM PC (14) HEWLETT-PACKARD (15) HEATH (16) ZENITH (17) OSBORNE

(18) XEROX (19) DEC (20) TEXAS INST. (21) NORTH STAR (22) COMMODORE (23) VECTOR (24) CROMEMCO (25) OTHER____ (26) OWNED HOW LONG____

DISK DRIVES____ (27) OWN HARD DISK?____ (28) OWN MODEM?____ (29) PRINTER BRAND____ (30) MONITOR BRAND____ (31)

APPROX. # DISKETTES OWNED_____ (32) BRAND PREFERENCE_____ (33) OPERATING SYSTEM(S)_____ (34)

AMOUNT OF MEMORY (Circle) (35) 8K (36) 16K (37) 24K (38) 32K (39) 48K (40) 64K (41) 128K (42) MORE____ # DISKETTES PURCHASED/YEAR____ (43)

LANGUAGES YOU PROGRAM WITH OR USE (Circle) (44) BASIC (45) FORTRAN (46) COBOL (47) MACHINE (48) ASSEMBLER (49) MONITORS

(50) FORTH (51) ALGOL (52) PASCAL (53) C (54) ADA (55) APL (56) LISP (57) CAI DO YOU WRITE YOUR OWN SOFTWARE (Circle) (58) YES (59) NO

APPROXIMATE NUMBER OF SOFTWARE PROGRAMS YOU OWN PER CATEGORY:

(60) EDUCATION____ (61) BUSINESS____ (62) GAMES____ (63) SCIENTIFIC____

(64) HOBBY____ (65) HOME USE____ (66) OTHER____

YOUR PROFESSION_____ (67) PRIMARY USE FOR YOUR COMPUTER_____ (68)

APPROX ANNUAL INCOME (Opitonal)_____ (69) MOST RECENT EDUCATION LEVEL (Circle) (70) HIGH SCHOOL (71) SOME COLLEGE

(72) COLLEGE GRADUATE (73) SOME GRADUATE SCHOOL (74) MASTER'S DEGREE (75) OTHER_____

WHICH PUBLICATIONS DO YOU SUBSCRIBE TO OR READ REGULARLY (Circle all that apply) (76) APPLE ORCHARD (77) BYTE (78) CALL-APPLE

(79) COMPUTE (80) COMPUTERWORLD (81) CREATIVE COMPUTING (82) DESKTOP COMPUTING (83) INTERFACE AGE (84) INFOWORLD

(85) MICROCOMPUTING (86) MICRO (87) MICROSYSTEMS (88) NIBBLE (89) PEELINGS II (90) POPULAR COMPUTING (91) PERSONAL COMPUTING

(92) SOFTSIDE (93) OTHERS_____ (94) _____ (95) _____ (96) _____ (97) _____

APPROX # COMPUTER BOOKS OWNED_____ (98)

WHAT NEW PRODUCTS, IDEAS, HARDWARE, SOFTWARE, PUBLICATIONS, ETC. DO YOU WISH TO SEE COME ABOUT?
Please be specific and use additional paper if necessary. Staple to survey when completed.

Signature _____

Mail all completed surveys by Oct. 31, 1982 to:
NATIONAL COMPUTER OWNERS' SURVEY • BOX 264WOB • WEST ORANGE, N.J. 07052

*Apple, Atari, TRS-80, IBM PC, Hewlett-Packard, Heath, Zenith, Osborne, Xerox DEC, Texas Inst., North Star, Commodore, Vector & Cromemco are all registered trademarks.

Imitate Imaginary char.

```
1505 IFPEEK(293)<>73THENCLS:PRINT"This Section will not work wit
h a Model I.":GOSUB180:GOTO110
```

Display range of Images and characters.

```
1510 I1=I:CLS:PRINT@14,"# Characters that can be Imitated #":PRI
NT" !"CHR$(34)"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRS
TUVWXYZ":FORR=92TO95:PRINTCHR$(R);:NEXT:PRINT"abcdefghijklmnopqr
stuvwxyz";:FORR=123TO127:PRINTCHR$(R);:NEXT:PRINT
1520 FORR=192TO255:PRINTCHR$(R);:NEXT
1530 PRINT@455,"# Characters used on <S>creen via <K>eyboard Ent
ry #":FORR=1TO13:PRINT" K S ";:NEXT:PRINTCHR$(8);:R1=0:FORR=32TO
63:GOSUB1650:NEXT:FORR=65TO90:GOSUB1650:NEXT:FORR=97TO122:GOSUB1
650:NEXT
1540 B=CHR$(95):I=157:PRINT@258,"Move Cursor by Arrows to Charac
ter and then Press <ENTER>";
```

Get image char. by inkey$ cursor move.

```
1550 W=PEEK(I+15360):A=CHR$(W):GOSUB160:PRINT@I,A;:IFASC(C)<>13A
ND(ASC(C)<80RASC(C)>10)ANDASC(C)<>91THEN1550ELSEIFASC(C)=13THENP
RINT@384,CHR$(W)" is the Image Character":GOTO1600
1560 IFASC(C)=8THENIFI=640RI=1280RI=192THEN1550ELSEI=I-1:GOTO155
0
1570 IFASC(C)=9THENIFI=1220RI=1620RI=255THEN1550ELSEI=I+1:GOTO15
50
1580 IFASC(C)=10THENIFI>191THEN1550ELSEIFI>98ANDI<128THENI=I+128
:GOTO1550:ELSEI=I+64:GOTO1550
1590 IFASC(C)=91THENIFI<128OR(I>250ANDI<256)THEN1550ELSEIFI<251A
NDI>226THENI=I-128:GOTO1550:ELSEI=I-64:GOTO1550
1600 PRINT@280,"to Keyboard Character to be used      ":PRINT@34
0," Then Press <ENTER>";:I=735
```

Get keyboard char. and change key. array.

```
1610 W1=PEEK(I+15360):A=CHR$(W1):GOSUB160:PRINT@I,A;:IFASC(C)<>1
3AND(ASC(C)<80RASC(C)>10)ANDASC(C)<>91THEN1610:ELSEIFASC(C)=13TH
EN1640
1620 IFASC(C)=8THENIFI=961ORI=897ORI=833ORI=769ORI=705ORI=641ORI
=577ORI=513THEN1610ELSEI=I-5:GOTO1610:ELSEIFASC(C)=9THENIFI=957O
RI=893ORI=829ORI=765ORI=701ORI=637ORI=573ORI=986THEN1610ELSEI=I+
5:GOTO1610
1630 IFASC(C)=10THENIFI<926THENI=I+64:GOTO1610:ELSE1610:ELSEIFAS
C(C)=91THENIFI<641THEN1610ELSEI=I-64:GOTO1610
1640 PRINT@256,STRING$(190,32):PRINT@I+2,CHR$(W);:D(W1)=W:PRINT@
329,CHR$(W)" is now a Image of the Keyboard Character ";CHR$(W1)
;:PRINT@461,"Is there any more to be changed (Y/N) ?";:GOSUB170:
IFC="Y"THENPRINT@329,STRING$(171,32);:GOTO1540ELSEI=I1:GOTO110
1650 R1=R1+1:IFR1=14THENR1=1:PRINTCHR$(8);" "CHR$(R)" "CHR$(D(R)
)" ";:RETURNELSEPRINT" "CHR$(R)" "CHR$(D(R))" ";:RETURN
```

Display file of Sketches

```
1660 CLS:PRINT@10,"Section for Directory of Sketches":OPEN"I",1,
"DIRWRITR/DIR":R=128
1670 IFEOF(1)ORR>950THEN1700
1680 INPUT#1,A:PRINT@R,A;:R=R+15:IFR-(INT(R/64)*64)>56THENR=(INT
(R/64)*64)+64
1690 GOTO1670
1700 PRINT@970,"Press <SPACEBAR> to return to Menu";:GOSUB170:CL
OSE:GOTO110
```

Add Sketch Name to file.

```
1710 CLS:PRINT@12,"Section to add a Sketch Name to a directory":
PRINT@139,"Type name of program to add to Directory
(Type STOP if mistake)"
1720 PRINT@274,"......../...":PRINT@274,;:LINEINPUTA1:IFLEN(A1)>
12THENPRINT@192,CHR$(31):GOTO1720:ELSEIFLEN(A1)<1THEN110ELSEIFA1
="STOP"THEN110
1730 OPEN"I",2,"DIRWRITR/DIR":R1=0:IFEOF(2)THEN1770
1740 R1=R1+1:INPUT#2,A$:IFA$=A1$THENCLOSE:PRINT@450,"The File na
me already exists":GOSUB180:GOTO110
1750 IFR1>60THENCLOSE:PRINT@450,"The Directory is full, Delete s
ome files first":GOSUB180:GOTO170
1760 IFEOF(2)THEN1780ELSE1740
1770 CLOSE:OPEN"O",2,"DIRWRITR/DIR":GOTO1790
1780 CLOSE:OPEN"E",2,"DIRWRITR/DIR":
1790 PRINT@840,"Placing the name into the file":PRINT#2,A1$:CLOS
E:GOTO110
```

Delete Sketch from Dir.

```
1800 CLS:PRINT@10,"Section to delete Sketches from Directory"
1810 PRINT@384,"What is the name of the Sketch (STOP to abort)":
PRINT"........"STRING$(8,24);:LINEINPUTA$:IFLEN(A$)>12THENPRINT@
448,CHR$(31);:GOTO1810ELSEIFLEN(A$)<1ORA$="STOP"THEN110
```

Transfer Dir. to temporary file.

```
1820 OPEN"I",2,"DIRWRITR/DIR":OPEN"O",3,"DIRTEMP/DIR":R=0
1830 IFEOF(2)THEN1850ELSEINPUT#2,A1:IFA1=ATHENR=1:PRINT@650,"Del
eted Sketch from Directory":GOTO1830
1840 PRINT#3,A1:GOTO1830
1850 CLOSE:IFR=0THENPRINT@650,"The file was not found":FORR=1TO2
000:NEXT:GOTO1880ELSEOPEN"I",3,"DIRTEMP/DIR":OPEN"O",2,"DIRWRITR
/DIR"
```

Transfer data back to original.

```
1860 PRINT@850,"Reorganizing the Directory file"
1870 IFEOF(3)THEN1880ELSEINPUT#3,A:PRINT#2,A:GOTO1870
1880 CLOSE:GOTO110
```

Load Ascii File or program.

```
1890 CLS:PRINT@12,"Section to Load a Ascii Program into Memory"
1900 PRINT@200,"The program must be saved with the ',A' option."
:PRINT@520,"......./...":PRINT@520,;:LINEINPUTA
1910 IFA="STOP"ORLEN(A)<1THEN110ELSEIFLEN(A)>12THENPRINT@680,"Il
legal number of characters":FORR=1TO2000:NEXT:GOTO1890
```

Load contents into lines :64 chars. wide.

```
1920 OPEN"I",1,A:R=0
1930 IFEOF(1)THEN1960ELSELINEINPUT#1,R$
1940 R=R+1:S(R)=LEFT$(R$,64):S(R)=S(R)+STRING$(64-LEN(S(R)),32):
IFLEN(R$)>64THENR$=MID$(R$,65):GOTO1940
1950 GOTO1930
1960 YM=R:XM=64:R$="":CLOSE:GOTO110
```

Save an ASCII Program to Disk.

```
1970 CLS:PRINT@14,"Section to save Ascii Program to disk"
1980 PRINTTAB(10)"It will erase prior contents of the file you s
pecify"
1990 PRINT@340,"...../...":PRINT@340,;:LINEINPUTA1:IFA1="STOP
"THEN110EL SEIFLEN(A1)>12ORLEN(A1)<1THENPRINT@800,"Illegal number
 of characters":FORR=1TO1000:NEXT:GOTO110
```

Open Buffer.

```
2000 R=1:OPEN"O",1,A1:A=""
```

Check for invalid line numbers.

```
2010 IFVAL(S(R))>0ANDVAL(S(R))<65530ANDINSTR(S(R)," ")<7THENA=S(
R):ELSE2050
2020 R=R+1:IFR>YMORS(R)=STRING$(XM,32)THEN2040ELSEIFLEN(A)+LEN(S
(R))>254THEN2060
2030 IFVAL(S(R))>R1ANDVAL(S(R))<65000ANDINSTR(S(R)," ")<7THENPRI
NT#1,A:A="":GOTO2010ELSEA=A+S(R):GOTO2020
2040 PRINT#1,A:CLOSE:GOTO1730
2050 PRINT@596,"ERROR IN LINE NUMBERING":CLOSE:GOSUB180:GOTO110
2060 PRINT@596,"ERROR IN LENGTH OF PROGRAM LINE":CLOSE:GOSUB180:
GOTO110
```

Error Section of Program.

Error handling for Out of String Space error.

```
2070 R3=ERR/2+1:IFR3=14THENCLS:R$="":PRINT@334,CHR$(23)"Not Enou
gh Memory":PRINT@966,"Press <SPACEBAR> for Menu";
2080 IFERL=600THENFORR=1TOR1:S(R)=LEFT$(S(R),XM):NEXT:GOSUB180:R
ESUME480
2090 IFERL=100THENFORR=1TOYM:S(R)="":NEXT:GOSUB180:RESUME80
2100 IFERL=570THENFORR=YM+1TOYM+D:S(R)="":NEXT:GOSUB180:RESUME11
0
```

Improper Input from Disk or Operator.

```
2110 IFERL=1420ORERL=1470ORERL=1430ORERL=1480THENPRINT@704,CHR$(
31);:CMD"E":PRINT@980,"Press <SPACEBAR> for menu";:CLOSE:GOSUB18
0:RESUME110
```

If there is no Dir. File.

```
2120 IFERL=1660RESUME1700
2130 IFERL=1730RESUME1770
```

If trying to load non-existent file.

```
2140 IFERL=1920THENIFR3=54THENPRINT@800,"File not Found":FORR=1T
O2000:NEXT:RESUME110ELSEPRINT@800,"Error in opening the file":FO
RR=1TO2000:NEXT:RESUME110
```

Print Error if not already handled.

```
2150 PRINT:PRINT"LINE NUM"ERL,"ERROR #"R3,"ERROR IS FOUND"
```

THE END OF THE PROGRAM

```
2160 END
```

TRS-80® SWAT TABLE FOR:
GRAPHIC WRITER - DISK VERSION

LINES	SWAT CODE	LENGTH
10 - 80	MH	619
90 - 120	ON	512
130 - 190	NS	508
200 - 300	GA	583
310 - 400	VP	512
410 - 490	GS	615
500 - 560	DT	562
570 - 620	HV	508
630 - 690	IC	506
700 - 790	KF	539
800 - 910	VF	522
920 - 1010	ZH	514
1020 - 1130	UD	426
1140 - 1205	RQ	506
1210 - 1310	ZZ	506
1320 - 1410	HF	523
1420 - 1500	NB	541
1505 - 1540	YS	526
1550 - 1610	GM	556
1620 - 1650	NL	510
1660 - 1740	EV	571
1750 - 1830	QZ	540
1840 - 1940	DI	569
1950 - 2040	DP	513
2050 - 2140	EC	570
2150 - 2160	BI	54

```
SS SS SS SS SS SS SS SS SS SS SS
SS                           SS
SS       TRS-80 BASIC        SS
SS     'GRAPHIC WRITER'      SS
SS     CASSETTE CHANGES      SS
SS  AUTHOR: DARWIN COLLINS   SS
SS    COPYRIGHT (C) 1982     SS
SS SOFTSIDE PUBLICATIONS, INC SS
SS                           SS
SS SS SS SS SS SS SS SS SS SS SS
```

```
DELETE 1480-1490
DELETE lines 1660-2060
DELETE lines 2110-2140
```

```
110 A$=CHR$(158):B=CHR$(173):L=0:POKE16409,1:CLS:PRINT@10,"Width
 Max.:"XM,"Height Max.:"YM:PRINT@276,"What is the Choice?"CHR$(1
0);TAB(20)"<C> haracter Write"CHR$(10);TAB(20)"<E> rase the Scre
en"
120 PRINTTAB(20)"<G> raphic Write"CHR$(10);TAB(20)"<H> ardcopy"C
HR$(10);TAB(20)"<I> mitate a Character"CHR$(10);TAB(20);"<L> oad
 Sketch"
130 PRINTTAB(20)"<M> ove a line of a sketch"CHR$(10);TAB(20)"<S>
 ave Sketch"CHR$(10);TAB(20)"<Q> uit, exit the Program"CHR$(10);
TAB(13)"** <Press the Appropriate Key> **";
140 I=295:GOSUB160:FORR=1TO10:IFMID$(" CMHGSLQEI",R,1)<>C$THENNE
XT:GOTO140
150 ONRGOTO290,290,480,610,790,1390,1440,1500,280,1505
200 FORR2=0TO14:PRINT@R2*64,MID$(S(R2+YT),XL);:NEXT:RETURN
```

TRS-80®

```
270 IFL=0THENRETURNELSER2=-((YM-YT)=>14)*14-((YM-YT)<14)*(YM-YT)
:FORR3=0TOR2:S(R3+YT)=LEFT$(S(R3+YT),XL-1)+I$(R3)+MID$(S(R3+YT),
XL+64):NEXT:L=0:RETURN
350 GOSUB270:IFE=0THENOND-23GOSUB420,440:PRINT@Y*64,MID$(S(P),XL
,64);:GOTO310ELSEOND-23GOSUB430,450:GOSUB200:GOSUB310
1390 CLS:PRINT@15,"* Section to Save Material to Tape *":PRINTCH
R$(10);CHR$(10);CHR$(10)"Name of Material to be Saved (STOP to A
bort)";:INPUTR$
1392 IFLEN(R$)>25THENPRINT"Illegal number of characters":FORR=1T
O2000:NEXT:GOTO1390:ELSEIFR$="STOP"THEN110
1400 PRINTCHR$(10);CHR$(10)"Press Record and play, press <SPACEB
AR> when ready":GOSUB180:IFC<>" "THEN110ELSEPRINTCHR$(10);CHR$(1
0)"Deleting all Commas and Colons from Material"
1410 FORR=1TOYM:FORR1=1TOXM:IFMID$(S(R),R1,1)=","ORMID$(S(R),R1,
1)=":"THENS(R)=LEFT$(S(R),R1-1)+" "+RIGHT$(S(R),XM-R1)
1420 NEXT:NEXT:PRINTCHR$(10);CHR$(10);CHR$(10)"Now saving it on
Tape"
1430 PRINT#-1,255,R$
1435 FORR=1TOYM:PRINT#-1,R,">"+S(R):NEXT:PRINT#-1,XM,"@END@":GOT
O110
1440 CLS:PRINT@15," * Section to Load Material from Tape *":PRIN
T@461,"Set Volume, Press Play on Tape Recorder"
1443 R$="":PRINT@588,;:INPUT" Name of the Program (STOP to Abort
) ";R$:IFR$="STOP"THEN110ELSEIFLEN(R$)>25THENPRINT" ILLEGAL NUMB
ER OF CHARACTERS":GOSUB180:GOTO1400
1445 PRINTCHR$(10)"Press (SPACEBAR) when Ready";:GOSUB180:IFC<>"
 "THEN110
1450 INPUT#-1,R,R1$:IFR$<>R1$ANDR$<>""THENIFR=255THENPRINT"NOW H
AVE PASSED-"R1$;:GOTO1450:ELSE1450ELSEIFR$=""ANDR<>255THEN1450
1460 FORR1=1TO91:INPUT#-1,R,S(R1):IFS(R1)<>"@END@"THENPRINT@577,
STRING$(132,32);:PRINT@576,S(R1);:NEXT
1470 XM=R:YT=1:XL=1:YM=R1-1:PRINT@920,"Initializing strings":FOR
R=1TOYM:S(R)=MID$(S(R),2)+STRING$(XM-LEN(S(R))+1,32):NEXT:GOTO11
0
2140 IFERL=1430ORERL=1440ORERL=1450ORERL=1460THENPRINT"TAPE RECO
RDER ERROR, PRESS SPACEBAR FOR MENU":GOSUB180:RESUME110
```

TRS-80® SWAT TABLE FOR: GRAPHIC WRITER - TAPE VERSION

LINES	SWAT CODE	LENGTH
10 - 80	MH	619
90 - 130	LA	601
140 - 250	ZK	481
260 - 320	UG	528
330 - 440	GO	535
450 - 500	GL	519
510 - 570	FG	590
580 - 640	OQ	560
650 - 710	MF	554
720 - 830	PN	535
840 - 950	DL	526
960 - 1070	YU	564
1080 - 1170	EG	547
1180 - 1280	NZ	489
1290 - 1390	NE	544
1392 - 1440	QO	537
1443 - 1500	EO	526
1505 - 1540	YS	526
1550 - 1610	GM	556
1620 - 1650	NL	510
2070 - 2160	FD	368

A feast of computing ideas.

If you work with a 6502/6809-based system, you're probably hungry for the facts and ideas that will help you understand the inner workings of your computer. You want to go beyond canned software—use your computer for more than games—learn the advanced programming techniques that enable you to get the most out of your 6502/6809 system.

MICRO, The 6502/6809 Journal, gives you page after page, month after month, of solid information to sink your teeth into. **MICRO** is the premier how-to magazine for serious users of the Apple, PET/CBM, OSI, Atari, AIM, SYM, KIM, and all 6809 based systems including the TRS-80 Color Computer. It's a resource journal internationally respected by professionals in business, industry, and education. Every issue of **MICRO** keeps you informed with up-to-the-minute data on new products and publications:

- **hardware catalog** with organized, concise description
- **software catalog** in an easy-to-use format
- **new publications** listed and annotated
- **reviews and evaluations** of significant products

And there's much more:

- **In-depth hardware tutorials** bring expert advice into your home or office.
- **Detailed discussions of programming languages** deepen and broaden your programming ability.
- **Complete program listings** enable you to increase your machine's capabilities.
- **Bibliography of 6502/6809 information** helps you to find pertinent articles in a timely manner.
- **Special monthly features** with in-depth treatment of one subject or system increase your knowledge of the field.
- **Balanced mix of machine-specific and general articles** for your everyday use as well as long-range reference needs.
- **Informative advertising** focused specifically on 6502/6809 machines keeps you abreast of latest developments.
- **Reader feedback** puts you in touch with other micro-computerists.

MICRO is the magazine you need to get the most from your own 6502/6809 system!

You'll love every byte.

To order, send your check or international money order (payable to MICRO) and the order form at left, to:

Subscription Fulfillment
MICRO, Dept. MI
34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824

Or, for your convenience, call our toll-free number:

1-800-345-8112

(In California, 800-772-3545, Ext. 564)

and charge your subscription to your MasterCard or VISA. (All orders must be prepaid in U.S. dollars or charged to your MasterCard or VISA.)

YES! I want to get more from my microcomputer. Please send me

__ year(s) of MICRO at $_____/year. (Outside U.S. and Canada, please indicate via ☐ surface or ☐ air mail.)

Name

Company

Street

City State Zip Code

☐ Check enclosed for $ _____
☐ Charge my credit card account
☐ VISA ☐ MasterCard

Signature

Card number

Expiration date

SUBSCRIPTION RATES (U.S. dollars) Yearly subscription (ISSN 027-9002) saves 20% off the single-issue price.

U.S. $24*

Canada $27

Europe $27 ($42 by air mail)

Mexico, Central America, Mideast, North and Central Africa $27 ($48 air)

South America, Far East, South Africa, Australasia $27 ($72 air)

* SPECIAL OFFER—U.S. ONLY: Save even more—30% off single-issue price: 2 years, $42

Dept. S S

EdCom™'82

OCTOBER 21, 22, 23 and 24, 1982
LOS ANGELES CONVENTION CENTER

DEMOS SEMINARS WORKSHOPS HANDS-ON

Computer Conference and Expo for Education

SOFTWARE • HARDWARE • FUNDING

DATA MANAGEMENT • BID SPECIFICATION • NETWORKING

HARDWARE AQUISITION • INSTRUCTIONAL MANAGEMENT

WORD PROCESSING • HIGHER EDUCATION APPLICATION • MATH AND SCIENCE

MICRO-MINI CONNECTION • TELECOMMUNICATION • COMPUTER ASSISTED INSTRUCTION

SIMULATION • PROGRAM DESIGN • DISTRIBUTED DATA PROCESSING • PURCHASE AND POLICY

MicroCourses* and Hands-On Sessions

ATARI **IBM** **PET** **TRS-80**

Authoring • Programming • Word Processing • LOGO • Management • Graphics

KEYNOTE ADDRESS

Governor Edmund G. Brown Jr.

Friday Luncheon: Andrew Molnar

Saturday Luncheon: Sylvia Charp

PRE-REGISTRATION: Please complete the form below as you wish it to appear on your name tag. Check the appropriate "Registration Fee" box below. After October 1, 1982, On-Site registration only.

(One day or full conference fees include admission to the exhibits)

	PRE-REG	ON-SITE REG
☐ CF — Full Conference	$125	$140
☐ LA — Full Conference + 1 Luncheon	$140	$155
☐ LB — Full Conference + 2 Luncheons	$155	$170
☐ CO — One Day Conference	$45	$55
☐ LC — One Day Conference + 1 Luncheon	$60	$70
☐ EO — Exhibits Only (Four Days)	$10	$10
☐ LE — Exhibits Only + 1 Luncheon	$25	$25
☐ LF — Exhibits Only + 2 Luncheons	$40	$40

☐ Information on Seminars and MicroCourses. *Additional Registration Fee For MicroCourses.

NAME	
POSITION	
AFFILIATION	
ADDRESS	
CITY	STATE ZIP
PHONE	AREA CODE — EXT

SP2

My registration check for the amount of $_____ is enclosed.
Make checks payable to Judco Computer Expos, Inc.
Mail to: Judco Computer Expos, Inc.
P.O. Box 963
Scottsdale, AZ 85252

For further information call: 800-528-2355 Toll Free (Outside Arizona)
602-990-1715 (Arizona Only)

UNIVERSITY CREDIT AVAILABLE

In the event cancellation is necessary, your written request for refund will be honored if received by October 1, 1982. A $15 handling fee will be charged.

TRS-80® Graphics For The Model I and Model III

Reviewed by Margaret Grothman

by David A. Kater and Susan J. Thomas. Publisher: Byte Books, Inc., 70 Main Street, Peterborough, NH 03458. Suggested retail price: $12.95.

Are you a beginning BASIC programmer looking for a good introductory treatment of TRS-80® graphics or an experienced programmer interested in acquiring a better command of graphics techniques? Perhaps you know all about graphics techniques, but want a book to help you generate ideas for your own special applications? Or, maybe you like to buy just one good book on a subject, at a reasonable price, which covers all areas adequately. If you are any of the above, *TRS-80® Graphics* may be for you. No book can be all things to all people, but this one succeeds in being several things to many people.

The book is divided into two major parts, the first called "Basic Tools," and the second "Applications." The material ranges from fundamental to sophisticated graphics techniques, each presented simply and clearly.

Only a little knowledge of BASIC is assumed in the first few chapters. They consist of a tutorial on TRS-80® graphics with some review of elementary BASIC language programming. The plot and print coordinate systems are explained well, emphasizing the relationship between them, a confusing matter for many beginners. These chapters are also a good review and a good base for experimentation for more experienced programmers. Although I was familiar with all of the fundamental techniques treated in these chapters, many of the sample programs gave me the idea for new uses of those techniques in my own programs.

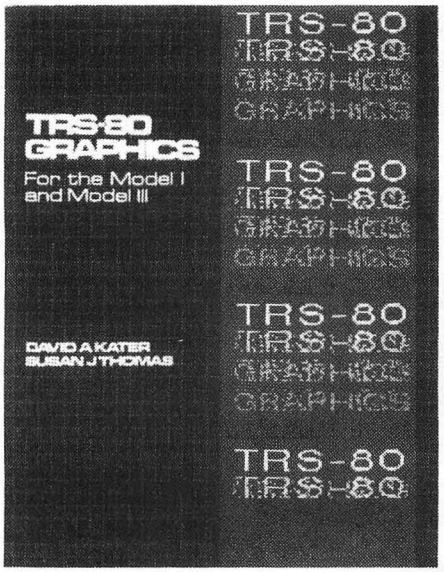
The chapters in Part II, "Applications," can be read independently of each other. Even if you are interested in one or a few them, all contain good ideas and techniques, useful in other applications.

Some of the book's strong points are listed below.

1) The sample programs are concise and efficient. Until the later chapters, few programs exceed 18 lines in length. Each one is developed in stages so the reader can learn exactly how it works.

2)The text material is minimal — you learn by illustration and example in this book.

3)A wide range of users is addressed. The book is an introduction to graphics for beginning programmers, as well as a sophisticated graphics techniques reference for more advanced programmers.



4)Model I and Model II users, with disk or non-disk systems, are included. Almost everything may be used by non-disk users, unlike some books in which large sections are useless to them. Model I and Model III differences are clearly pointed out.

5)The book is well organized. Early chapters, which are devoted to beginners, do not attempt too much. Finer points are covered in Part II. The index and table of contents are good, allowing the book to be of continuing use for reference.

6)Several utility programs are included which will be of lasting value to many programmers.

7)The various graphics techniques are compared, so the reader can learn which is most appropriate for his or her needs.

One of the strongest features of the book is the way in which the sample programs are developed. The programs are short and are built up in small segments, allowing you to learn exactly how they work. An example from Chapter 13 is shown below. The program lines are presented to make a single point move randomly around the screen.

```
10 X=64 : Y=24 : CLS
20 H=RND(3)-2 : K=RND(3)-2
100 RESET(X,Y) : X=X+H : Y=Y+K : SE
T(X,Y) : GOTO 20
```

The second step expands the program to allow the user to move the point with the arrow keys.

```
20 REM
30 I$=INKEY$ : IF I$="" THEN 80
40 IF ASC(I$)=8 H=-1: K=0: GOTO 80
50 IF ASC(I$)=9 H=1: K=0: GOTO 80
60 IF ASC(I$)=10 H=0: K=1: GOTO 80
70 IF I$=" " H=0: K=-1
```

The third set of changes incorporates a subroutine to draw a border around the screen. Line 100 is changed so that the point is not erased as it moves, but leaves a trail.

```
10 X=64 : Y=24 : CLS : GOSUB 110
100 X=X+H : Y=Y+K : SET(X,Y) : GOTO
20
110 FOR J=0 TO 127: SET(J,3) : SET(
J,47)
120 IF J<48 SET(0,J) : SET(127,J)
130 NEXT J : RETURN
```

If you experiment with the program at this stage, you discover an error message when the point travels too far in any direction. The program is next developed into a game — a player loses when the point bumps into a boundary or against its own trail. The POINT function accomplishes this in line 90.

```
80 IF H=0 AND K=0 SET(X,Y): GOTO 30
90 IF POINT(X+H,Y+K) PRINT@ 980, "Y
OU LOSE 140 GOTO 140
```

And a timer is added so the player can tell how long he or she survived.

TRS-80®

```
10 X=64:Y=24:CLS:GOSUB110:T=0
20 T=T+1:PRINT@28,"TIME:"INT(T/10);
90 IFPOINT(X+H,Y+K)PRINT@980,"YOU LOSE! TIME:"INT(T/10);:
GOTO140
```

A final, 14 line listing of the game is provided for the programmer to proofread. For those who want to take the exercise further, suggestions are given for improving the game. This method of presentation not only has educational merit, but reduces the tedium of typing, since only a few lines are entered at a time.

There are many programs in the book that I especially enjoyed and intend to use regularly: The graphics editor subroutine for poking characters into strings (Chapter 5); the hexentry program for packing a Machine Language subroutine into a string (Chapter 6); the cartesian coordinate program for graphing functions, (Chapter 7); the histogram program, and the correlation/regression program, both of which automatically produce scales to accomodate the data entered (Chapter 8).

If you are interested in figure animation, a la Leo Christopherson, you will be especially interested in Chapter 13. A character named Critter is made up of line by line graphics strings. Animation is achieved by alternating a line string with other strings, each a variation of the first.

The Ultrasketch program in Chapter 15 is a utility program that allows you to produce screen pictures with both print and graphics characters. When you are satisfied with your creation, it can be packed into strings for use in your own programs. Because Ultrasketch packs the screen into an array with a separate string for each line, it could be used successfully to develop characters for animation. It may lack the frills that commercial programs of its kind offer, but it works perfectly, and may even be better than the commercial programs for some users.

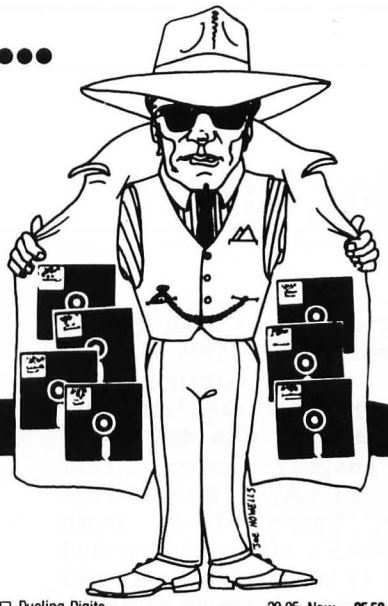
Apart from the Ultrasketch program, Chapter 15 is disappointing. The authors make an attempt to describe the types of graphics products available, both software and hardware. The result is unsatisfying and thin — you can learn more about what is available by paging through any computer magazine. Although the preface to Chapter 15 warned me not to be disappointed if my favorite was missing, I was still disappointed that the examples of commercial software given did not include *GEAP*, by Bill Mason (distributed by J.F. Consulting, Buttonwood, CA). Chapter 15 also includes a short description of printer capabilities in the section on graphics hardware. This is the only mention of printers in the book. The omission seems wise, in view of the lack of printer uniformity, but is something you should be aware of if you have a special interest in printer graphics.

The programs presented in the book are available on disk or cassette, but having them in this format may reduce the learning effect. Few readers owning the disk would have the perseverance to follow the line by line explanations, without the reinforcement provided by typing and running the programs as they develop.

There are a few typographical errors of the obvious, and harmless, kind. The Machine Language routine to reverse graphics in Chapter 6 does not work for me — I haven't yet discovered why. Lack of typing or proofreading skill may be to blame, rather than the book.

In summary, *TRS-80® Graphics for the Model I and Model III* is an excellent book — both useful and fun. Because it contains so much material, you can expect to spend many hours with it.

WE'RE EXPOSING OURSELVES...
With these shamelessly low prices on apple® Software!



GREAT SERVICE
GREAT PRICES
WHAT ELSE YOU WANT?

micro mountain

OVER 1,000 ITEMS IN STOCK — STOCK ITEMS SHIPPED SAME DAY — ASK FOR FREE CATALOG

HARDWARE SPECIALS

☐ Smarterm 80 Column board	345.00	Now	289.50
☐ Z-Card (Z-80, CP/M)	279.00	Now	239.50
☐ ALF 9-Voice Card w/Software	199.50	Now	179.50
☐ Low Res Color Monitor	449.00	Now	389.50
☐ RGB Color II Interface	199.00	Now	169.50
☐ Dana Cooling Fan	49.95	Now	42.50
☐ Elephant Diskettes (Box of 10)	39.95	Now	24.50
☐ Hand Controller (Paddles)	29.95	Now	28.50
☐ Numeric Keypad	149.50	Now	139.50
☐ Z-80 Softcard with CP/M	399.95	Now	299.50
☐ CPS Multifunction Card	239.95	Now	189.50
☐ Music System	395.00	Now	339.50
☐ NEC 12" Hi-Res Green Monitor	285.00	Now	189.50
☐ The Grappler (Specify printer)	149.95	Now	139.50
☐ 16k Ramcard	195.00	Now	119.50
☐ Game Paddles	39.95	Now	32.50
☐ Joystick	59.95	Now	49.50
☐ Expand-a-Port	59.95	Now	49.50
☐ Thunderclock Plus	139.00	Now	129.50
☐ Station II	129.95	Now	119.50

UTILITY PROGRAM SPECIALS

☐ Hi-Res Secrets	124.95	Now	99.50
☐ Dos Boss	24.00	Now	21.50
☐ Utility City	29.50	Now	25.50
☐ Alpha Plot	39.50	Now	34.50
☐ Tip Disk #1	20.00	Now	17.50
☐ 3-D Graphics System	39.95	Now	34.50
☐ Home Money Minder	34.95	Now	29.50
☐ Home Accountant	74.95	Now	64.50
☐ TASC Compiler	175.00	Now	152.50
☐ The Voice	39.95	Now	34.50
☐ Master Diagnostics	50.00	Now	45.50
☐ The Locksmith	99.95	Now	86.50
☐ Lisa Assem. Lang. Dev. Sys.	79.95	Now	69.50
☐ Expediter II	99.95	Now	87.50
☐ The Manipulator	34.95	Now	29.50

DYNOMITE DEAL #1

Kensington Microware

SYSTEM SAVER

REGULAR $89.95

NOW $69.50

☐ Graphics Magician	59.95	Now	52.50
☐ Special Effects	39.95	Now	33.50
☐ Super Disc Copy	30.00	Now	25.50
☐ Disk Recovery (The Scanner)	30.00	Now	25.50
☐ Dos Plus	25.00	Now	21.50
☐ Back It Up (Nibble Copier)	59.95	Now	52.50
☐ Original Quick Loader	24.95	Now	21.50
☐ E-Z Draw 3.3	49.95	Now	39.50
☐ Data Capture 4.0/80 (Videx)	89.95	Now	75.50
☐ ASCII Express	64.95	Now	61.50
☐ Z-Term (Req CP/M	99.95	Now	92.50
☐ Z-Term Professional (Req CP/M)	149.95	Now	125.50
☐ Speed Star	134.95	Now	119.50
☐ A2-3D Graphics Package	59.95	Now	52.50
☐ Program Line Editor	40.00	Now	32.50
☐ Higher Graphics II	35.50	Now	27.50

EDUCATIONAL PROGRAM SPECIALS

☐ Auto Atlas	47.50	Now	41.50
☐ Compu-Math Arithmetic	49.95	Now	39.50
☐ Compu-Math Fractions	39.95	Now	33.50
☐ Compu-Math Decimals	39.95	Now	33.50
☐ Algebra 1	39.95	Now	33.50
☐ Statistics 3.0	29.95	Now	25.50
☐ Spelling Bee w/Reading primer	39.95	Now	34.50
☐ Counting Bee	29.95	Now	24.50
☐ Pythagoras and the Dragon	39.95	Now	35.50
☐ Isaac Newton, Fig Newton	49.95	Now	43.50
☐ Master Type	39.95	Now	34.50
☐ Englash SAT #1	30.00	Now	26.50
☐ U.S. Constitution Tutor	30.00	Now	26.50
☐ Typing Tutor II	24.95	Now	22.50
☐ Division Skills	44.95	Now	39.50
☐ Mixed Numbers	44.95	Now	39.50
☐ Vocabulary (Preifx, Suffix, Roots)	44.95	Now	39.50
☐ Punctuation (Commas)	44.95	Now	39.50
☐ Elementary Math Edu-Disk	39.95	Now	36.50
☐ Lisa Educational Sys.	119.95	Now	99.50

DYNOMITE DEAL #2

Silicon Valley

WORD HANDLER

REGULAR $249.95

NOW $159.50

BUSINESS PROGRAM SPECIALS

☐ CPA #1 General Ledger	250.00	Now	199.50
☐ CPA #2 Accounts Receivable	250.00	Now	199.50
☐ CPA #3 Accounts Payable	250.00	Now	199.50
☐ CPA #4 Payroll	250.00	Now	199.50
☐ The Budget Planner	150.00	Now	125.50
☐ Tax Beater	129.95	Now	105.50
☐ Real Estate Analysis Program	129.95	Now	105.50
☐ Financial Partner	175.00	Now	152.50
☐ Apple Pie (All Versions)	129.95	Now	109.50
☐ Tax Preparer	150.00	Now	135.50
☐ Creative Financing	150.00	Now	135.50
☐ Datadex Data Base Manager	150.00	Now	129.50
☐ Datafactory 5.0	300.00	Now	249.50
☐ Time Manager	150.00	Now	127.50
☐ Screenwriter II	129.95	Now	109.50
☐ The General Manager	99.95	Now	86.50
☐ Dictionary	99.95	Now	86.50
☐ Executive Speller	75.00	Now	69.50
☐ Magic Window Word Proc.	99.95	Now	84.50
☐ Magic Mailer	69.95	Now	57.50
☐ Magic Words	69.95	Now	57.50
☐ PFS: Personal Report Sys.	95.00	Now	79.50
☐ IFO Database Manager	120.00	Now	105.50
☐ Inventory System 3.3	200.00	Now	174.50
☐ D-B Master	229.95	Now	189.50
☐ D-B Utility Pack	99.95	Now	89.50
☐ D-B Utility Pack #2 (May)	99.95	Now	89.50
☐ Visacalc 3.3	250.00	Now	209.50
☐ Visifile	250.00	Now	209.50

GAMES SPECIALS

☐ Eliminator	29.95	Now	25.50
☐ Temple of Apshai	39.95	Now	32.50
☐ Crush, Crumble and Chomp	29.95	Now	24.50
☐ Ricochet	19.95	Now	17.50
☐ Zero Gravity Pinball	29.95	Now	25.50
☐ Star Blazer	31.95	Now	26.50
☐ Labrinth	29.95	Now	25.50
☐ Dueling Digits	29.95	Now	25.50
☐ Bug Attack	29.95	Now	25.50
☐ Apple Barrel II (20 Programs)	34.95	Now	29.50
☐ Snack Attack	29.95	Now	24.50
☐ County Fair	29.95	Now	25.50
☐ Casino	39.95	Now	34.50
☐ The Prisoner	29.95	Now	25.50
☐ Rendezvous (April)	39.95	Now	32.50
☐ Alien Ambush	24.95	Now	19.50
☐ Horizon V	34.95	Now	28.50
☐ Sargon II	34.95	Now	29.50
☐ Reversal (Othello)	34.95	Now	29.50
☐ Zork II	39.95	Now	32.50
☐ Robot Wars	39.95	Now	34.50
☐ A.B.M.	24.95	Now	22.50
☐ Three Mile Island	39.95	Now	34.50
☐ Castle Wolfenstein	29.95	Now	25.50
☐ Wizard and the Princess	32.95	Now	29.50
☐ Hi-Res Soccer	29.95	Now	25.50
☐ Threshold	39.95	Now	35.50
☐ Time Zone	99.95	Now	86.50
☐ Marauder	34.95	Now	28.50
☐ Microchess 2.0	24.95	Now	22.50
☐ Gammon Gambler	24.95	Now	22.50
☐ Falcons	29.95	Now	25.50
☐ Star Blaster	29.95	Now	25.50
☐ Congo	32.95	Now	28.50
☐ Fastgammon	24.95	Now	21.50
☐ Beer Run	34.95	Now	29.50

DYNOMITE DEAL #3

WHILE THEY LAST!

Buy 1 Edu-Ware "Space" at $29.95
Get "Space II" (Reg. $24.95)

FREE!

☐ Kabul Spy	34.95	Now	28.50
☐ Bandits	34.95	Now	28.50
☐ Draw Poker	29.95	Now	24.50
☐ Computer Baseball	39.95	Now	33.50
☐ Tigers in the Snow	39.95	Now	34.50
☐ Computer Air Combat	59.95	Now	49.50
☐ Napoleons Campaign	59.95	Now	49.50
☐ Road to Gettysburg	59.95	Now	49.50
☐ Pursuit of th Graft Spee	59.95	Now	49.50
☐ Flight Simulator	33.50	Now	29.50
☐ Odyssey	30.00	Now	24.50
☐ Adventure to Atlantis	40.00	Now	34.50
☐ Space Raiders	29.95	Now	25.50

FREE OFFER

—Your Choice—
Original
Adventure Game
or **Applesoft™ Tutorial**
on disc with purchase of Any 3 Programs!

DEDUCT 3% if payment accompanies order. WE PAY SHIPPING on all software orders OVER $50 in Continental U.S. (Foreign & Air Extra). ADD $2.50 SHIPPING & HANDLING on orders under $50. CALIF. RESIDENTS ADD 6% SALES TAX. We accept MASTER CARD and VISA. C.O.D.'S ADD $5.00.

NAME _____

STREET _____

CITY _____ STATE & ZIP _____

CARD # _____ EXP. DATE _____

SIGNATURE _____

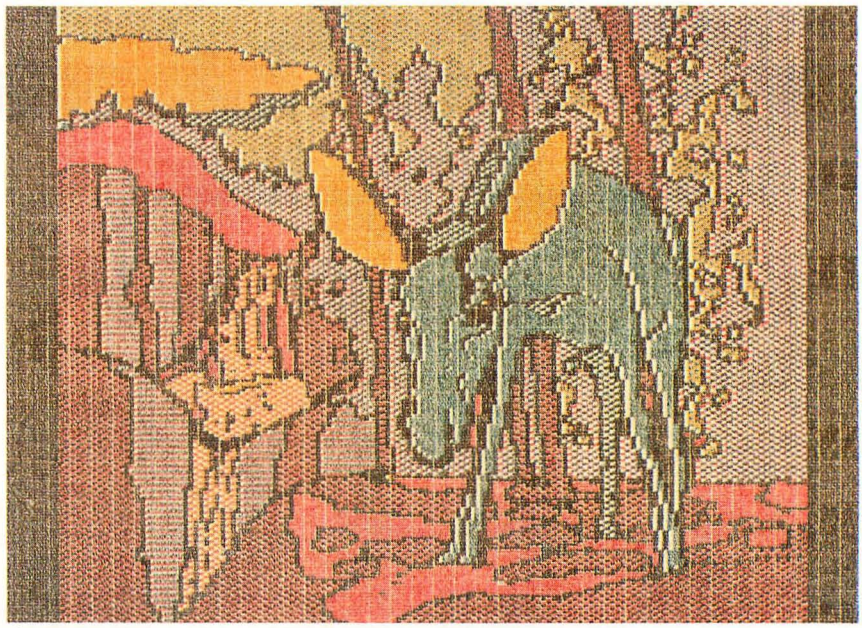
★ **518 E. ECHO CT., SAN BERNARDINO, CA 92404**
Outside Calif. Call (800) 854-5649
Calif. Residents Call (714) 886-0761

Prices Subject to Change without Notice

Color Print

Reviewed by Rick Nichols

by Courtney Goodwin, distributed by Datasoft, Inc., 19519 Business Center Drive, Northridge, CA 91324. System requirements: ATARI® 400/800 with 40K RAM, BASIC Language Cartridge, Disk Drive, ATARI® 850 Interface Module, Epson MX-80 (with GRAFTRAX), a color monitor, and 8 1/2 X 11 inch fanfold paper. All other materials are supplied with the diskette. Suggested retail price: $34.95.



A sample printout of one of the pictures supplied with *Color Print*.

When I recently purchased the Epson MX-80 printer with Graftrax-80 ROM, I was anxious to experiment with its abilities. Therefore, I purchased the *Color Print* program, which will print high resolution graphics mode 7 or 8 pictures in black and white or full color. *Color Print* is directly compatible with *Micro Painter* and *Graphic Master*, both distributed by Datasoft, and can also be used with other graphics programs, such as ATARI®'s *Graph-It.*

The diskette is autoboot and is copy protected. Upon loading the program, the first menu appears. You are given a choice of print directions, vertical or horizontal. After you've selected a print direction, the corresponding program is loaded from diskette and another menu is displayed. You then enter the first letter of the option desired. They are as follows:

(S)ave — This feature is used to save the contents of the graphics screen to diskette.

(L)oad — This will load a previously saved picture from diskette.

(P)rint — Self-explanatory.

(D)irectory — Lets you step through the directory for specific files or check on remaining sectors.

(Q)uit — Lets you exit the program.

Now the fun part — printing pictures. My first printouts were done in black and white because, quite truthfully, I thought printing in color would present too much of a challenge initially. There are four graphic pictures provided on the disk. A butterfly, rosebud (mode 7), peacock, and squares (mode 8). A mode 7 display will use 31 disk sectors of storage, while mode 8 will use 62 sectors per display. Printing black and white pictures is a snap — just follow the menu prompts. It's hard to believe a dot-matrix printer can print high resolution pictures. What will they think of next? Well, Epson has just announced *Graphtrax-Plus ROMs* and *Color Print* is compatible with them, as well. I don't know what more they could have to offer, short of allowing printing graphics from two to three times faster.

Color prints involve a somewhat more difficult procedure, (My description may even make it sound somewhat tedious.) but after forcing myself to try it, I found it wasn't so bad. After you select a print option, a series of menus will offer a choice of two print sizes, two print densities (horizontal only), and either color or black and white printouts. In the horizontal mode, a large size picture is about 6 1/2 X 9 inches, with a medium size of 3 1/4 X 4 1/4 inches. When printing vertically, the large size is 5 1/4 X 3 3/6 inches and medium is approximately 2 1/2 X 3 5/8 inches. You then select the print density, Normal (60 dots/inch) or Emphasized (120 dots/inch). The printer will make two or three passes to complete a line in the Emphasized mode, taking considerably longer to print a picture. When you've made the selection, you have the chance to INVERT the colors on the display. Inverting may be called for in a black and white print because the white areas on the screen are printed as black areas on the paper. The next menu asks for (1) Full Screen Print or, (2) Define Print Window. Full Screen is self-explanatory. With "Define Print Window," you can define which portion of the picture you wish printed. If you select option 2, you are prompted for left margin width.

To make the color print, you're required to make a "carbon sandwich." This is accomplished by tearing off a two page segment of fanfold paper. With two pieces folded at the perforation, insert a sheet of the supplied colored carbon between them. The perforated end should be

considered the top. In order for the printer to make colored pictures, multiple passes are required (in this case, 4). But, alas! How does the printer accurately print in the proper areas each time? This is done by placing a piece of tape on the tractor feed gates and drawing a horizontal line as a reference for the top starting point. When you close the gates, you have a point (the horizontal line) with which to align the top of the page. Sound hard? It really isn't.

Color printing will switch to the four default colors:

Mode 8 — #3-White, #2-Green or Blue, #1-Purple or Red, and #0-Black (or background).

Mode 7 — #3-Blue, #2-Yellow, #1-Orange, and #0-Black (or background).

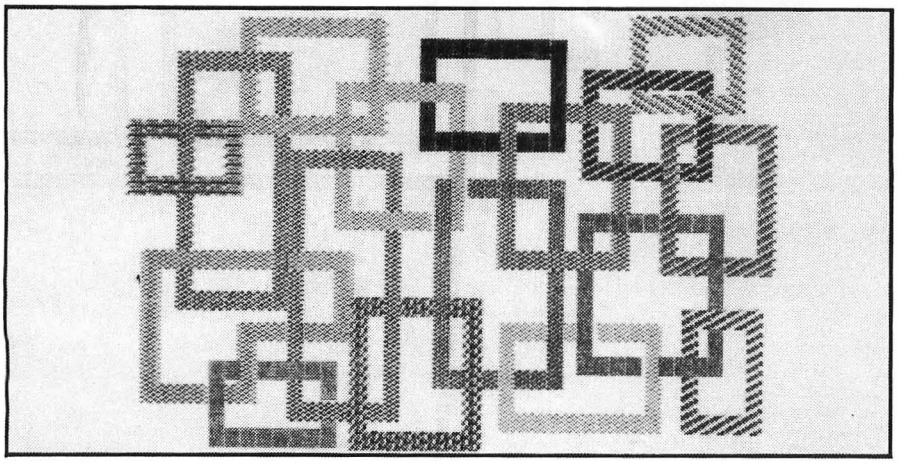
You can press the Select key and step through the colors to change the order of printing or bypass a color. The text window, at the bottom of the screen, will indicate which color number will be printed next. It is not necessary to use the default colors for the carbon inserts. You can experiment using different colored carbon for different screen colors.

I do have one, relatively minor, criticism of *Color Print.* What happens when you've used all of the colored carbons included in the package? (There are five sheets each of red, blue, green, yellow, orange and black.) There is one sentence in the instruction manual — "Colored carbon replacements are available from Datasoft." It would have been nice if they had enclosed an order form with the diskette stating prices and available colors, or suggested an alternative source. By the way, I did try a couple of stationery stores; no colored carbons. So, Datasoft may be your best bet.

I said at the outset that this program can be used with "other" graphics programs. To paraphrase the instruction manual, "if you have the necessary programming skills," you can probably find a way to make it work. Datasoft, not all of us are that skilled, yet, but we're trying. It's good programming practice, however.

If the graphics program uses a text window at the bottom of the screen, there is *hope*! I have successfully used the following procedure with the 3-D package on ATARI®'s *Graph-It.* However, it did not work with the PIE or BAR GRAPH portions (program modification needed). By pressing the Break key, I was able to freeze the screen at that point. I then typed in RUN "D:HARDCOPY", (This program is provided on the diskette and is programmed to preserve the graphic screen.) the program prompts appeared and away it went! I might add that the ATARI®3-D program can provide some spectacular printouts.

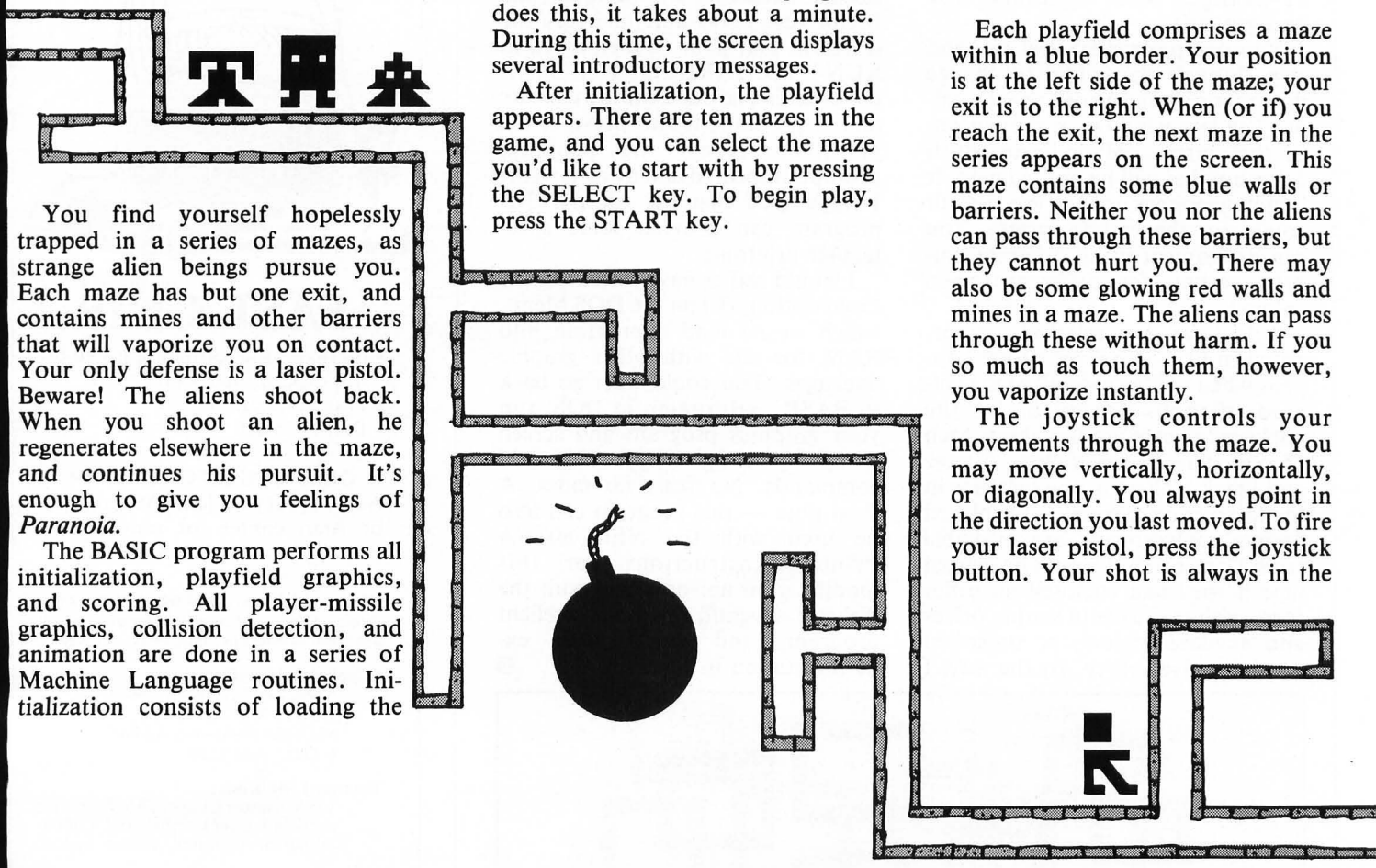
I would rather have seen a Binary Load option, (L) on the DOS Menu, which would load a program into RAM for use with other graphic packages. You could then go back to BASIC, (B) option in DOS, run your graphics program and screen dump by using simple keyboard commands. No fuss, no muss. A final note — this program can also be used with the NEC 8023-A Printer. Instructions for this modification are provided with the diskette. Overall, this is an excellent program, and much more experimentation is possible.



PARA

by Michael A. Moody

***Paranoia* is an arcade style game for an ATARI® with 32K RAM, ATARI® BASIC cartridge, disk drive, and a joystick. It is included as the bonus program on this month's ATARI® DV.**



You find yourself hopelessly trapped in a series of mazes, as strange alien beings pursue you. Each maze has but one exit, and contains mines and other barriers that will vaporize you on contact. Your only defense is a laser pistol. Beware! The aliens shoot back. When you shoot an alien, he regenerates elsewhere in the maze, and continues his pursuit. It's enough to give you feelings of *Paranoia.*

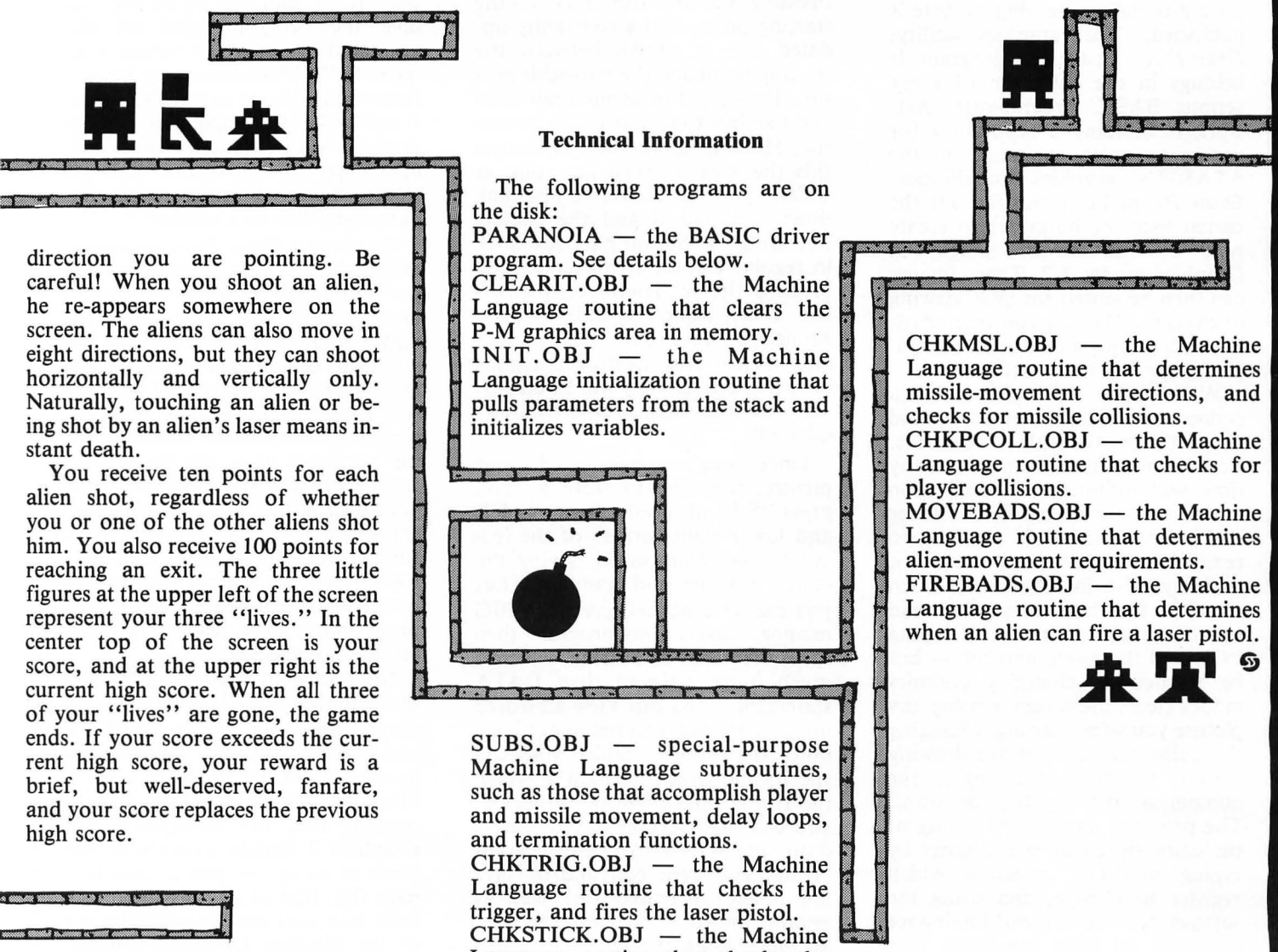
The BASIC program performs all initialization, playfield graphics, and scoring. All player-missile graphics, collision detection, and animation are done in a series of Machine Language routines. Initialization consists of loading the Machine Language routines from disk, redefining the character set, and miscellaneous "housekeeping" functions. Since a BASIC program does this, it takes about a minute. During this time, the screen displays several introductory messages.

After initialization, the playfield appears. There are ten mazes in the game, and you can select the maze you'd like to start with by pressing the SELECT key. To begin play, press the START key.

Each playfield comprises a maze within a blue border. Your position is at the left side of the maze; your exit is to the right. When (or if) you reach the exit, the next maze in the series appears on the screen. This maze contains some blue walls or barriers. Neither you nor the aliens can pass through these barriers, but they cannot hurt you. There may also be some glowing red walls and mines in a maze. The aliens can pass through these without harm. If you so much as touch them, however, you vaporize instantly.

The joystick controls your movements through the maze. You may move vertically, horizontally, or diagonally. You always point in the direction you last moved. To fire your laser pistol, press the joystick button. Your shot is always in the

NOIA



direction you are pointing. Be careful! When you shoot an alien, he re-appears somewhere on the screen. The aliens can also move in eight directions, but they can shoot horizontally and vertically only. Naturally, touching an alien or being shot by an alien's laser means instant death.

You receive ten points for each alien shot, regardless of whether you or one of the other aliens shot him. You also receive 100 points for reaching an exit. The three little figures at the upper left of the screen represent your three "lives." In the center top of the screen is your score, and at the upper right is the current high score. When all three of your "lives" are gone, the game ends. If your score exceeds the current high score, your reward is a brief, but well-deserved, fanfare, and your score replaces the previous high score.

Technical Information

The following programs are on the disk:

PARANOIA — the BASIC driver program. See details below.

CLEARIT.OBJ — the Machine Language routine that clears the P-M graphics area in memory.

INIT.OBJ — the Machine Language initialization routine that pulls parameters from the stack and initializes variables.

SUBS.OBJ — special-purpose Machine Language subroutines, such as those that accomplish player and missile movement, delay loops, and termination functions.

CHKTRIG.OBJ — the Machine Language routine that checks the trigger, and fires the laser pistol.

CHKSTICK.OBJ — the Machine Language routine that checks the joystick and passes player-movement directions to the player-movement subroutine.

CHKMSL.OBJ — the Machine Language routine that determines missile-movement directions, and checks for missile collisions.

CHKPCOLL.OBJ — the Machine Language routine that checks for player collisions.

MOVEBADS.OBJ — the Machine Language routine that determines alien-movement requirements.

FIREBADS.OBJ — the Machine Language routine that determines when an alien can fire a laser pistol.

Draw Pic

Reviewed by David Plotkin

by Dennis Zander (Artworx Software Company, 150 North Main Street, Fairport, NY 14450). System requirements: ATARI® 400/800, ATARI® BASIC Cartridge, 16K Cassette, 24K Disk. Suggested Retail Price: $29.95 Cassette; $33.95 Disk.

Every so often, a piece of software especially deserving of note is published. The graphics utility, *Draw Pic*, is just such a program. It belongs in the collection of every serious BASIC programmer. Artworx is building a reputation for programs which capitalize on the ATARI®'s graphics capabilities. *Draw Pic* makes it possible for the casual user (or hobbyist) to create part or full screen images in Graphics modes 3-7. These images can then be stored for later viewing or changes. They are constructed using only the joystick and a few simple, single-key commands.

When *Draw Pic* is run, a menu comes up on the screen to remind you of the available commands. You can return to this menu at any time, without hurting the picture on the screen, by typing in the letter "M". The commands are easy to remember, however, and I did not find myself using the menu much after the first hour or so. You can select a graphics mode by typing "G" and the mode number — but be warned — changing graphics modes clears the screen, erasing any picture you were drawing. Changing the color register you are drawing with is a matter of typing in the number of the register you want. The program also allows setting of the color stored in each register by typing in "C", selecting which register to change, and using the joystick to vary hue and luminance until you get what you want. This process is made easier by the fact that the colors on the screen change to match your current choice. Also, the color selections are stored with the image.

The actual drawing commands are Plot Point (P), Draw Line (D), and Rubber Band (R). To plot a point, you move the flashing cursor to the point you want using the joystick, and press the fire button. To draw a line, place the cursor at the starting point, press fire, then move to the endpoint and press the fire button again. Rubber Band is an interesting and useful variation on Draw Line and another commonly seen graphics command, Fill. Pressing the fire button specifies the starting point, and a constantly updated line is drawn between the starting point and the moveable cursor. Pressing fire again draws the line one last time and frees the cursor. Holding down the fire button fills the area between the point at which you first started holding down the button and the cursor. Releasing the button puts you back in regular Rubber Band mode, and pressing fire yet again frees the cursor. When using the Rubber Band Mode, you have to be careful not to draw over previously drawn sections of the screen, as they will be erased. To clear the screen, type (SHIFT) CLEAR.

Once you've constructed your picture, it's time to store it. You press "S" and specify the upper left and lower right corners of the rectangle you want saved, giving the image a name and number. Thus you can save partial screens, a BIG memory saver. The program then saves the image as a string, which is much more efficient than DATA statements. You can View all stored images, or load a stored image back onto the screen. Since you can specify the screen location at which the stored image will be displayed, you can load several stored images onto one screen to create a composite, and even edit and/or save the image generated this way as another image. You can even load an image onto the screen in a different graphics mode than it was drawn. The only requirement is that the screen mode and the original image graphics mode be compatible. (Modes 3, 5, and 7 are compatible; 4 and 6 are compatible.) To delete a stored image, you just type (SHIFT) DELETE, and the program will delete all images from the one specified to the end. This can be a problem, as you might want to delete only image #2, but not any others. What you can do then is to load your last image onto the screen, then store it again as image #2. The program automatically erases the old image #2, and puts your new one in its place. Delete the last image, and you're all set. To save the program, and all the generated pictures for future use, type "Q". Make sure you have a formatted diskette in the disk drive. If you've left the original disk in the drive, it will be written over by the new copy of *Draw Pic* with its stored images. I prefer to use the purchased disk as a master.

As part of *Draw Pic*'s documentation, Mr. Zander provides a sample routine showing how to use images in your own program. It involves some initialization steps, the string data statements containing the image, a Machine Language routine stored in an ADDR (M.L. routine) format, and a USR call. All the necessary lines can be LISTed from *Draw Pic* and ENTERed into your own program, so no keyboard entries are necessary — it's all done for you by the program. You may want to renumber the subroutines to put them at the beginning of your program (they run faster that way), but it's not necessary.

Not only does *Draw Pic* eliminate the drudgery of creating images with graph paper and a mass of PLOT and DRAWTOs, but, since the images are sent to the screen by a Machine Language routine, it is incredibly fast. For example, a full Graphics 7 Screen appears in the blink of an eye — just as fast as a page flip. Best of all, this speed gets built into your own program by use of the Machine Language routine. You can animate your sequences in full color (not just one-color shapes like *Player-Missile* graphics), even if you don't understand how to adjust memory to "flip pages". Of course, each image, when stored as string data, takes up just as much memory as if it were actually drawn (For example, almost 4K in Graphics 7, for

a full screen). Since partial screens can be saved, however, you can do significant animation with small portions of the screen without using up too much memory.

A few relatively minor complaints: In switching from Draw Line to Plot Point mode, when you press the "P", a line is drawn from the cursor back to the end of the last line. This means you must switch modes *before* you move the cursor. Also, the program will error out and stop if you forget to enter a number when asked what image at which to begin VIEWing. I'm told by Mr. Zander that this is being fixed, however, so don't expect to see it. It would also be nice to sacrifice the text update on the bottom of the screen on occasion, so that a full screen (GRAPHICS N+16) image could be generated.

In summary, the marks of a good utility are:

1) That it be necessary — it shouldn't duplicate something that is easy to do from a language.

2) That it be easy to use — a complex set of instructions defeats the purpose, and sufficient error traps should be provided so that the beginner doesn't keep stopping the program.

3) That it do a complete job — once you've generated something of use, you should be able to move it over into your own program with a minimum of effort.

With *Draw Pic*, Dennis Zander has done an admirable job of constructing a great utility, well worth its price and a favor to BASIC programmers. I highly recommend it.

The display used in Artworx' game, *Beta Fighter,* was created using *Draw Pic.*

SpiralGraphics

by Sheldon Leemon

Program by Tom Giese

***SpiralGraphics* is written in ATARI® BASIC, and requires 16K RAM (Cassette) and 24K RAM (Disk). Disk drive, program cassette and MX-80 printer are optional.**

When fellow MACE (Michigan ATARI® Computer Enthusiasts) member, Tom Giese, showed me this program, I had to ask him to let me document it to share with other ATARI® owners. It may be short, but is jammed with programming goodies. Besides the main drawing routine, which demonstrates an excellent way to speed up function plotting in ATARI® BASIC, it contains a screen dump which allows you to print the graphics on an MX-80 printer without Graftrax chips, by using the block graphics mode. I added a command menu and a short routine to save the screen display quickly to disk or cassette, and restore it even more quickly. Even if you are not interested in the technical details of the program, you are bound to appreciate the almost endless variety of beautiful spiral graphics that it will put on your ATARI® screen.

Instructions

When the program comes up, it will draw a design, and then prompt for a choice from the following commands: Auto, Draw, Save, Load, or Print. You need only press the key for the first letter of the command. The Auto command will draw designs, one after the other, without pausing for a command in between drawings unless you press a key. The Draw command will draw one design, and then pause for another command. The Save command will prompt you for a filename to save the screen data. Cassette users will enter the name "C:", and disk users will enter the complete name of the disk file, starting with "D:". The screen data will then be saved to cassette or disk. Similarly, the Load command will prompt for a filename, and then load the screen data from that file. The disk save requires 63 sectors, and will only take a few seconds to save and load. Unfortunately, cassette data takes quite a bit longer to save and load. The Print command will copy the screen to an MX-80 printer, using the old block graphics. The screen will be turned off to speed up the time required, as this BASIC print routine runs fairly slowly. Also, since the horizontal resolution of the printer is only 160 dots across, the picture is turned sideways to get as much of it in as possible.

Variables

S(X),C(X): Arrays holding data for sine and cosine values of angles 0-360 degrees.

FLAG: If set, Autorun mode is in force.

CHOICE: Holds menu selection.

FILE$: Holds the name of the file to save or load.

RAMTOP, DL, BYTES, HI, LO, DUMMY: used in Save and Load routines to hold high and low byte values for the address of screen memory, the address of top of memory, and the number of bytes in between to save.

XMAX, XMIN, YMAX, YMIN: Determine how many screen rows and columns you wish to dump to the printer.

XX, YY, Z1, Z2, Z3, Z4, Z5, Z6: Temporary variables used to assemble byte values for six bytes into one printer block character.

LINE$: Holds the finished characters to print out as one line.

```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS        ATARI BASIC          SS
SS     'SPIRAL GRAPHICS'       SS
SS     AUTHOR: TOM GIESE       SS
SS MODIFIED BY:SHELDON LEEMON SS
SS     COPYRIGHT (C) 1982      SS
SS SOFTSIDE PUBLICATIONS, INC SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```

Initialize arrays for sine and cosine values.

The first POKE turns off DMA to allow the initialization to run more quickly, thus shortening the delay until the drawing begins. Next, arrays S(X) and C(X) are set up for sine and cosine values, and the sine value for angles 0-90 degrees are read in. The rest of the array data is derived from manipulating this data, so that only 90 DATA values are required to fill 720 array elements.

```
110 POKE 559,0
120 DIM S(360),C(360),LINE$(80),FILE$(
20)
130 FOR X=0 TO 90
140 READ A:B=A*0.09:C=A*0.1
150 S(X)=B:S(180-X)=B:S(180+X)=-B:S(36
0-X)=-B
160 C(270+X)=C:C(90+X)=-C:C(90-X)=C:C(
270-X)=-C:NEXT X
```

Alter display list.

This segment changes the screen display from graphics mode 8 to graphics mode "7½," one of the modes not supported by BASIC or the Operating System. It displays 192 lines or 160 pixels. Each pixel is a full color clock wide, and can therefore be any one of four colors, just as in graphics mode 7. One line of text is retained at the bottom for prompts and commands.

```
170 GRAPHICS 8
180 X=PEEK(560)+256*PEEK(561)
190 FOR Y=X TO X+166
200 IF PEEK(Y)=79 THEN POKE Y,78
210 IF PEEK(Y)=15 THEN POKE Y,14
220 NEXT Y
230 FOR Y=X+169 TO X+167 STEP -1:POKE
Y+24,PEEK(Y):NEXT Y
240 FOR Y=X+167 TO X+190:POKE Y,14:NEX
T Y
250 POKE X+194,65:POKE X+195,PEEK(560)
:POKE X+196,PEEK(561)
260 X=160:Y=96:XMIN=66:XMAX=254:YMIN=8
:YMAX=87
270 POKE 559,34
```

Main drawing routine

The COLOR 125:PLOT 1,1 combination is an easy way of clearing both screen and text window — the COLOR 125 sets the value to be plotted to the ATASCII value of the "clear screen" character. Next, colors are chosen randomly, and a random starting point is chosen. The coordinates of this starting point are also used as subscripts for sine and cosine plotting functions. These subscripts are incremented through 200 passes, a compromise figure chosen to allow intricate designs without cluttering the screen. You will notice that although four colors are allowed, the SETCOLOR command just sets these to different brightness values of the same color. This arrangement allows for some contrast, giving a dimension of depth. Separate colors do not show up well when assigned to individual dots of an integrated design such as this one, but the user is free to experiment to develop more pleasing combinations.

```
280 COLOR 125:PLOT 1,1
290 IF FLAG=1 THEN PRINT CHR$(125);"Hi
t any key to pause after drawing":POKE
 764,255
300 COLOR 1:C=INT(16*RND(1)):SETCOLOR
0,C,6:SETCOLOR 1,C,4:SETCOLOR 2,C,8
310 A=100*RND(1):B=100*RND(1)
320 PLOT X,Y:FOR C=1 TO 200
330 N=N+A:IF N>360 THEN N=N-360
340 M=M+B:IF M>360 THEN M=M-360
350 D=S(N)*0.01
360 X=C(M)*D+160
370 Y=S(M)*D+96
380 DRAWTO X,Y:NEXT C
```

Command input routine.

This routine reads the keycode register (764) to see if the A, D, S, L, or P key was hit. If the A key was hit, FLAG is set, and the draw routine will continue until a key is hit. If D was hit, the draw routine will execute once and come back to the command menu. S, L, or P will call the appropriate subroutines for these functions.

```
390 CHOICE=PEEK(764):IF CHOICE<>255 TH
EN 410
400 IF FLAG=1 THEN FOR DELAY=1 TO 100:
NEXT DELAY:GOTO 280
410 TRAP 410:POKE 764,255:PRINT "}AUTO
, DRAW, SAVE, LOAD OR PRINT":FLAG=0
420 CHOICE=PEEK(764):IF CHOICE=255 THE
N 420
430 IF CHOICE=63 THEN FLAG=1:GOTO 280
440 IF CHOICE=10 THEN PRINT CHR$(125):
GOSUB 700:POKE 764,255:GOTO 410
450 IF CHOICE=62 THEN GOSUB 570:POKE 7
64,255:GOTO 410
460 IF CHOICE=0 THEN GOSUB 640:POKE 76
4,255:GOTO 410
470 PRINT CHR$(125):POKE 764,255:GOTO
280
480 DATA 0,17,34,52,69,87,104,121,139,
156
490 DATA 173,190,207,224,241,258,275,2
92,309,325
500 DATA 342,358,374,390,406,422,438,4
53,469,484
510 DATA 499,515,529,544,559,573,587,6
01,615,629
520 DATA 642,656,669,681,694,707,719,7
31,743,754
530 DATA 766,777,788,798,809,819,829,8
38,848,857
540 DATA 866,874,882,891,898,906,913,9
20,927,933
550 DATA 939,945,951,956,961,965,970,9
74,978,981
560 DATA 984,987,990,992,994,996,997,9
98,999,999,1000
```

Save screen to disk or cassette.

This subroutine INPUTs the filename into FILE$, OPENs the file for output, PUTs the value of the color registers in the first 5 bytes of the file, and then uses a Machine Language call to the Central I/O Utility to quickly store *all of* that area of memory from the beginning of screen storage (POKE(88)*256* PEEK(89)) to the top of RAM (POKE(106)*256)). The ability of the Operating System to quickly store or retrieve any number of bytes is not supported by BASIC, but is not hard to use with a few well chosen POKEs, and a USR call to set the X register before jumping to CIOV, the CIO vector. The machine code in the USR call is just PLA, PLA, PLA, TAX, JMP $E456 (CIOV). A more complete explanation and examples are given on page 8-37 of the book De Re ATARI®, *available from APX (the ATARI Program Exchange).*

```
570 TRAP 570:CLOSE #2:PRINT CHR$(125);
"Save to which file";:POKE 764,255:INP
UT FILE$
580 OPEN #1,8,0,FILE$
590 FOR I=708 TO 712:PUT #1,PEEK(I):NE
XT I
600 RAMTOP=PEEK(106)*256:DL=PEEK(88)+2
56*PEEK(89):BYTES=RAMTOP-DL:HI=INT(BYT
ES/256):LO=BYTES-(HI*256)
610 POKE 850,11:POKE 852,PEEK(88):POKE
 853,PEEK(89):POKE 856,LO:POKE 857,HI
620 DUMMY=USR(ADR("hhh*LVd"),16):CLOSE
 #1
630 RETURN
```

Load screen data from file.

This subroutine reverses the process of the previous one. The file is opened for input, and the color registers are restored from the first 5 bytes. Then, the rest of screen data is read in using the same CIO routine. Only the POKE for the command need be changed from POKE 850,11 (write bytes) to POKE 850,7 (read bytes).

This quick dump to disk or cassette can easily be added to almost any program which has, as its object, the display of some type of computer art. It can be used in programs where the pattern is random, such as this one, to preserve an especially nice display, or where the pattern is drawn using slow function plotting, and takes a long time to produce.

```
640 TRAP 640:PRINT CHR$(125);"Load fro
m which file";:POKE 764,255:INPUT FILE
$
650 OPEN #1,4,0,FILE$
660 FOR I=708 TO 712:GET #1,A:POKE I,A
:NEXT I
670 POKE 850,7:POKE 852,PEEK(88):POKE
853,PEEK(89):POKE 856,255:POKE 857,255
680 DUMMY=USR(ADR("hhh*LVd"),16):CLOSE
 #1
690 RETURN
```

Screen dump to MX-80 printer.

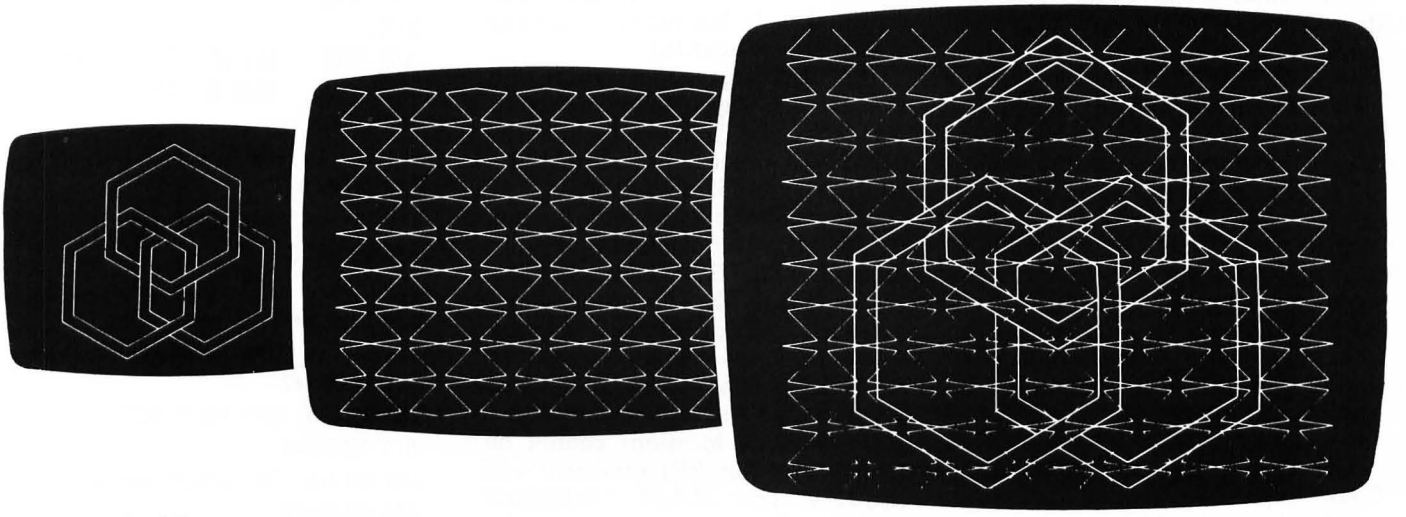
We again turn off the screen to save time, because, though the coding of this routine is compact, it runs rather slowly. The method used is to GET columns of bytes from one edge of the screen to the other, three bytes at a time. The formula in line 800 turns two rows of three bytes into one character of MX-80 block graphics. When LINE$ is filled with a whole row of these characters, the line is printed. The formula used in line 800 performs many additions, rather than multiplying, to save time. This short subroutine can be easily added to any program that produces a graphics mode 8 display, such as the program MAP-WARE, which creates high resolution maps, to conveniently produce hardcopy output. Because the routine is slow, and can only copy 160 of the 192 possible rows on the screen, variables XMAX, XMIN, YMAX, and YMIN can be set by the user to allow a dump of any portion of the screen up to 320 columns by 160 rows. XMAX and XMIN correspond to the actual column numbers. YMAX and YMIN correspond to the row numbers divided by 2, as each printed line contains two rows worth of data. Therefore, YMAX-YMIN must equal 80 or less, when using the printer in normal, 80-column mode. When it is necessary to reproduce the entire screen, a 132-column mode may be used, although the resulting printed image will be somewhat "squashed."

```
700 REM ....SCREEN DUMP....
710 LPRINT CHR$(27);"0":POKE 559,0
720 REM YMAX-YMIN   must be <=80
730 REM YMAX, YMIN = 1/2 SCREEN Y
740 FOR XX=XMIN TO XMAX STEP 3
750 FOR YY=YMAX TO YMIN STEP -1
760 LOCATE XX,YY+YY+1,Z1
770 GET #6,Z3:GET #6,Z5
780 LOCATE XX,YY+YY,Z2
790 GET #6,Z4:GET #6,Z6
800 LINE$(YMAX+1-YY)=CHR$(160+Z1+Z2+Z2
+Z3+Z3+Z3+Z3+8*(Z4+Z5+Z5+Z6+Z6+Z6+Z6))
810 NEXT YY:LPRINT LINE$
820 NEXT XX:POKE 559,34:RETURN
```

ATARI® SWAT TABLE FOR: SPIRAL GRAPHICS

LINES	SWAT CODE	LENGTH
100 - 210	PX	469
220 - 310	EC	534
320 - 430	QL	422
440 - 540	YN	524
550 - 640	YY	561
650 - 760	QK	409
770 - 820	QR	179

FLIKER FLIKER FLIKER



by Duane L. King

***Fliker* is a graphics program for an ATARI® 400/800 with 24K RAM (tape) or 32K RAM (disk). See the end of this article for modifications for 16K RAM (tape) or 24K RAM (disk).**

Fliker provides a way to get 16 simultaneous colors on the screen in *any* graphics mode! Two independent graphics displays are alternately displayed every 60th of a second. This rapid alternation is not readily apparent to the viewer except for a slight "flikering" of the screen.

Fliker demonstrates one of the many effects achievable through the use of multiple-display graphics. The modified version produces mode 5 graphics displays. The two displays produced by *Fliker* are mode 7 graphics displays. Each contains three colors, plus the background color. Since both displays use the same background color, the total number of primary colors is seven. Where two primary colors (one color in one display and the other color in the other display) overlap, a third color is produced. Nine different overlap colors may be produced from the six primary drawing colors. Thus, *Fliker* displays a total of 16 colors.

Fliker takes about three seconds to load in the machine code routines and initialize the two mode 7 displays. Then, the screen will turn gray and *Fliker* will start drawing two different pictures that will display simultaneously. The first display is a diagonal line of small squares drawn in quad or biaxial symmetry. The second is four identical snowflake-like patterns. Since these two different displays are symmetrically related, the interaction of the colors and patterns between the two displays produces some striking effects. As *Fliker* draws, the colors in the two displays are randomly varied. The background color will not change automatically, but you can change it by pressing the OPTION key. Every time you press OPTION, the background color register value will advance. Continually pressing the OPTION key causes all 16 colors to be displayed sequentially. The SELECT key allows you to look at the individual displays. Press the SELECT key, and you will be looking at the first display with the squares in it. Press SELECT again and the snowflake-like display will appear. Press SELECT a third time, and the two displays will alternate.

The sounds are determined by the colors being displayed. A note value is assigned to each color register (C,E,G and high C for color registers 0 through 3 respectively). The timbre and, sometimes, the octave of the note, is determined by the color value. The result is an intriguing collection of melodic patterns mainly in the key of C.

Due to the impact of the machine code routines on the entire program, we'll start in the basement with the interrupt routine and the two support routines, working our way upstairs to the BASIC subroutines and main program.

The television set displays a complete picture every 60th of a second. The CTIA or GTIA chip (display processor) produces a non-maskable interrupt (NMI) for the CPU every time the TV is ready to start another frame. This is called the Vertical Blank Interrupt (VBI). All Vertical Blank Interrupt processing passes through the RAM vector VVBLKI (location 0222 hex) to the OS ROM VBI processing code.

The most direct way to insure that the two graphics displays are alternated every 60th of a second is to intercept the VBI by modifying VVBLKI. A special routine, SETVBV, is in the OS ROM for exactly this purpose. SETBV is also used to "let go" of the VBI and return that processing entirely to the OS ROM code.

At this point, three machine code routines have been identified:

1 — a routine to force VBI processing to pass through our routine.

2 — a routine to return VBI processing to the OS ROM code.

3 — a VBI routine to alternate the two graphics displays.

The SETVBV (set vertical blank vector) routine must be called with specific values in the CPU registers. These values tell SETVBV what to do and the new vector value. The accumulator must contain 6. The X-register must contain the most significant byte (MSB) of the starting address of our VBI routine. The Y-register must contain the least significant byte (LSB) of the starting address of our VBI routine. SETBV is called with the JSR instruction after the A, X and Y registers have been set. When SETBV returns, our VBI routine will be entered instead of the VBI routine in the OS ROM.

Here is the actual code for the VBI insertion routine:

```
  *= $06D3
SETBV = $E45C
  PLA  REMOVE ARG COUNT
  LDA #06
; OUR VBI ROUTINE STARTS AT $066A
  LDX #06  MSB OF VBI ROUTINE
  LDY #$6A LSB OF VBI ROUTINE
  JSR SETBV
  RTS  RETURN TO BASIC
```

This routine may be loaded anywhere in memory. For convenience, it was assembled at location 1747 decimal. See line 30022 for the DATA statement equivalent of this routine.

This routine is called from BASIC with the USR command: U = USR (1747). The value returned by USR in variable U may be discarded because it has no meaning here.

To return VBI processing to the VBI processor in the OS ROM, the SETBV routine is called with the OS VBI routine address in the X and Y registers. It is not necessary to save this vector value before calling our insertion routine because this address is in the OS ROM at hex locations E460 and E461 (LSB and MSB respectively). Here is the actual code to reinstate the OS VBI processor:

```
*= $06ED
PLA  REMOVE ARG COUNT
LDA #06
LDX $E461 MSB OF OS VBI
LDY $E460 LSB OF OS VBI
JSR SETBV  (DEFINED ABOVE)
RTS  RETURN TO BASIC
```

This routine may also be loaded anywhere and was assembled at location 1773 decimal. See line 30026 for the DATA statement equivalent of this routine.

This routine is also called with the BASIC USR command: U = USR (1773). Just as before, the value returned in U has no real meaning and may be discarded.

Our VBI processing routine will flip the displays, load the color registers, and jump into the OS ROM VBI processing routine. This strategy minimizes the amount of machine code that has to be written to exchange the displays and color registers. The OS ROM VBI processor loads the hardware color registers and display list pointer from the RAM "shadow register" equivalents. The RAM copies may be both read and written but the actual hardware locations cannot be read from. Our VBI processor will modify the RAM "shadow registers" and then jump into the OS ROM VBI routine where these shadow registers are copied into the hardware locations. By using this approach, the BASIC SETCOLOR command becomes useless because it tries to modify the RAM color registers that our VBI processor modifies every vertical blank interrupt. The BASIC GRAPHICS command also becomes useless for the same reason.

The screen colors and graphics modes are controlled by entries in a table that our VBI processor copies in the "shadow registers" every VBI. We can control the colors in both displays by modifying these entries with our application program. This table is fifteen bytes long and is divided into three major sections.

1 — display 1 shadow registers

2 — display 2 shadow registers

3 — display flip switch

The display flip switch is a variable used by our VBI processor to keep up with which display goes on the TV screen next. A value of 0 indicates that display 1 is next, and a value of non-zero indicates that display 2 is next. This variable is toggled by our VBI processor and should not be bothered by application programs. The display shadow registers consist of two bytes for the display list address followed by five bytes for the color register values.

Our VBI routine tests the switch. If it is zero, the display 1 shadow registers are loaded, otherwise the display 2 shadow registers are loaded, the display switch is flipped, and the OS ROM routine is entered.

Here is the actual code for our VBI processor:

```
 JMP $E45F   JUMP TO OS VBI
DOSET2
 LDA SET2    SET UP DISPLAY LIST
             POINTER FOR DISPLAY 2
 STA DLHEAD
 LDA SET1+1  NOW DO MSB
 STA DLHEAD+1
 LDA SET1+2  COPY COLOR REG. 0
 STA COLORB
 LDA SET1+3  COPY COLOR REG. 1
 STA COLORB+1
 LDA SET1+4  COPY COLOR REG. 2
 STA COLORB+2
 LDA SET1+5  COPY COLOR REG. 3
 STA COLORB+3
 LDA SET1+6  COPY COLOR REG. 4
 STA COLORB+4
 LDA #0      FLIP DFSW
 STA DFSW
 JMP $E45F   JUMP TO OS VBI
; THE 15-BYTE TABLE STARTS HERE
 *= $06DE  (1758 DECIMAL)
 .BYTE 0,0,0,0,0,0,0
 .BYTE 0,0,0,0,0,0,0
 .BYTE 0
```

This routine is *very* address dependent. It was assembled to be loaded at decimal address 1642 and must be loaded starting at that address. See lines 30000 to 30020 for the DATA statement equivalent of the VBI routine. Line 30024 shows the DATA statement representation of the VBI shadow register table. This BASIC loader for the machine code routines for display swapping is called by GOSUB 20000 and takes about three seconds to load in the three VBI routines. The three second delay is the reason for the "ONE MOMENT PLEASE..." message that displays at the top of the screen.

After the VBI routines have been loaded into memory, the two displays must be made. The BASIC GRAPHICS command will create a display list and a display data area for the graphics mode you specify. The start of the display list is placed in decimal locations 560 and 561. The start of the display data area is saved in locations 88 and 89. The

location of the display lists must be saved in the table for use by our VBI processor. The location of display data areas must be saved for use by the application program. The address in locations 88 and 89 is used by the PLOT and DRAWTO command as the upper left corner of the screen. By manipulating this address, we can draw in either of the two displays. The display initialization procedure is:

1 — make display 1 with GRAPHICS command
2 — D1L = PEEK(88):D1H = PEEK(89)
(save the starting address of display 1 data area)
3 — POKE 1758,PEEK(560):
POKE 1759,PEEK(561)
(save address of display list 1 in VBI table)
4 — push memory down 4096 bytes for display modes 6 or less, push memory down 8192 bytes for display modes 7 and 8:
POKE 106,PEEK(106)-16 or
POKE 106,PEEK(106)-32
5 — make display 2 with GRAPHICS command
6 — D2L = PEEK(88):D2H = PEEK(89)
(save the starting address of display 2 data area)
7 — POKE 1765,PEEK (560):POKE 1766,PEEK(561)
(save address of display list 2 in VBI table)
8 — U = USR(1747)
(inserts our VBI processor ahead of the OS ROM code)

The BASIC subroutine that will set up two displays in the same graphics mode is located in lines 19000 to 19010. The desired graphics mode is assigned to the variable GM (i.e., GM = 7) before calling this subroutine with the GOSUB 19000 command. This BASIC subroutine calls the VBI loader subroutine and then sets up the pointers for the two displays in the VBI table.

If you want the two displays to be in different graphics modes, you could "hard-code" the graphics mode in this subroutine instead of using the variable GM. Also, the graphics mode will need to be POKEd into location 87 before drawing in either graphics display. The value in location 87 is used by PLOT and DRAWTO to compute screen addresses from the X and Y coordinates used by these graphics commands.

The VBI color shadows have not been initialized up to this point. To initialize the color shadow registers for display 1, POKE 1760 + CR, HUE*16 + LUM where CR is the color register number (0 through 4). HUE and LUM are the values normally used with the BASIC SETCOLOR command. To initialize the color shadow registers for display 2: POKE 1767 + CR,HUE*16 + LUM. The general method for drawing in display 1 is:

1 — POKE 88,D1L:POKE 89,D1H and optionally POKE 87,MODE
2 — do your COLOR, PLOT, and DRAWTO commands for display 1

To draw in display 2:

1 — POKE 88,D2L:POKE 89,D2H and optionally POKE 87,MODE
2 — do your COLOR, PLOT, and DRAWTO commands for display 2

The three VBI subroutines and the two support routines in BASIC give you a platform on which to build very sophisticated multiple-display pictures. Since this method uses two full displays (each display has its own display list as well as the display data area), you can incorporate display list interrrupts with a separate DLI routine for each of the displays. (This requires some additional code in the VBI processor presented here.)

This presentation only scratches the surface. It is up to you to explore the full capabilities of your ATARI® computer.

To execute *Fliker* on a 16K cassette or 24K disk based system, type in the following lines (replacing the ones in the listing):

```
0 GM=5+16:GOSUB 19000:GOSUB 2000
400 GOSUB 1900:POKE 88,D1L:POKE 89,D1H:
I=79:J=47:GOSUB 700
451 J=23:X=40:Y=0:GOSUB 1500:J=47:Y=24:
GOSUB 1500:I=39:X=0:GOSUB 1500:I=23:Y=0:
GOSUB1500
700 LX=X1:LY=Y1:U=INT(RND(0)*2)+1
2002 M=47/79:XS=20:YS=12:LCS=7:DM=3:
RETURN
```

Variables

GM: The graphics mode of both displays (set to mode 7 — full screen.
U: Temporary or scratch variable, used for loops, calculations, etc., but **never** holds anything important.
D1L: The low 8-bits of the address of the display data area of display 1.
D1H: The high 8-bits of the address of the display data area of display 1.
D2L: The low 8-bits of the address of the display data area of display 2.
D2H: The high 8-bits of the address of the display data area of display 2.
LUM: The luminosity (intensity) of all foreground colors in both displays (may be user altered — default is 10.)
BLUM: The luminosity of the background color in both displays (2 less than the foreground luminosity.)
CH: Array of four elements (0-3) containing the mode values associated with the foreground colors:

CH(0) low C (243) COLOR 0
CH(1) low E (193) COLOR 1
CH(2) low G (162) COLOR 2
CH(3) C (60) COLOR 3

P: Current pen color used with COLOR command and with POKEs to color registers, etc.
M: Aspect ratio of full screen mode 7 display — slope of the diagonal from (0,0) to (159,95), used by the display 2 plot routine to reflect lines about 4 axes, since two of the axes are on diagonals.
XS: Maximum x-value generated by display 2 coordinate generator (GOSUB 800.)
YS: Maximum y-value generated by display 2 coordinate generator (GOSUB 800.)
LCS: Last console switch settings. Used to detect changes in the positions of the three console switches.
DM: Display mode. DM = 1 display 1 is up; DM = 2 display 2 is up; DM = 3 flipping displays with our VBI routine. Controlled by SELECT key code in subroutine starting at 1000.
CC: The number of consecutive lines/squares to draw with the current pen, set between 2 and 10.
CCC: Randomly generated color register value, intensity is always LUM (GOSUB 1900.)
I: Maximum x-value for reflecting squares or lines in both drawing routines (GOSUB 1400 and GOSUB 1500.)
J: Maximum y-value for reflecting squares or lines in both drawing routines (GOSUB 1400 and GOSUB 1500.)
LX: Last x-value for display 1.
LY: Last y-value for display 1.
X1: New x-value for display 1.
Y1: New y-value for display 1 (The coordinate pairs (LX,LY) and (X1,Y1) define the upper left and lower right corners of the squares.)
LX2: Last x-value for display 2.
LY2: Last y-value for display 2.
X12: New x-value for display 2.
Y12: New y-value for display 2 (The coordinate pairs (LX2,LY2) and (X12,Y12) define the start and end of the line.)
Y2: Saved value of Y1 (to restore after drawing the square.)
X: The minimum x-value for reflection of lines in display 2 (GOSUB 1500.)
Y: The minimum y-value for reflection of lines in display 2 (GOSUB 1500) (GOSUB 1500 draws inside an area of the screen defined by (X,Y) and (I,J) — upper left and lower right respectively.)

```
SS SS SS SS SS SS SS SS SS SS SS
SS                           SS
SS        ATARI BASIC        SS
SS         'FLIKER'          SS
SS     AUTHOR: DUANE KING    SS
SS     COPYRIGHT (C) 1982    SS
SS SOFTSIDE PUBLICATIONS, INC SS
SS                           SS
SS SS SS SS SS SS SS SS SS SS SS
```

Calls subroutines to load the VBI routines, initialize the displays, and the color register shadows and several program variables.

```
0 GM=7+16:GOSUB 19000:GOSUB 2000
```

Generate the color count (number of squares/lines to draw before changing pen colors with COLOR command)

```
300 CC=INT(RND(0)*9+2)
```

Change screen color of current pen, set-up to draw in display 1 and set maximum bounds of screen. Also generate endpoints of square.

```
400 GOSUB 1900:POKE 88,D1L:POKE 89,D1H
:I=159:J=95:GOSUB 700
```

Draw square (quad symmetry on entire screen).

```
442 Y2=Y1:FOR Z=LY TO Y1 STEP SGN(Y1-L
Y):LY=Z:Y1=Z:GOSUB 1400:NEXT Z:Y1=Y2
```

Set-up to draw in display 2, set-up drawing area for upper-left quadrant of the screen, and generate new endpoints for line of snowflake.

```
450 POKE 88,D2L:POKE 89,D2H:GOSUB 800
```

Draw line of snowflake in upper right quadrant, then lower right quadrant, then lower left quadrant, and then upper left quadrant.

```
451 J=47:X=80:Y=0:GOSUB 1500:J=95:Y=48
:GOSUB 1500:I=79:X=0:GOSUB 1500:J=47:Y
=0:GOSUB 1500
```

Decrement and test color count to see if it is time to change pens.

```
540 CC=CC-1:IF CC>0 THEN 400
```

Change to next pen.

```
545 GOSUB 900:GOTO 300
```

Generate next pair of coordinates for drawing squares in display 1. The squares move diagonally (left-to-right and top-to-bottom). When the endpoints meet a screen boundary the direction is reversed (creates the effect of the square bouncing off the boundary). Squares are 1 to 4 units on a side.

```
700 LX=X1:LY=Y1:U=INT(RND(0)*4)+1
702 IF LX<4 THEN DX=1
704 IF LX>I/2 THEN DX=-1
706 IF LY<4 THEN DY=1
708 IF LY>J/2 THEN DY=-1
710 X1=LX+U*DX:Y1=LY+U*DY:RETURN
```

Generate new endpoint for drawing the snowflake, saving the precious new endpoints as the current last endpoints (causes every line to be connected to the previous line. The new endpoint is constrained to 1/8 of the snowflake pattern.

```
800 LX2=X12:LY2=Y12
820 X12=INT(RND(0)*XS):Y12=INT(RND(0)*
YS):IF Y12>X12*M THEN 820
830 RETURN
```

Select the next pen color with wraparound from 3 to 0, also reset attract mode flag with the POKE.

```
900 P=P+1:IF P>3 THEN P=0
910 COLOR P:POKE 77,0:RETURN
```

Poll the console keys testing for OPTION and SELECT keys (in that order). The OPTION key increments the background color (1002 and 1010-1012). The SELECT key cycles from both displays, to display 1, to display 2, and back to both displays (1004, 1024-1029). When DM reaches 3, our VBI routine is inserted again, when DM is 1 our VBI routine is removed and display 1's vector is placed in the OS display vector locations. When DM is 2, display 2's vector is placed in the OS display vector locations. When DM is anything else but 3, the appropriate VBI color register shadows are copied into the normal OS color register shadows (1029).

```
1000 U=PEEK(53279):IF U=LCS THEN RETUR
N
1002 LCS=U:IF LCS=3 OR LCS=11 THEN 101
0
1004 IF LCS=5 OR LCS=13 THEN 1024
1006 RETURN
1010 U=PEEK(1764)+16:IF U>255 THEN U=B
LUM
1012 POKE 1764,U:POKE 1771,U:RETURN
1024 DM=DM+1:IF DM>3 THEN DM=1
1025 IF DM=3 THEN U=USR(1747):RETURN
1026 IF DM=1 THEN U=USR(1773):POKE 560
,PEEK(1758):POKE 561,PEEK(1759)
1028 IF DM=2 THEN POKE 560,PEEK(1765):
POKE 561,PEEK(1766)
1029 FOR U=0 TO 2:POKE 708+U,PEEK(1760
+(DM-1)*7+U):NEXT U:RETURN
```

Quad draw routine (2 axes of symmetry) for display 1.

```
1400 GOSUB 1000:PLOT LX,LY:DRAWTO X1,Y
1
1402 PLOT I-LX,LY:DRAWTO I-X1,Y1
1404 PLOT I-LX,J-LY:DRAWTO I-X1,J-Y1
1406 PLOT LX,J-LY:DRAWTO X1,J-Y1:RETUR
N
```

Oct draw routine (4 axes of symmetry) for display 2.

```
1500 GOSUB 1000:PLOT LX2+X,LY2+Y:DRAWT
O X12+X,Y12+Y:PLOT I-LX2,LY2+Y:DRAWTO
I-X12,Y12+Y
1505 PLOT INT(LY2/M)+X,INT(LX2*M)+Y:DR
AWTO INT(Y12/M)+X,INT(X12*M)+Y
1507 PLOT I-INT(LY2/M),INT(LX2*M)+Y:DR
AWTO I-INT(Y12/M),INT(X12*M)+Y
1510 PLOT I-LX2,J-LY2:DRAWTO I-X12,J-Y
12:PLOT LX2+X,J-LY2:DRAWTO X12+X,J-Y12
1515 PLOT I-INT(LY2/M),J-INT(LX2*M):DR
AWTO I-INT(Y12/M),J-INT(X12*M)
1517 PLOT INT(LY2/M)+X,J-INT(LX2*M):DR
AWTO INT(Y12/M)+X,J-INT(X12*M)
1520 RETURN
```

Generate random color with intensity LUM and "flip a coin" to see which display gets the color change. Set the sound using the generated color value and copy the color value into the OS color register shadow if our VBI routine is not executing.

```
1900 CCC=INT(RND(0)*16)*16+LUM:U=INT(R
ND(0)*10):IF U<5 THEN POKE 1767+P,CCC
1902 IF U>5 THEN POKE 1760+P,CCC
1904 SOUND P,CH(P),10,INT(CCC/5):IF DM
<>3 THEN POKE 708+P,CCC:POKE 712,PEEK(
1764)
1906 RETURN
```

Initialize important variables and the color shadow registers of the VBI table.

```
2000 LUM=10:BLUM=LUM-2:FOR U=1760 TO 1
762:POKE U,INT(RND(0)*16)*16+LUM:POKE
U+7,INT(RND(0)*16)*16+LUM:NEXT U
2001 POKE 1764,BLUM:POKE 1771,BLUM:DIM
 CH(3):CH(0)=243:CH(1)=193:CH(2)=162:C
H(3)=60:P=1:COLOR P
2002 M=95/159:XS=40:YS=24:LCS=7:DM=3:R
ETURN
```

The following lines are documented in detail in the preceding article.

```
19000 GOSUB 20000:GRAPHICS GM:D1L=PEEK
(88):D1H=PEEK(89)
19002 POKE 1758,PEEK(560):POKE 1759,PE
EK(561)
19004 U=PEEK(106)-16:IF (GM>6 AND GM<1
6) OR GM>22 THEN U=U-16
19006 POKE 106,U:GRAPHICS GM:D2L=PEEK(
88):D2H=PEEK(89)
19008 POKE 1765,PEEK(560):POKE 1766,PE
EK(561)
19010 U=USR(1747):RETURN
20000 ? "}ONE MOMENT PLEASE..."
20002 FOR U=1642 TO 1785:READ Z:POKE U
,Z:NEXT U:RETURN
30000 DATA 173,236,6,208,50,173,222,6,
141,48,2,173,223,6,141,49,2,173,224,6
30004 DATA 141,196,2,173,225,6,141,197
,2,173,226,6,141,198,2,173,227,6,141,1
99
30008 DATA 2,173,228,6,141,200,2,169,1
,141,236,6,76,95,228,173,229,6,141,48
30012 DATA 2,173,230,6,141,49,2,173,23
1,6,141,196,2,173,232,6,141,197,2,173
30016 DATA 233,6,141,198,2,173,234,6,1
41,199,2,173,235,6,141,200,2,169,0,141
30020 DATA 236,6,76,95,228
30022 DATA 104,169,6,162,6,160,106,32,
92,228,96
30024 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
30026 DATA 104,169,6,172,96,228,174,97
,228,32,92,228,96
```

ATARI® SWAT TABLE FOR: FLIKER

LINES	SWAT CODE	LENGTH
0 - 700	FO	504
702 - 1002	RF	355
1004 - 1404	TD	458
1406 - 1906	FI	502
2000 - 19004	YH	513
19006 - 30016	UW	538
30020 - 30026	MS	148

Paint

Reviewed by J. Harmon Grahn

by Superboots®, Capital Children's Museum, 800 Third Street, NE, Washington, DC 20002. From Reston Publishing Co., 11480 Sunset Hills Road, Reston, VA 22090. System requirements: 48K ATARI® 400/800 with disk drive and color television or monitor. Suggested retail price: $39.95.

Although creative artists may have had a traditional aversion to the world of computers until recently, startling developments in computer imaging may change all that — and bring into existence a popular new medium of fluid, creative expression.

ATARI® users now have an opportunity to explore some of the rich possibilities open to computer artists at an astonishingly affordable price. *Paint* is a program which allows you to do just that: paint colorful pictures on your color screen and save them for future retrieval on a disk. Briefly, *Paint* is to painting what word processing is to writing.

Paint is menu-driven with one-letter mnemonic commands. It is designed so a child can use it, but its fascination is by no means limited to children. You paint with a joystick (a handicap if you've ever worked with a digitizing pad and stylus — but then there's an enormous price difference between an $800 input device and a simple $10 joystick, so you are amply compensated for the inconvenience).

Virtually every letter on the keyboard represents an easily associated mnemonic command, from "A" for "Art Show" (retrieval of stored compositions) to "Z" for "Zoom" (two levels of magnification: 4X and 6X). You can imagine the astronomical variety of options available to you when you mix these commands in different ways.

To begin, you are presented a blank screen with a palette of ten little "paint pots" at the bottom. One contains the background screen color (for erasures), so you have nine colors to work with: three solid and six textured. With the joystick, you manipulate your cursor and place it in the color of your choice; press the firebutton. You have just "dipped your brush" and now have an inexhaustible supply of your chosen color. Move the cursor around the screen: any time you press the firebutton, and as long as you keep pressing it, your cursor will leave a linear trace of your chosen color. Dip your brush again, and paint freely over your first lines, or between them, or whatever. To erase, of course, just dip your brush into the background color and retrace an existing line. (To erase your whole screen, type an "E". *Paint* prompts for a confirmation, to which you may reply by typing a "Y" or an "N".)

The mnemonic command, "W" for "Width of Brush" and an integer (1-9) selects one of nine different brush sizes, from the equivalent of a pencil line to a wide swath of electric color. Additionally, there are nine brush types ("B" for "Brush") which give the leading edge of your brush stroke different profiles for some interesting effects.

For instance, you can select a calligrapher's brush which will enable you (with some practice) to create beautiful Gothic script with thin diagonal, and thick vertical or horizontal strokes. Other brushes give you solid horizontal and vertical strokes, but render diagonals in a surprising checkerboard pattern. Interesting things happen visually when you draw a closed figure, such as a circle, with a wide brush of this type.

This, however, is only the beginning of the rich possibilities awaiting you. For me, one of the most exciting moments came when I began to explore *Paint's* color mixing capabilities. There is a graphic color/texture menu which enables you to subtly blend the shades and hues in your palette. A few minutes (or, preferably, hours) spent mixing colors can teach you volumes about the visual interactions among them. If you have drawn something on your screen, you might enter the color menu and change your palette around. Presto! Your composition is now rendered in an entirely new color scheme. Surprisingly, it may look quite different to you — not only in color, but in shape as well. The Impressionists spent years exploring these subtle color relationships with paint and canvas. Now you can do it in moments with electronic speed and facility.

Included with the color menu is a texture menu with seemingly endless variety. Use the joystick to isolate within a movable square a small portion of variably textured mosaic screen area. The colors in the mosaic may be altered with the same techniques used to alter solid colors. A mosaic texture pattern, once isolated and defined, is thus captured in a designated paint pot as a "color" in your palette and may be used like any solid color. The mosaic pattern will be repeated along any line you trace with a brush dipped in its "color." The wider your brush, the more of your pattern will be represented in a stroke. The effect can be something like Seurat's visual mixing of points of pure color.

Using the Zoom function, you may greatly magnify your composition for detailed work. At 16X, a thin line becomes a chain of sharply defined square picture elements, or pixels, which are the smallest units of color in your composition. With patience and skill, you may manipulate your composition one pixel at a time.

Paint operates at a basic resolution of 160 X 80 pixels. The "Z" command is a switch that, when pressed once, takes you to 4X magnification and a resolution of 80 X 40 pixels. Pressed a second time, you're at 16X magnification and a resolution of 40 X 20 pixels. "Z" again, and you're back to 4X. Again, and you're back to your 160 X 80 pixel, full screen.

Paint's palette has 128 different shades and hues. That breaks down to 16 different hues, each with 8

levels of luminance. The texture menu offers some 7,000 or 8,000 different textures. Nobody knows for sure just how many are possible.

So, what is *Paint* good for? Is it a toy, or a tool? The manual, by educator Alex Packer, is written for children and is quite unlike the documentation that comes with most software packages. It is breezy and light in tone, and carefully calculated to disarm the wary or skeptical computer illiterate, whether child or adult. For example, in going through the alphabetical list of mnemonic commands, you will come upon: "No. How to say No when asked to confirm whether to erase a picture: 1. Press N for No. 2. Stamp your feet. 3. Pull your hair and yell No!! No!! No!! Never!! I won't erase!! Never!! 4. Hold your breath. Turn red in the face..." Or again, when you get to "Vee. Vee don't seem to have a V here. Tee hee." (There isn't a "T" mnemonic either — "unless you have a cup.")

Yet the book economically and efficiently gets across all the information needed to operate *Paint* — with room to spare for a chapter on "How Computers Work," a lightning, and in some ways enlightening, overview of art history; "From Cave to Computer," a discussion of current trends in "Computer Imaging;" and a more detailed look at the work of three contemporary "Computer Artists." The final chapter, "Idea Shop," is intended to shake loose old concepts of art and get your juices flowing into this new and very different creative medium. Profusely illustrated and with eight pages of color plates, the book is only somewhat over 100 pages in length.

"If I could influence your review at all," co-creator of *Paint*, Guy Nouri, told me, "it would be to say that what went into the program was the effort of several people above and beyond the call of duty." Eric Podietz deserves the lion's share of the credit, according to Nouri. "Eric is a master programmer who kept 120 pages of hexadecimal code in his head throughout the project," Nouri said.

Paint was conceived and designed by Nouri and Podietz, and was developed and extensively tested at the Capital Children's Museum in Washington, D.C.. The book was written by Alex Packer in collaboration with Heather Harney at the Museum, and Guy Nouri. The program underwent months of prerelease testing at the Museum — and something needs to be said about that.

The Capital Children's Museum (CCM) is a hands-on museum created and designed especially for children. It became intensively involved with computers early in 1981 when ATARI® donated 30 computer systems to what became the Museum's Future Center, a computer classroom.

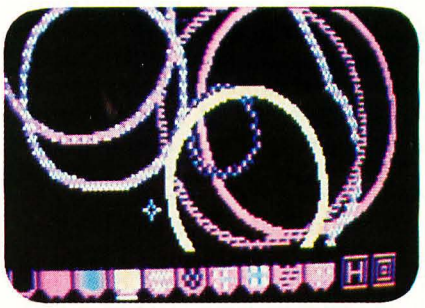
Since its inception in 1974, CCM has grown rapidly and now serves over 200,000 visitors per year. Schools from all over the District of Columbia and surrounding suburbs regularly send classrooms of children to the Future Center to gain hands-on experience with computers. This is the environment in which *Paint* was developed and debugged. It has withstood the trial by fire, which bodes well for those who might be worried about software support.

On this count, Nouri praised Reston Publishing Co.. *Paint* is Reston's first venture into the software market. "They have been very supportive in getting the product together," Nouri said. "They know the program very well and can answer most questions that may arise."

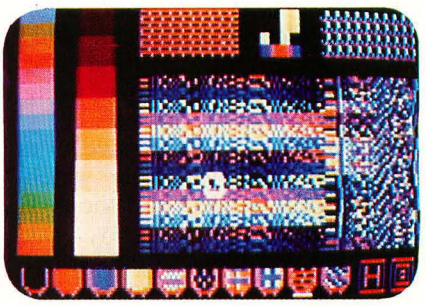
For questions on the software, Nouri suggested three levels of inquiry: 1. Reston Publishing Co. 2. Heather Harney at CCM. "Heather knows every corner of the program," he said. "She probably knows it better than I do." 3. And "for really esoteric questions," Nouri said, "you can always call Eric."

So what's *Paint* good for? That's up to you. Considering that commercial computer paint systems currently run at upwards of $100,000 — for forty dollars I'd give *Paint* a five-star rating. In fact, if you're seriously inclined to explore the world of computer imaging, this software package might make it well worth your while to invest in an ATARI®.

The opening sequence of *Paint* illustrates some of the color and pattern capabilities of the program in a startling animation.



***Paint's* Circle command allows you to overlay circles of many sizes and colors with ease by choosing a "paint pot" and plotting only two points.**



The color/texture menu allows you to mix hundreds of color combinations in thousands of textures to fill your "paint pots."

One of the pictures contained in the "Art Show" portion of *Paint* shows the commercial art capabilities of the program.

3-D Supergraphics

Reviewed by S. Berfield

by P. Lutus (United Software of America). System requirements: 40K ATARI® 400/800 with cassette or disk (recommended). Suggested retail price: $39.95.

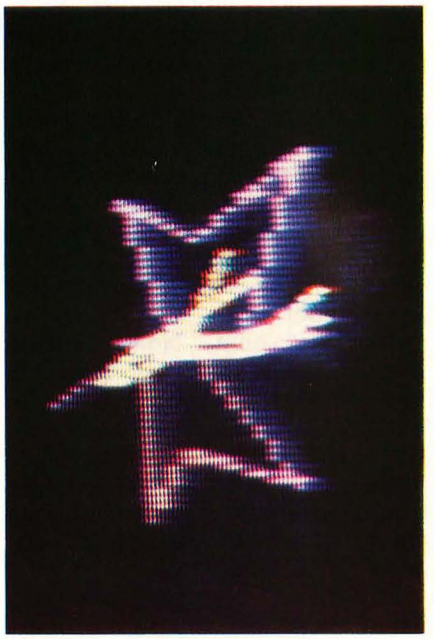
I remember my first exposure to computer games. As for most people, it was a mainframe *Star Trek* with no graphics. Since the sum of my previous exposure to computers was in the realm of science-fiction and movies, I was a bit disappointed. As I have learned more about the capabilities of computers I have come to appreciate the difficulties of producing advanced graphics. Understanding does not always help, however. I despaired of ever being able to create 3-D images, let alone animate them, without purchasing a $10,000 system.

P. Lutus and United Software of America have come to the rescue with *3-D Supergraphics and Color Game Development System*. This package allows the creation and manipulation of 3-D shapes using BASIC Print # commands.

The packaging is impressive — a professional looking binder which holds both the manual and the disk or cassette. Unfortunately, the manual has one serious flaw for the ATARI® user: it's written for the Apple™. The program was originally written for the Apple™ and the only concession to ATARI® owners is a four-page appendix noting system specific differences. This makes for rather tedious reading. I suggest that you first go through the manual and write in all the appropriate changes. This speeds things up considerably. The appendix also contains loading instructions for the ATARI®.

The cassette uses both sides. Side "A" holds the loader, the "GRFBAS" binary file, a demonstration program, and the shape table used by the demo. These are all chained and load each other sequentially. The other side holds the utility and a shape development program which makes the creation of shapes extremely simple. The shapes thus created can be saved to tape and used in other programs.

To use *Supergraphics* from BASIC, you must first poke the utility into memory. Then, all that needs to be done is X = USR(22016), and OPEN #IOCB,12,0,"G:". This enables the utility and opens a control block for input and output to the new device "G:". The commands to manipulate shapes are all in the form of PRINT #IOCB;"% .....".



As many shapes may be defined as there is memory space available. The space for this extends from the top of the program to $4FFF (20479). A shape at memory 17000 would be displayed by the command PRINT #ICOB + "%SH17000". This displays at the center of the screen. Manipulations available include rotation, scaling, translation, position, and color. (Yes, there are three distinct colors available in GR.8!) By the use of loops which execute at machine speed, up to 30 frames per second can be displayed. Another feature is the creation of a second graphics screen. By creating different versions of a shape on each screen and alternating between them, (a very easy thing with this utility) parts of a shape can be made to move.

Table 1 shows the commands available in "GRAFBASIC". All are implemented by the previously given command format.

Table 1

RX ROTATE X
RY ROTATE Y
RZ ROTATE Z
TX TRANSLATE X
TY TRANSLATE Y
TZ TRANSLATE Z
PX POSITION CENTER X
PY POSITION CENTER Y
SX SCALE X (+-31)
SY SCALE Y
SZ SCALE Z
SC SCALE X,Y,Z
CG COLOR GRAPHICS (1-3)
CT COLOR TEXT (1-3)
VT VERTICAL TAB
HT HORIZONTAL TAB
YD ENABLE DRAW
ND DISABLE DRAW
NW NEW(CLEAR)
QT QUIT
SH SHAPE(ADDRESS)
SH+ SHAPE LEAVES OLD SHAPE ON SCREEN 2

All in all, I'm satisified with this package. My only complaint concerns the documentation, and that (even though it is for the wrong machine) is so thorough that it's not too difficult to understand. The listings are fairly useful in that the command format is similar for both computers. After a day of experimenting, I was able to emulate the examples in the demo program. A few days later, I had started on my first 3-D game. This program is one of the few packages which really takes advantage of the graphics capabilities of the ATARI®. For the money, I can't think of anything to beat it.

PASSAGE RESEARCH
945 Turquoise St., Ste. G
San Diego, CA 92109
(714)488-5358

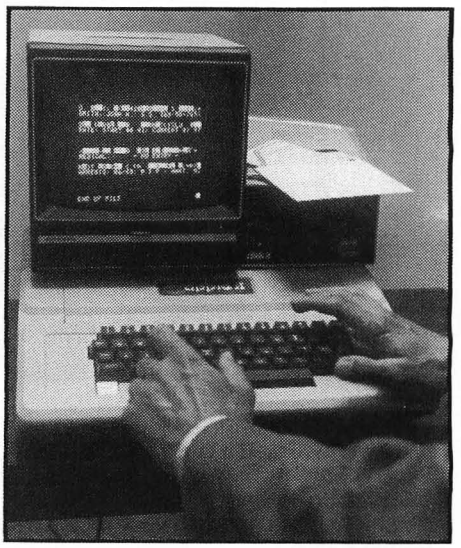
CLASSIFIED™, a software package for Apple™ users, encrypts and decrypts the information stored in any standard DOS 3.3 diskette file.

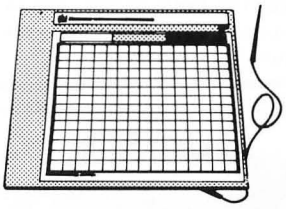
The use of personal computers to store confidential information is a brand new application brought about by the introduction of low cost encryption software. For example, selected phone numbers can be kept private by storing them in an encrypted file. Other prime candidates for file encryption are financial data files, such as those created by VisiCalc®, and program files containing "trade secret" software.

For the encryption algorithm, *CLASSIFIED*™ duplicates the DES (Data Encryption Standard) enciphering computation in Machine Language code. The encryption key is entered from the keyboard as a nine character combination of letters, numbers, or symbols.

The encryption manual supplied with *CLASSIFIED*™ helps the user to encrypt a file within ten minutes of first opening it. Only twenty four pages long, the manual is "to the point" for the busy professional, yet comprehensive enough for the most dedicated computer hobbyist.

System requirements for *CLASSIFIED*™ are a 48K Apple II+™ or Apple III™ (Apple II™ emulation mode) and one disk drive. It is now available for $39.50 (postpaid to anywhere in the U.S. or Canada) from Passage Research.





vs.

RAINBOW COMPUTING, INC.
19517 Business Center Dr.
Northridge, CA 91324
(213)349-0300 (in Calif.)
(800)423-5441

The POOR MAN'S GRAPHICS TABLET features an almost unlimited palette of colors, which can be applied in more than 59 textures. It also contains a unique marking feature which allows you to trace transparencies overlaid on the screen of your monitor or television set. Simply mark a few points on the figure, and the *POOR MAN'S GRAPHICS TABLET* traces the figure for you. Full shape table functions are included, as well as full manipulation of shapes and pictures. All manipulation and drawing are done through the Apple™ keyboard for greater accuracy.

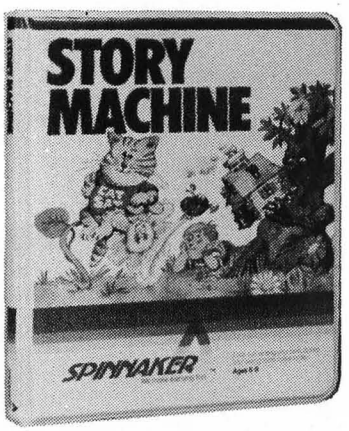
The POOR MAN'S GRAPHICS TABLET requires an Apple II™ with 48K, Applesoft in ROM, and a disk drive with DOS 3.3. Retail price is $49.95 (disk), plus $2.50 (U.S.) or $10.00 (foreign) for shipping and handling.

STERLING SWIFT PUBLISHING COMPANY
1600 Fortview Road
Austin, TX 78704

The Apple II™ *DIAL-A-COMMAND,* a two-sided reference tool, weighs less than one ounce, exactly fits the hand, and "speaks" commands an Apple II™ or II+™ understands. The user can select from 166 functions and thereby automatically dial the proper command to be entered for the following operations: Editing and Format, Input/Output, Flow of Control, Sequential Text File, System and Utility, Graphics and Game Controls, Machine Language Text File, Access, Housekeeping, Math, and Arrays and Strings.

Also listed on the *DIAL-A-COMMAND* are keystroke entries for Algebraic Operators, Relational and Logical Operators, and Simple Variables.

DIAL-A-COMMAND will soon be available from dealers, and may now be ordered directly from Swift Sterling Publishing Company. Retail price is $9.95.



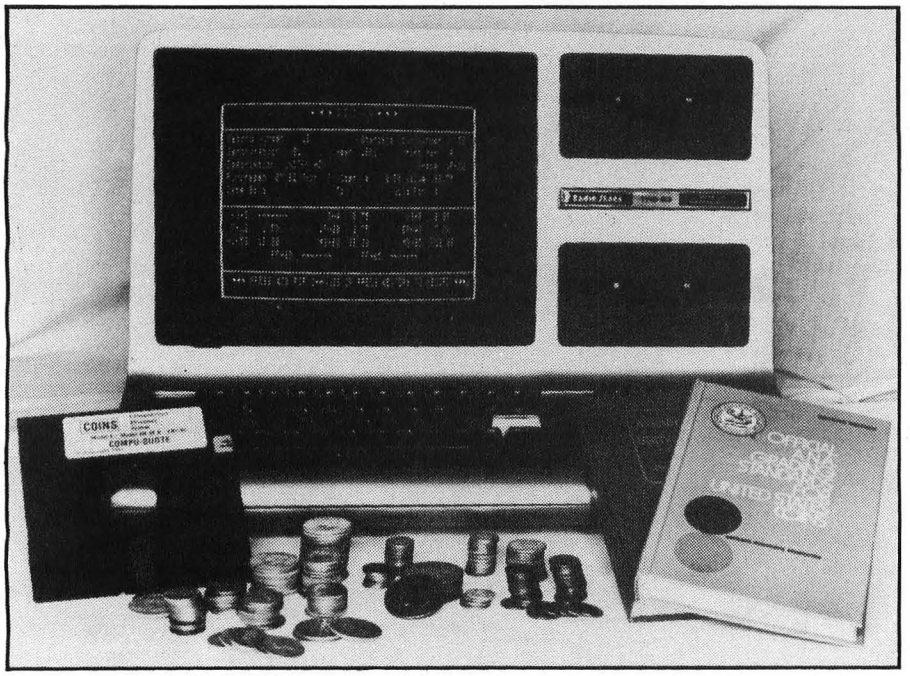
SPINNAKER SOFTWARE CORPORATION
215 First Street
Cambridge, MA 02142
(617)868-4700

Face Maker and *The Story Machine* are two new learning games designed for children ages four to nine. Both are geared toward a balance between amusement and learning, with a minimal amount of required text material. They are also designed to foster understanding and enjoyment of the use of a personal computer.

Face Maker provides an animated format for familiarizing a child with the graphics capabilities of a computer. By choosing from a varied menu of eyes, ears, noses, mouths, etc., the user composes a face which he can then animate with smiles, winks and wiggling ears. A memory development game, in which the child tries to repeat the computer's series of facial expressions, is also incorporated into the program.

The Story Machine provides an opportunity for children to develop and strengthen their sentence and paragraph skills. Sentences and paragraphs, composed by the user from a substantial list of nouns, verbs, prepositions and other parts of speech, are animated with full color graphics and sound. *The Story Machine* also provides keyboard practice and introduces the child to the editing capabilities of the computer.

Both *Face Maker* and *The Story Machine* are now available for the Apple II+™ with 48K of memory and disk drive, and will soon be available for the IBM PC® and the ATARI 800®. A color monitor is recommended. Suggested retail price, for both games, is $34.95. They are for sale in retail microcomputer stores across the country.

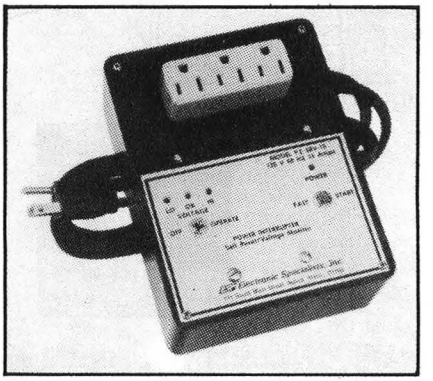


COMPU-QUOTE
6914 Berquist Avenue
Canoga Park, CA 91307
(213)348-3662

COINS (COmputerized INventory System) is a sophisticated computer program for coin collectors. It is intended to run on either the TRS-80® Model I or Model III, 48K, 2 disk machine.

Using this program, the collector may enter information on the coins in his collection effortlessly and produce various printed reports which can be used for personal investment and tax purposes. The strongest feature of this unique program is the included file of information and market prices on 1500 of the most common U.S. coins in all grades. This information is updated and distributed on a quarterly basis and provides for automatic re-evaluation of the collection. Non-standard coins may also be listed and maintained by the user. More information on the program is available by requesting a brochure from Compu-Quote.

The program is priced at $95, which includes the latest Value file. Quarterly updates are available for $25 each.



ELECTRONIC SPECIALISTS, INC.
171 South Main Street
Natick, MA 01760
(617) 655-1532

The *SELF-RESET POWER INTERRUPTER* disconnects AC power from controlled apparatus, should AC Line Voltage be disrupted or exceed pre-set safety limits. A four minute time delay, followed by automatic self-reset, helps avoid wide voltage fluctuations associated with power line malfunctions. An optional Line Voltage Monitor is available.

Intended for installations operating unattended for long periods, the *SELF-RESET POWER INTERRUPTER* provides safety and protection for equipment and users. It connects to the AC line with a standard three prong plug, and can accomodate a 15 amp resistive load or a 10 amp inductive load.

The *SELF-RESET POWER INTERRUPTER* is offered at a retail price of $185.95. With the optional Line Voltage Monitor, the price is $205.95.

MARKET/SIDE

Advertiser's Index

Advertising Representatives
Sue Rowland
Bob Mackintosh
(603) 673-0585

National Representative
Christopher Smith
Inter-Marketing Associates
(603) 827-3976

Associate Publisher
Nancy Lapointe

Publisher
John G. Grow

Next time in SoftSide #34.....

All of this and much more will be ringing your way in Issue 34 of SoftSide. Computer sound and music will be our focus and there are some special treats in store.

Alpha Syntauri and Soundchaser

— These two synthesizers for the Apple™ have been causing a lot of buzz (and music) at recent computer shows. We've put them side by side for a comparative review and will report the results.

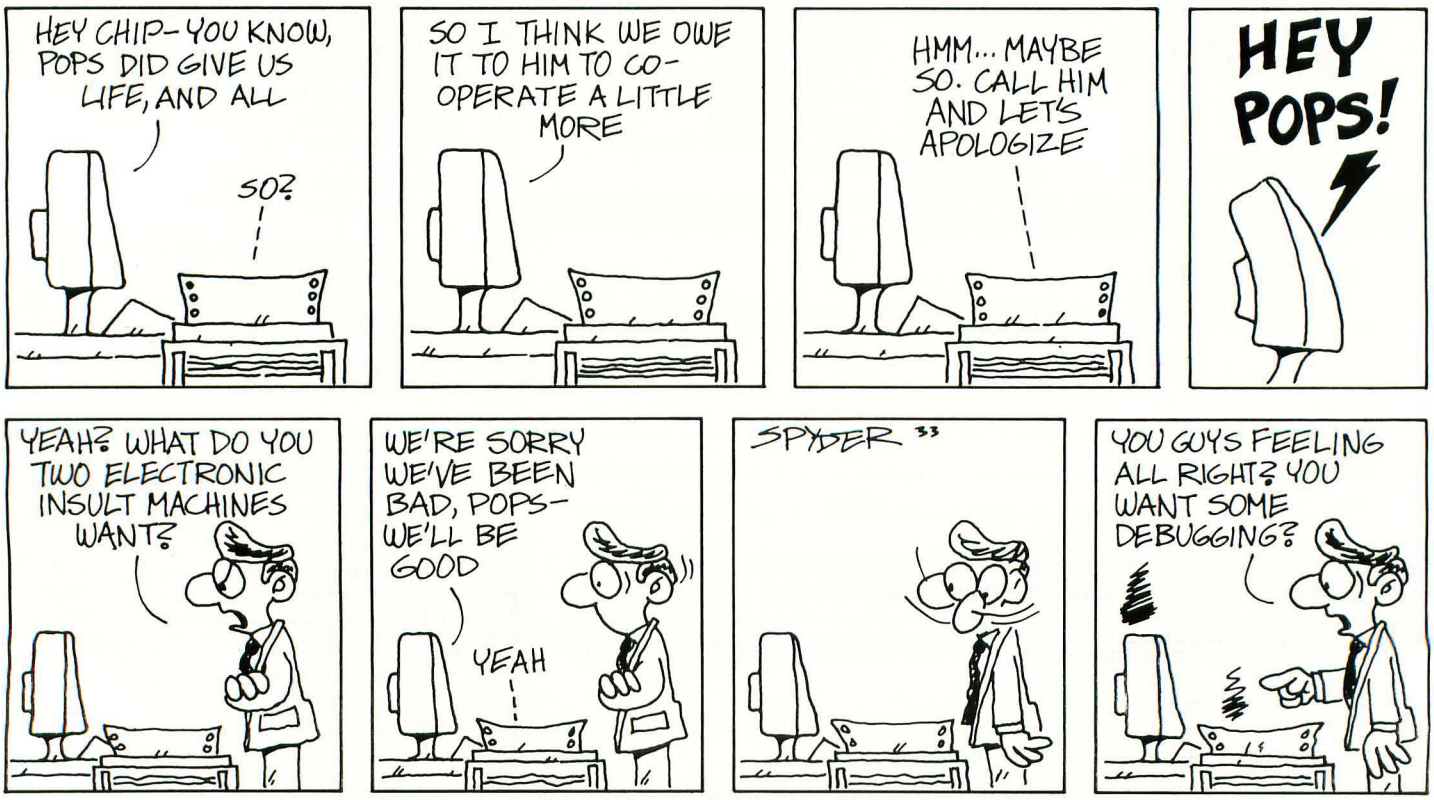
Pokey Player — You'll be able to make your ATARI® sing in three voices **and** add percussion with this amazing program.

Orchestra 90 — Professional music from your TRS-80®? Yes, it's possible with this little add-on. We'll tell you about it and give some tips on making your computer sound like the Philharmonic.

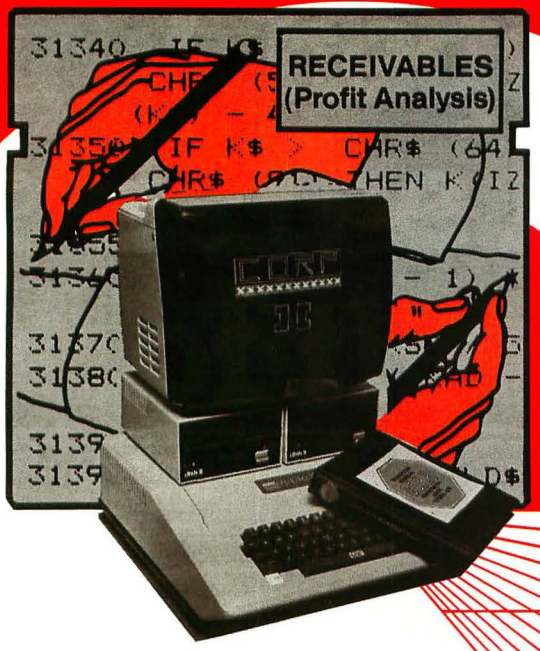
The IBM® PC — The Big Blue joins the **SoftSide** family and we'll have a miniature synthesizer to delight your ears.

PLUS — The Bach "Little Fugue" for the Apple™, The Rottberg Synthesizer on ATARI® DV, another Apple™ Diskourse, a user's Computer Nightmare to make you chuckle, reviews, articles, and a whole lot more.

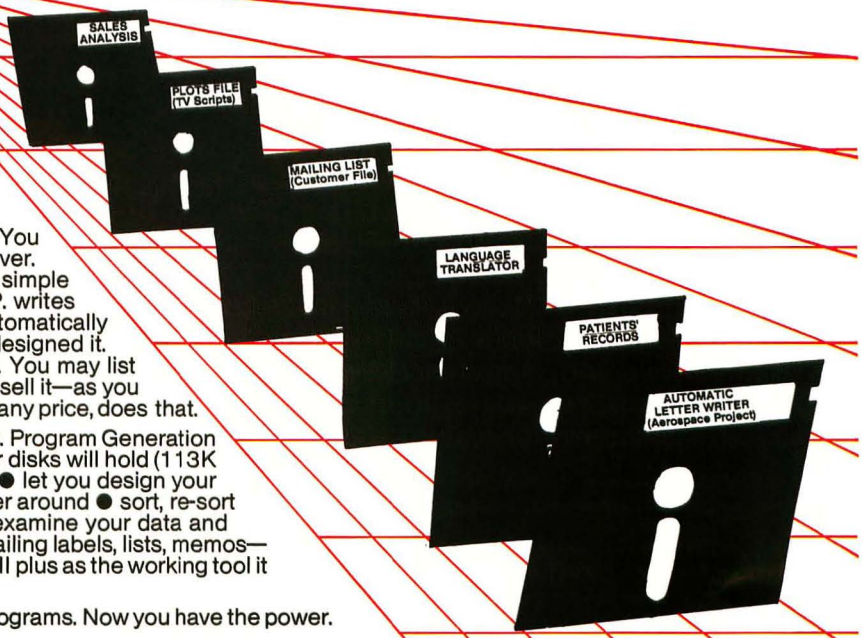
MACHINE HEAD BY SPYDER



your power is ready...
software that writes software
for Apple II™



NEEDED: Apple II Plus with Auto Start, 2 drives, Applesoft in Rom, 48K.



C.O.R.P.™ writes software. Perhaps you should read that again. Not "simplifies programming" or "makes debugging easier." C.O.R.P. writes APPLESOFT BASIC—complete, stand-alone programs which run, bug-free, the very first time. You need not type a single character of basic code—ever. C.O.R.P. does that. Your answers to questions in simple English "design" the program. In minutes, C.O.R.P. writes all the program code, scrolls it to the screen and automatically saves it to your disk. Your program because you designed it. Once written, your program runs without C.O.R.P. You may list your program, examine it, modify it further or even sell it—as you wish, royalty free. No mere "data base manager," at any price, does that.

The applications are almost limitless. Your C.O.R.P. Program Generation System will: ● handle nearly as much data as your disks will hold (113K per disk!) ● find any record in less than a second ● let you design your own screens as quickly as you can move the curser around ● sort, re-sort and update your data for you ● examine and re-examine your data and then print reports, checks, invoices, statements, mailing labels, lists, memos—you name it. C.O.R.P. allows you to use your Apple II plus as the working tool it really is. Isn't that why you bought one?

Computers should write code. You sould design programs. Now you have the power.

C.O.R.P. I features: ● a data entry program generator ● a full sort sub-system ● an update sub-system ● a full diagnostic package ● a print program generator $235

C.O.R.P. II features: ● all of C.O.R.P. I plus: ● a system menu generator ● a complete forms letter ● a full files editor ● a system demo package ● an Applesoft tutorial ..... $425.

CORTECHS CORPORATION

900 North Franklin, Chicago, Illinois 60610

See your dealer today or call toll-free 1-800-621-4109

(In Illinois, call (312) 943-0700)

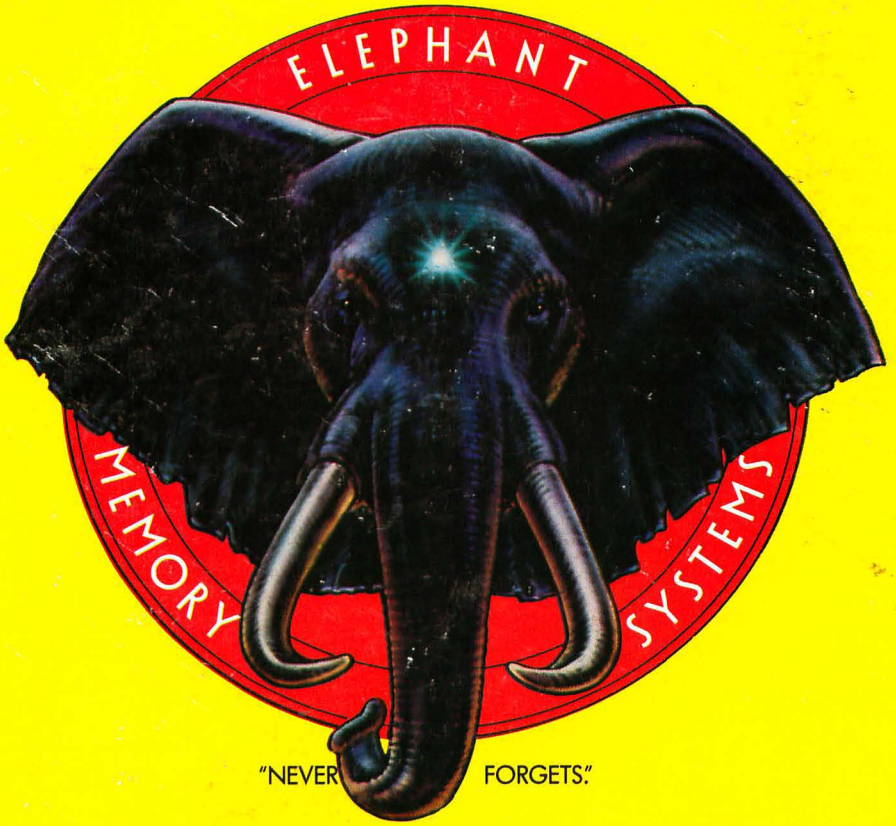
Dealer inquiries welcome.

C.O.R.P.™

C.O.D. welcome.

C.O.R.P. is a trademark of Maromaty Scotto Software Corporation.

REMEMBER:



MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to make life miserable for everyone in the disk-making business.

How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

ELEPHANT.™ HEAVY DUTY DISKS.

For a free poster-size portrait of our powerful pachyderm, please write us.
Distributed Exclusively by **Leading Edge Products, Inc.**, 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.