SoftSide™

TWO DOLLARS

VOLUME TWO • NUMBER ELEVEN • AUGUST 1980

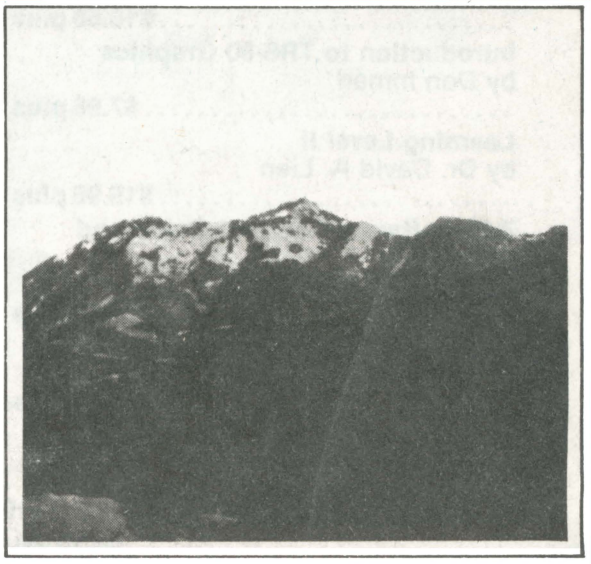
This month: Fly a SAILPLANE and soar to new heights!

Race with the wind - CARIBBEAN CRUISING - our SAILBOAT feature.

Armchair GOLF can be challenging. No golf bags to carry!

Be a professional race car driver and try the GRAND PRIX!

Also - Sound for your computer, games to challenge your mind, graphics features of the most popular computers, and programming tips from the pros!



The Best Books for your computer!

From Harvard Pennington, Scelbi Publications, SoftSide Publications, Dr. David Lien and many more ! . . .

Sargon Handbook
by Dan and Kathe Spracklen
..... **$15.95 plus $1**

The BASIC Handbook
by Dr. David A. Lien
..... **$14.95 plus $1**

Z-80 Instruction Handbook
by Scelbi Publications
..... **$4.95 plus $1**

The Little Book of BASIC Style
by John Nevison
..... **$5.95 plus $1**

TRS-80 Assembly Language Programming
by William Barden, Jr.
..... **$3.95 plus $1**

APL - An Interactive Approach
by Gilman and Rose
..... **$15.50 plus $3**

Introduction to TRS-80 Graphics
by Don Inman
..... **$7.95 plus $1**

Learning Level II
by Dr. David A. Lien
..... **$15.95 plus $1**

Z-80 Software Gourmet Guide and Cookbook
from Scelbi Publications
..... **$14.95 plus $1**

TRS-80 Disk and Other Mysteries
by Harvard Pennington
..... **$22.95 plus $1**

TRS-80 Interfacing
by Jonathan Titus
..... **$8.95 plus $1**

Secret Guide to Computers
by Russ Walters
..... **$5.95 plus $1**

6502 Gourmet Guide and Cookbook
from Scelbi
..... **$10.95 plus $1**

Introduction to Low Resolution Graphics
from Scelbi Publications
..... **$7.95 plus $1**

Z-80 and 8080 Assembly Language Programming
by Kathe Spracklen
..... **$7.95 plus $1**

Pathways through the ROM
from SoftSide Publications
..... **$15.95 plus $1**

Personal Information Management System (PIMS)
..... **$9.95 plus $1**

Calculating in Basic
..... **$7.95 plus $1**

Understanding Microcomputers
..... **$9.95 plus $1**

Take my computer . . . Please
..... **$5.95 plus $1**

Learn Micro Computers
..... **$14.95 plus $1**

Microcomputer Potpourri
..... **$2.95 plus $1**

To order Toll-Free **1-800-258-1790**
(In NH call 673-5144)

TABLE OF CONTENTS

SOFTSIDE • VOLUME TWO • NUMBER ELEVEN • AUGUST 1980

ARTICLES

PROGRAMS

HEAVY STUFF — (BEGINNERS BEWARE!)

DEPARTMENTS

STAFF

Lester Anderson
Donna Bennett

Brian Berkebile
Diana Bishop:
Subscriptions
George Blank
Atari submissions
Richard Bouchard
Joseph Breton
Suzanne Breton
Philip Brown
Clif Campbell

Nancy Chase
Elaine Cheever

Brenda Cookinham
Donna Cookinham
Pam Demmons
Mary Edwards:
Software information
Linda Fedas
Phoebe Fensterman
Karen Fissette
James Garon:
S-80 submissions
William F. Gollan:
Advertising Director
Sheila Jahnle:

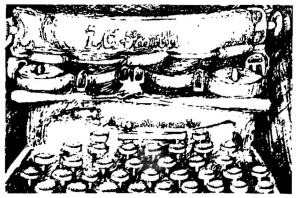
Donna Jean
Bette Keenan:
Customer service
Bea Kimball:
Shipping
Rick Koch
Kathy Maloof
Thomas Marshall

Donna McMahon
Lance Micklus
Doris Miller
Lauri Miller:

Glen Ohlund
Mark Ohlund
Mark Pelczarski:
Apple submissions
Mary Reed
Advertising
David Robitaille
Elizabeth Robitaille
Roger Robitaille Sr
Publisher
Eric Skyten
Richard Taylor
Jack Torres:
Hardware Information
Anmar Williams
Gary Young

SoftSide is published each month by SoftSide Publications, 6 South Street, Milford, New Hampshire 03055. Telephone 603-673-5144. Subscription rates: USA Bulk Rate $18.00 per year. USA First Class, APO, FPO, Canada, Mexico, Overseas surface mail - $25.00 year. Overseas Mail $30.00 year. All remittance must be U.S. funds. Mail subscription inquiries to SoftSide Subscriptions, P.O. Box 68, Milford, New Hampshire, 03055.

**SoftSide Publications
6 South Street
Milford, New Hampshire 03055
Controlled Circulation Pending Milford, New Hampshire.**



EDITORIAL

by Mark Pelczarski

Surprise! If you haven't already noticed, what was a tiny little magazine for your TRS-80 or Apple, depending upon your previous vantage point, is now a real size magazine for the TRS-80, Apple, Atari, and more! "What does this mean?" you might ask. A good question, and quite worthy of an editorial.

What will we offer? Why change from a magazine that was entirely for one machine, thus always applicable to YOUR computer, to one that will hold a mixture of items for more than one? The answer to the second question lies in the first. We will still give you the same kind of software we've always given you for YOUR computer, but now we will also show you what people are doing with other machines. We've found that most of the ideas we receive are not necessarily machine-specific, so why not share them with all? Furthermore, along with our format change, we have resolved that we will offer the best in documentation for ALL programs, telling you exactly how they work, defining the variables, and helping you understand them as you type. This means that not only will the programs for your computer be documented in a somewhat tutorial form; programs for other computers will be listed in this manner also. Although the languages vary, actual programming techniques are seldom much different; so you'll learn from the other programs, even though they may not fit directly into your computer. Eventually you may even want to try converting a program from another computer to your own, and we'll provide as much information as we can about the differences in language to help you. In the meantime, you'll still have the best programs we can bring you specifically for your machine.

Another advantage you may find with the new format is that it will inspire some cross-fertilization. A case in point is with the Mod-Prog article this month. When George, James and I put our heads together, we can occasionally come up with something at the fringes of being useful. When James wondered if Thomas Hanlin's programming hint for the TRS-80 would work with any other computers, we tried it, and lo and behold, it did! (For the Apple, but not the Atari -- perhaps explained by the fact that Microsoft wrote both Applesoft and TRS-80 Level II BASIC). We expect that this type of occurrence will not only be happening in our beautiful downtown Milford offices. We also hope that a lot of people out there will see articles meant specifically for one computer and say "Why not with mine?" -- and make great new advances that would never have occurred if they hadn't been seen done on other computers. Of course, we'd like to hear about this type of event so we can share more with our readers.

You will also notice that we've expanded our general offerings. In addition to regular features on graphics, sound, BASIC in general, and program reviews, we will feature regular columns about adventure programs and simulation games. In the works for future issues, we have several other general features that we hope you'll like.

Concerning direction, we promise to keep our content understandable. We believe that computers are not difficult to use unless they are made so. We don't want to make any pretensions that the use of computers is, or should be, an elitist activity. We will gear our content toward "everyman" (everyperson?), yet there will always be underlying information available if you have the desire to pursue a topic. If you want programs that you can key into your computer, if you want to learn more about how your computer works, or if you want to learn the details of the art of programming, SoftSide will be for you.

New products from . . .

● RITAM ● ON-LINE-SYSTEMS ● SYNERGISTICS ● SENSATIONAL SOFTWARE

RITAM

MONTY PLAYS MONOPOLY

"Monty plays Monopoly" is a computer opponent program designed to be used along with your Parker Brothers Monopoly game. You will need the board and all of the equipment that comes with the game to use this computer opponent program.

Monty provides a new dimension in microcomputer software. You will come to know him as simply another player, a bright, entertaining guest who provides real excitement for many of your favorite board games.

SYNERGISTIC SOFTWARE

PROGRAM LINE EDITOR

Stronger than Integer!!! More powerful than Applesoft!!! Faster than a speeding cursor!!! Able to leap the longest statements in a single move!!!

The Program Line Editor by Neil Konzen is the most sophisticated program development tool yet for the Apple II computer. Program Line Editor includes full feature line editors for both Integer and Applesoft programs that render all other techniques obsolete. Program development and modification can now be completed in a tenth the time previously required. The package includes a set of versatile user definable escape functions.

HIGHER TEXT

Higher text by Darrell & Ron Aldrich provides the display capabilities you need when the Apple's text output just isn't sufficient. With Higher Text, you can print characters in upper case, lower case, upside down, sideways and in any color. If standard characters still aren't adequate try printing in script, Old English, foreign languages, or special scientific symbology. Define your own specialized or decorative character sets, and easily create professionally customized displays without any hardware modifications.

ON-LINE-SYSTEMS

HI-RES ADVENTURE MYSTERY HOUSE

Through the use of over a hundred Hi-Res pictures you play and see your adventure! You communicate with HI-RES ADVENTURE in plain English. (It understands over 300 words!) All rooms of this spooky old house appear in full Hi-Res Graphics complete with objects you can get, carry, throw, drop, or ???

In this particular HI-RES ADVENTURE game, you are transported to the front yard of a large, old Victorian house. When you enter the house you are pulled into the mystery, murder, and intrigue and can not leave until you solve the puzzles. Your friends are being murdered one-by-one. You must find out why, and who the killer is. Be careful, because the killer may find you! As you explore the house there are puzzles to be solved and hazards to overcome. The secret passage way may lead you to the answer.

SENSATIONAL SOFTWARE

AIR TRAFFIC CONTROLLER

Air Traffic Controller retains the basic realism of air traffic control. This program requires the same steady nerves under pressure and the same instant, almost instinctive, analyses of complex emergencies which are demanded of a professional air traffic controller. But "ATC" adds the excitement and well-defined goals of a game. This is just a simulation, and all passengers left in air traffic limbo by a panicked player will live to fly another day.

TO ORDER CALL
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

OUTGOING MAIL

UNDER ATTACK!?

The warning whistle sounds shrilly. Every member of the SoftSide staff quickly grabs for the nearest support. Some hold on to a nearby computer, others reach for each other. The anticipated explosion shakes the entire building. Small pieces of ceiling-plaster drift slowly to the floor, twinkling in the rays of the afternoon sunlight. In a few moments, the reverberations subside. Bravely and calmly the intrepid men and women -whose job it is to bring you SoftSide each month - return to their vital duties (except for a few scattered couples whose lives have taken on a new meaning in the sudden desperate embrace of a fellow worker).

Alien invasion? Nuclear attack? Real-life Kriegspiel? What devastating forces have been unleashed against the innocent yet opulent Milford offices of SoftSide Publications?

UNDERGROUND RUMBLINGS

Fear not, loyal reader; this is NOT the END! This is just the beginning of a shiny new sewer system, urgently needed by the burgeoning town of Milford. It seems the old one was too small to continue to serve the needs of local residents, and so a new one is being blasted out of the ancient New England Bedrock.

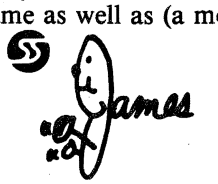
Perhaps it is fitting that the explosive changes taking place in Milford should coincide with the expansion of our own beloved SoftSide. With this issue, we take a bold step into the future. Many of you have asked for a bigger magazine, claiming that several of your favorite issues have slipped into the crack between the keyboard and the video monitor, never to be seen again. We have been moved by your plight! The past few weeks have been filled with frenzied activity here, as we reworked, retooled and redesigned the entire magazine.

The most obvious change, other than the larger size, is the inclusion of support for one or two "other" computers. Do not let this scare you. We plan to include just as much "pure TRS-80" material as ever - games, simulations, articles, Take A Parts and programming hints.

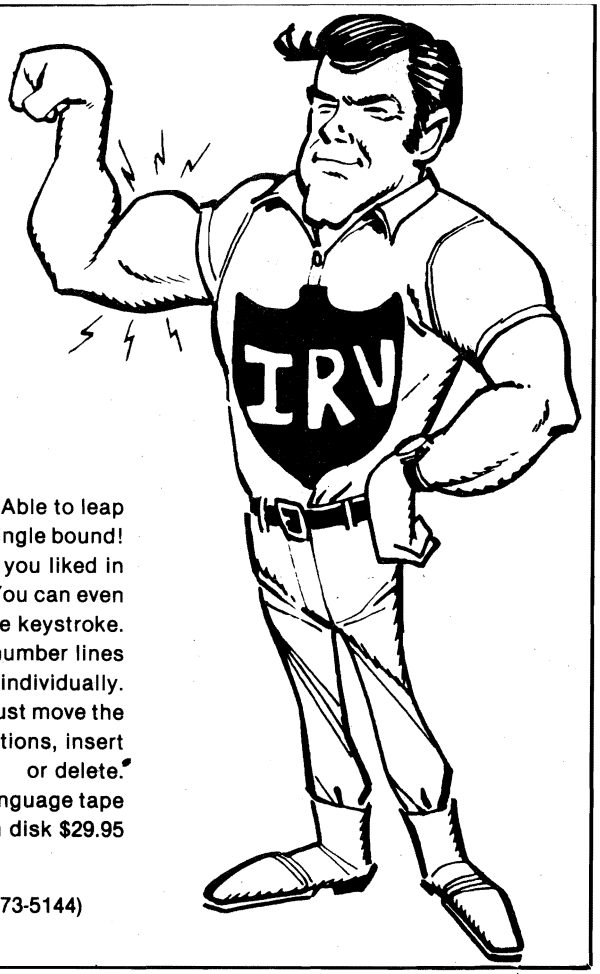
The added advantage to you is that when you have gotten all the TRS-80 programs up and RUNning -when you have wrung the last drop of inspiration from that final programming hint, there is still more! It never hurts to get "off your block" and see how the other third lives. We will be making an effort to document the programs in this and future issues in such a way that TRS-80 people can learn something from "those other computers" and vice versa.

Be honest now; aren't you just a LITTLE bit curious about the Atari's capabilities for four-part harmony - or the fact that the Atari has a grand total of sixteen different graphic modes? Did you know that the Apple has two graphic modes? One mode, called "low resolution" or just "lo- res", has fewer pixels than our own 127 by 48 grid! Just 40 by 40 relatively large boxes can be turned on or off. Of course, having 15 colors as well as black to choose from can make for some eye-catching displays, nevertheless. In "hi-res", the Apple boasts a resolution of 280 by 192.

Except for each computer's unique handling of its own special features (sounds, graphics, joysticks etc.) there is a remarkable similarity among the various BASICS. For those of you who use SoftSide as a learning tool, think of how proud you'll be when the day comes that you can take a program written, say, for the Apple, and get it running on your TRS-80! In fact, if you DO succeed in creating a GOOD translation of a SoftSide program from its original language into LEVEL II, by all means, send it in. If your version has appeal in its own right, you could be on your way to fame as well as (a modest) fortune!?



Put IRV on your programming staff!



IRV is faster than Level II BASIC! More powerful than a speed typist! Able to leap ten subroutines at a single bound!

Input shorthand -- one keystroke can enter a whole line! The feature you liked in Level III BASIC and T-Short, now with all keys user programmable. You can even turn your cassette player on or off with a single keystroke.

Relocate a line by simply editing the line number -- renumber lines individually.

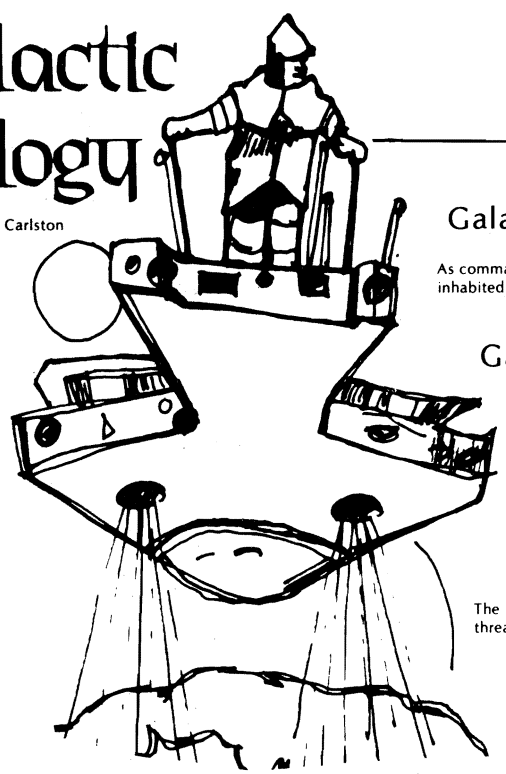
Video Editor allows cursor oriented editing of your programs. Just move the flashing cursor to the line to be edited and type the corrections, insert or delete.

IRV comes with manual and auto-repeat on every key. Machine language tape $24.95; on disk $29.95

To Order TOLL FREE **1-800-258-1790** (In NH call 673-5144)

Galactic Trilogy

by Douglas Carlston



Galactic Empire:

As commander of Galactica's Imperial Forces, you must conquer and hold the inhabited worlds of the galactic system. Deploy armies, raise taxes and conscript soldiers, gather intelligence, manage resources.

Galactic Trader:

You have succeeded in uniting the universe. Now you are an ex-soldier who must scramble for a living as an intergalactic tramp freighter. Outwit the locals at bartering, struggle with the fuel cartel, and outmaneuver the big trading monopolies as you seek your fortune.

Galactic Revolution:

The emperor is a bungler and is becoming unpopular. Your popularity is a threat, and he is seeking to kill you. Turnabout is fair play, so you start a revolution to unseat him. Will you succeed?

Each game: 16K Level II
Cassette: $14.95
Special - all 3 on disk 32K $39.95

TO ORDER TOLL FREE 1-800-258-1790
(In NH call 673-5144)

Garon's GOODIES

FIDDLING WITH VARPTR

Another in a continuing series of programming concepts for the S-80.

by James Garon

As is often the case with new discoveries, they seem to leap into the minds of several individuals at once. A famous example of this is the simultaneous invention of Calculus by both Sir Isaac Newton,in England, and Baron Gottfried Wilhelm von Leibnitz, in Germany.

In a similar fashion, the technique outlined here is based on a recent discovery made by several SoftSide authors. It involves the seldom-used and little-understood word, VARPTR. Beginning on page 8/8 of the Level II manual, we learn that when VARPTR in used with a string variable (as in:)

LET V = VARPTR(T$)

the value returned is a memory location (or a Function Call Error if the string has not been defined). Location V contains a number from 0 to 255. To see this, just type:

PRINT PEEK(V)

The number which is printed is the length of the string (T$ in this case). The next two locations (V + 1 and V + 2) contain the actual address of the first character in the string. You can have the computer reconstruct this address by having it

PRINT PEEK
(V + 1) + 256*PEEK
(V + 2).

The new discovery mentioned above involves the deliberate alteration of these three locations for our own nefarious purposes. For example, your program might create a graphic display with some random (hence unrepeatable) elements. It is possible, through the use of VARPTR, to save the picture in a string array. Later in the program, the picture can be recreated almost instantly, no matter how long it took to draw originally. It is even possible, once VARPTR has done its magic, to save the entire picture to disk or tape!

Let's assume that the picture is on the screen. The following routine will save the picture in the array T$:

```
1000 CLEAR2024:DIMT$(15):T$=""
1010 V=VARPTR(T$)
1020 POKE V,64'SET LENGTH TO 64
1030 FOR I=0 TO 15
1040   S=15360+64*I'  ACTUAL SCREEN ADDRESS
1050   H=INT(S/256):L=S-256*H'  CONVERT TO HI/LOW BYTES
1060   POKE V+1,L:POKEV+2,H'  MAKE T$ POINT TO SCREEN
1070   T$(I)=T$'  SAVE IN ARRAY
1080 NEXT
```

If the screen contains only text (no graphics), then it is now a simple matter to make a printed copy of the screen:

1500 FOR I = O TO 15:LPRINT T$ (I): NEXT

It is slightly more difficult to redraw a saved screen. This is due to the fact that anything PRINTed in the last position of the last line will cause the screen to scroll, thus messing up the picture. Here is one approach:

```
2000 CLS
2010 FORI=0TO14
2020   PRINT@I*64,T$(I);
2030 NEXT
2040 PRINT@960,LEFT$(T$(15),63);
2050 POKE16383,ASC(RIGHT$(T$(15),1))
2060 GOTO2060
```

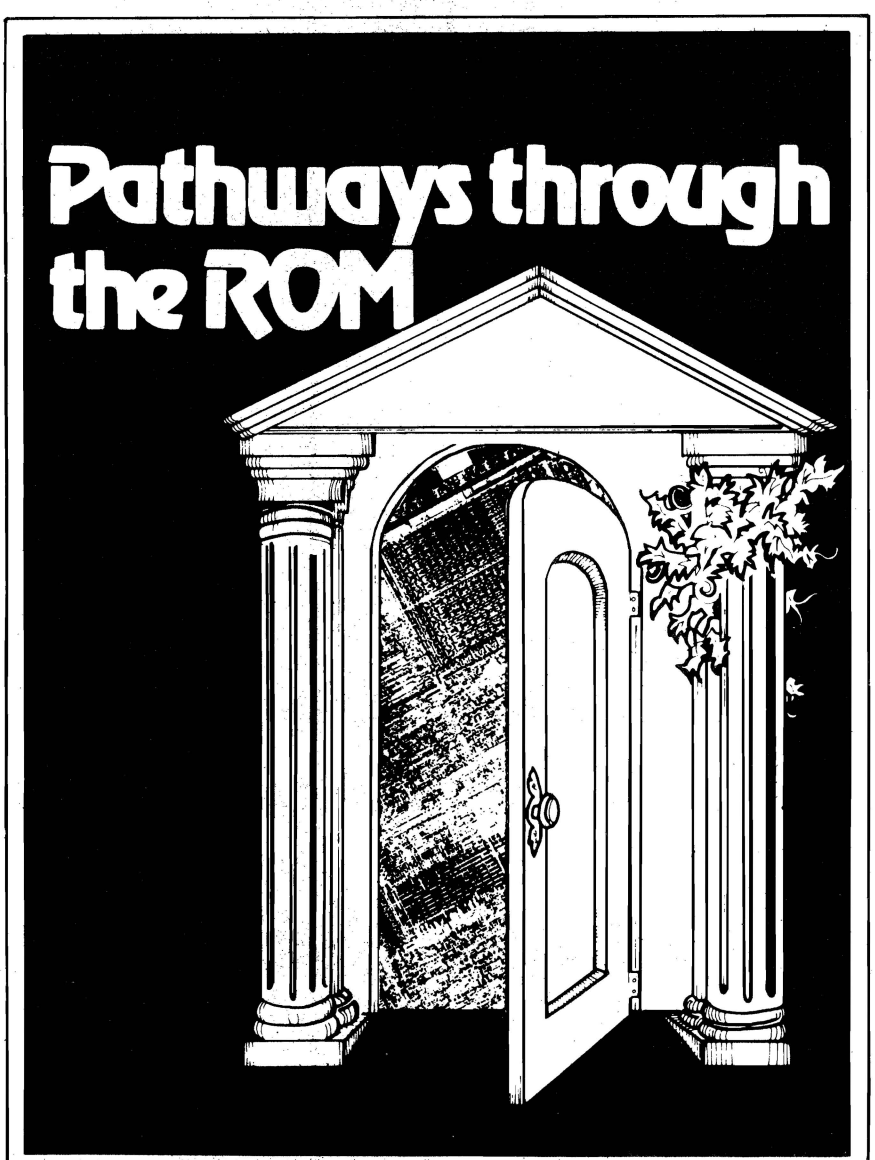

continued on page 80

● Unlock the Hidden Power Of Your Computer!

● Program fast & easy!

● Use ROM routines in your BASIC and Assembly Language programs!

All you need to know is in . . .



INCLUDES:

SUPERMAP
from Fuller Software
previously sold at $18.95

TRS-80 Disassembled Handbook
by Robert Richardson
previously sold at $10.00

Hex Mem
by John Phillipp
monitor written in BASIC.

Z-80 Disassembler
by George Blank

TO ORDER CALL TOLL-FREE
1-800-258-1790
In NH call 673-5144

● Guide to Level II BASIC and DOS Source Code $19.95 plus $1 shipping

Description of the contents of the Level II BASIC ROM by memory location, by function, and in lesson format. Includes several BASIC and Assembly Language programs in listing format to examine and use ROM routines.

A SoftSide Publication

INPUT

A Column of Your Letters

Dear Sirs:

I just had to write to let you know how much I enjoyed the "Galaxy Combat II" program in this month's issue. The graphics are just great and I really get quite a kick out of watching the Base planet revolve. Mr. Case is quite imaginative. Hope he keeps up the great work.

While playing, we (the neighborhood kids and I) were confused as to how many pieces of the planet killer weapon we had. So I added a gauge under the number of enemy count that tallies the weapon pieces. Keeps us from scratching our heads wondering if we have gotten close to the number of pieces needed.

Now, for an onion! In the May issue in the preceeding documentation for "... Tholian Sector," page 73, someone labeled the diagram of the Enterprise wrong. The impulse engines, as every Trekkie knows, are located on the rear of the saucer section - not at the ends of the warp engines. The two systems were purposefully separated so that the saucer section could be used as a lifeboat in an emergency. The saucer can jettison the secondary hull and can accommodate the entire crew as a lifeboat. Conversely, the secondary hull may be used as a lifeboat should the saucer be damaged. Each part is fully capable of independent operation.

Well, keep up the good work. I tell every TRS-80 owner I meet about SoftSide; I've even memorized your address. Can I get a commission?

Sincerely,
Sherry M. Taylor

Thanks for the informative letter, Sherry. You sure know your Enterprise! Yes, we plan to feature more of Phil Case's excellent work, possibly in our very next issue. As for that commission - your software certificate is already on its way for your helpful programming hint (originally included in your letter, but now in our Official Programming Hints section - right next to some hints from Phil Case himself!) ..... Ed

continued on page 80

PROGRAMS FOR YOUR COMPUTER

KINGDOM (CX4102)

This educational program cassette simulates the running of a small agricultural kingdom. Catastrophe may befall your people, your food supply or your crops. You determine the farming strategy and if you are successful you are allowed to reign another term. Teaches and develops mathematical skills.

BASKETBALL (CXL4404)

You can dribble, shoot, pass, block shots, and even make steals. Five different game variations allow one, two, three or four players to compete. Play against the computer, your family or friends. This game uses up to four Joystick Controllers. Cartridge.

BLACKJACK (CX4105)

Play one of the most popular casino games at home with your ATARI Personal Computer System. The computer is the dealer and it plays by the rules. You, on the other hand, bet money and decide whether to stand, draw another card or double down in your efforts to hit 21. One cassette.

COMPUTER CHESS CXL4009)

You can play chess against the computer. Eight levels of play are available. Moves are made using a Joystick Controller. The chess board and pieces are shown on the screen. Both "castling" and "en passant" moves can be made by you or the computer. The board can be set up for a particular chess problem or situation. The computer can play either white or black. Uses one Joystick Controller. Cartridge.

STAR RAIDERS (CXL4011)

A state of war exists between the Atarian Federation and the Zylon Empire. Your mission: Destroy all Zylon star ships. This strategy and action cartridge features four mission skill levels and rates a player at the end of each session. Dramatic sounds and screen displays make this the most exciting space game available today. Your battleground is the entire galaxy. Meteors fly through space; Zylon fleets surround and destroy friendly star bases, and docking at a starbase enables you to receive a new supply of energy and repair damaged equipment. For one player; uses a Joystick Controller.

SUPER BREAKOUT (CXL4006)

This cartridge is a sophisticated version of the popular electronic video game BREAKOUT™, in which you smash through layers of bricks and knock out a wall. Four different games are available, up to eight players can play, and a rating is displayed at the end of each game. You can optionally suspend play or request up to five additional serves. This cartridge uses up to four Paddle Controllers.

HANGMAN (CX4108)

This educational program cassette has 3 levels of play — Beginner, Intermediate, and Expert. You play against the computer by guessing the word the computer has selected. If you do not guess the word after six tries the computer will hang your man. You may use the computer keyboard or an optional Joystick Controller to guess the letter. Suitable for ages 8 to adult.

3-D TIC-TAC-TOE (CXL4010)

Now you can play Tic-Tac-Toe from a three-dimensional perspective. This game cartridge also contains Bottoms-Up, a variation of 3-D Tic-Tac-Toe. Match wits with the computer or compete against another player in either version. One-player games feature eight difficulty levels. Uses Joystick Controllers.

VIDEO EASEL (CXL4005)

You can create a design or pattern on the screen using the keyboard or up to four Joystick Controllers, then watch as the computer expands on the original pattern indefinitely. Many variations allow you to create imaginative and colorful graphic designs. Options include Drawing, Quad Drawing, Life (which uses the mathematic principles of John Conway's game of "LIFE"), resetting the color and six preprogrammed Painting designs. Uses Joystick Controllers. Cartridge.

AN INVITATION TO PROGRAMMING 1, 2, and 3

These lessons are designed to teach the fundamentals of programming in ATARI BASIC. A recorded voice asks you questions, waits for you to respond, and indicates whether you are correct or incorrect. These self-teaching courses will teach you to use BASIC programming statements as well as the color, graphics, sound and music capabilities of the ATARI Personal Computers. Each lesson requires 8K of RAM.

MUSIC COMPOSER™ (CXL4007)

Use this cartridge and the computer keyboard to compose songs, recreate old tunes, or experiment. When you program musical notes through the keyboard, you hear the melody as it is simultaneously displayed on the screen. You can change any note or the tempo. And you can arrange the music to play different sections in the order you choose. Once you have entered music, you can save it on the ATARI 410 Program Recorder or the ATARI 810 Disk Drive. You can also write BASIC programs that can convert a melody into melody plus harmony.

BIORHYTHM (CX4107)

Use this program cassette to calculate physical, emotional and intellectual highs, lows, and critical days. By giving the computer your birthdate you can plot a biorhythm chart on your television screen or optional ATARI Printer.

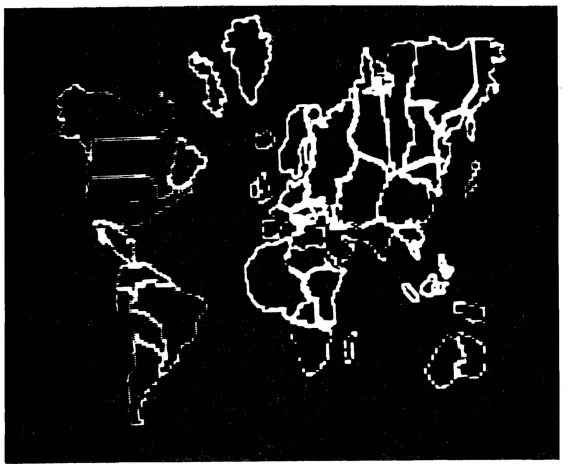
TO ORDER TOLL FREE

1-800-258-1790

(In NH call 673-5144)

Global War

Who will take complete control of the earth? In Global War, a highly detailed projective map of the earth's continent is divided into forty regions. The computer assigns each player a portion of the regions and starting armies. The players can allocate armies to specific regions or attack another player's territory.



The object of the game is to take control of the earth. This adult strategy game for two to nine players comes complete with instruction manual and rule book, and needs Applesoft ROM or Apple II Plus. Cassette (32K0 $17.95; Disk (48K) $24.95.

MUSE™

To order call TOLL FREE 1-800-258-1790 In NH call 673-5144

MONTY PLAYS MONOPOLY

A 16K COMPUTER OPPONENT PROGRAM FOR THE APPLE II AND TRS-80.

"MONTY TM plays Monopoly"* is a computer opponent program designed to be used along with your Parker Brothers Monopoly game. You will need the board and all of the equipment that comes with the game to use this computer opponent program.

MONTY TM provides a new dimension in microcomputer software. You will come to know him as simply another player, a bright, entertaining guest who provides real excitement for many of your favorite board games.

MONTY TM written in machine language, works with a 16K TRS-80 Level II or 16K Apple or Apple II plus. Cassette **$24.95.** Diskette **$27.95**.

Order TOLL FREE **1-800-258-1790** (In NH call 673-5144)

S-80 PROGRAMMING TIPS

To make the computer go into EDIT mode automatically with ALL errors (not just SN or SYNTAX errors), add these lines to your program:

```
10 ON ERROR GOTO 60000
60000 PRINT"THE ERROR IS IN LINE (no.)";ERL;"--CODE";ERR/2plus 1
60001 EDIT.
```

Sherry Taylor
Haines City, FL

Users of NEWDOS plus may have experienced problems when using LMOFFSET to save on disk a machine language program which normally loads into a DOS conflicting area. LMOFFSET creates a version that loads into high memory. It also adds an "appendage" which moves the entire program back into low memory once DOS is no longer required for the actual load. The difficulty occurs if the program is moved into the location where NEWDOS patches into the keyboard scan (for debounce and JKL features). Since two objects cannot occupy the same space, the result is chaos. Here's the solution: Disable the debounce routine by pressing and holding down the shift-up-arrow during initialization (power-up or reset). This will deactivate the debounce and re-enable the normal Level II keyboard routines.

Michael Stewart
College Point, N.Y.

LOWERCASE FOR SUPERZAP

As many more TRS-80 programs are being written in both upper and lower case, some users may prefer to be able to display both cases in the ASCII display columns of SUPERZAP.

The following DATA statement can be found at line 14150 in SUPERZAP 2.0 and at line 32100 in SUPERZAP 3.0:

```
nnnnn DATA 254,91,48,...,46
```

Change this line to read:

```
nnnnn DATA 254,123,48,...,46
```

This change will allow SUPERZAP to display lowercase ASCII characters instead of changing them to periods. As the display routine "pokes" directly to the screen, no lowercase driver is necessary with the R.S. lowercase mod. With the "normal" mods, you may either keep the lowercase switch in the "normal" position, except to view the lowercase, or you may have a driver active and keep the switch in the "lowercase" position. All other functions of SUPERZAP are unchanged.

Clay Schneider

continued on page 86

FOUR-OH-ONE-SIX

FOUR-OH-ONE-SIX- is an S-80 article which will be of most interest to Level II users - although most of the ideas presented will also work on disk systems.

The secret to "Load and Go" SYSTEM tapes.

One can redirect the RESET button and cause machine language cassette programs to jump directly to their starting point without having to type "/ENTER" after the tape is loaded. Both of these magical feats may be accomplished by changing the values stored in memory locations 4016H and 4017H.

To understand why these things can occur, one must first understand the function of the two memory locations mentioned above. These two locations contain the address of the keyboard driver. They will be referred to using this term for the rest of this article. When one is in BASIC, the keyboard is constantly being scanned. This keyboard scan is not contained within the Basic controller (ROM). ROM finds the scan routine by looking at the keyboard driver. These two memory locations hold the address of the scan routine. The ROM then JPs to the address contained in the keyboard driver. Fortunately for us mere programmers, the keyboard driver is not located in the ROM and is, therefore, accessible.

After a machine language program is loaded from tape, the prompt '*?) appears. During the time between when you see the prompt and when you enter your response, the keyboard is scanned via the address in the keyboard driver. If one uses the ORG statement in the EDITOR/ASSEMBLER to write into the keyboard driver an address other than that of the scan routine, ROM will attempt to scan the keyboard; however, it will JP to the changed address. For example, if you locate a machine language program at 4AOOH (18944 decimal), you could cause the program to be accessed immediately after the program was loaded from tape by putting the following subroutine in the program:

```
ORG   4016H   ;keyboard driver
DEFW  4A00H   ;the DEFW will flip
              ;so that it will load
              ;LSB first and MSB last
ORG   4A00H   ;the rest of the program
              ;starts here
```

To alter the location, change the second ORG statement and the DEFW statement. The DEFW statement writes OOH into 4016H and writes 4AH into 4017H. This causes ROM to JP to location 4A00H when it tries to scan the keyboard.

After going into your program after tape loading, if you leave the keyboard driver pointing to the start of your program, the program will restart whenever the RESET button is used (provided, of course, that there is no expansion interface). This is because the keyboard driver was changed so that whenever the RESET button is used, ROM jumps to your program.

In some cases, one may not want to restart the program when the RESET button is used. If one wants to save buffers that are cleared at the start of the program, one should again redirect the scanning routine by changing the keyboard driver. One might use the following routine to change the contents of the keyboard driver to JP to any part of the program when the RESET button is hit:

```
LD   HL,LABEL  ;LABEL can point to
               ;any location in your
               ;program
LD   (4016),HL ;change keyboard drvr.
```

By using this method, one can save buffers that would usually be cleared at the start of a program.

One can also write directly on the screen using an ORG statement not set to the start of the program. By using an ORG, the screen will be written on while the tape is loading, not afterwards. The following routine writes directly to the screen while the tape is loading:

```
ORG   3COOH   ;start of video memory
DEFM  'YOU MAY PUT ANYTHING HERE
      THAT YOU WANT.'
DEFM  'LEAVE BLANKS TO FILL UP THE LINE.'
```

One can have a very slow clear screen by using this method. If the messages will not all fit on the screen, one may load the screen, then the rest of the program, and,once again, load the screen with different messages.

We hope you find that these subroutines and ideas will enhance your programs of the future and also help you to better understand how your computer works. REMEMBER: A program with only one ORG goes into only one part of the vast computer memory!

by Robert W. Lord
and Rajeev Jayavant

HOW MUCH COMPUTER DO YOU NEED?

How much computer do you need?

About six months ago I received an irate letter addressed to me through SoftSide: Apple edition. The letter was from a businessman who had just purchased an Apple II computer. He complained that the salesman had told him that the computer would help him in his business, so he bought it. But when he went back to the business with his new "tool", he read the manual and found for the first time that he also needed programs to make it do anything useful. He thought he had purchased something like a cash register or adding machine, when in fact he had merely the beginning of a computer system.

My experience with hardware salesmen has led me to believe that many consider price a major obstacle in selling a computer system. For that reason, they often tend to sell a minimal system without explaining that more purchases will be needed before the customer will be satisfied.

A few years ago, when I became a computer hobbyist, memory cost about $100 per K, computers came in kits and software was something you wrote yourself. The sayings at that time were that any working computer was by definition extensively modified, and that a computer hobbyist put a down payment on a computer and kept buying memory until he went bankrupt. Things have changed a lot since then.

Today you can buy a good quality complete computer system for some purposes for less than $1000, take it home, plug it in, and use it with little trouble. The kicker in the previous sentence is the words "for some purposes". The real question concerning a personal computer is what you want to do with it.

The chief uses of computers in the home seem to be arcade games, simulation games, word processing, computer education, general education, hobby programming, support of another hobby or job, and providing a secondary source of income.

If your interests are limited to arcade games, the Atari Video Computer System, Atari 400, or Mattel's Intellivision are probably enough to meet your needs. Most of the general purpose home computers will amount to overkill for this purpose. The Atari 400 also has enough power to allow a minimal amount of computer education, general education, and support of some hobbies or jobs.

Beyond that there are really three different types of home computer systems using the same computers as the core of the system. These three types are cassette and disk based general purpose systems, and systems with accessories for special applications.

The first level is the cassette based general purpose system. This includes the Atari 800, the Radio Shack TRS-80 with Level II BASIC, and the Apple II. None of the three systems is really complete in its minimum configuration. With the Atari, you really need at least 16K of memory and a television set, with the TRS-80 you need at least 16K of memory, and with the Apple you need a cassette recorder, video modulator, and television set. (Note: Atari is shifting over to supplying 16K of memory, but is taking the cassette recorder out of the package.) But once you have 16K of memory, a good tape recorder, and a display, all three of these computers are good general purpose systems. They will support any of the above mentioned purposes except word processing, special applications, and providing extra income. The majority of home computer users will find their basic computer needs met with this type of system and the necessary programs.

The next step up is to move to diskette storage. Since the disk operating system takes memory space, this also means adding at least another 16K of memory. In fact, 32K of memory is usually enough for the TRS-80, with its limited graphic ability and dedicated screen memory. However, since many programs written for disk systems tend to be more complex and sophisticated than programs for cassette systems, more memory is often needed in the Apple and Atari. Many of good Apple disk programs assume a complete 48K of memory, and since memory for the Apple is relatively cheap, any Apple user who gets a disk drive and a controller card will probably want 48K of total memory. With the Apple you will also want to be concerned about which version of Basic you have: Integer or Applesoft. Integer Basic has been around longer and has had more software available until recently. Most of the programs are now being written in Applesoft, and we recommend starting with the Apple II Plus, which has Applesoft Basic at the same price as the Apple II. Either way, you can buy a language card if you want the other language. In the Atari 800, the disk operating system takes up 10K of memory and high resolution graphics can take another 8K, leaving very little in a 32K machine. At the same time, the BASIC cartridge or any other cartridge preempts the last 8K of memory, so any memory over 40K is usually wasted. Therefore, the logical memory for an Atari 800 disk system is 40K. Unfortunately, the Atari bus structure has problems handling the last memory slot, and program reliability is significantly less with 40 or 48K than with 32K. I personally keep 32K in my machine at all times, with another 8K available for programs that require it.

A disk system adds some extra income potential to the features of a cassette based general purpose system. Without disk storage, a programmer is really not productive enough to pay for the computer easily, much less to generate income beyond that. Too much valuable time is wasted

continued on page 58

PRODUCT REVIEWS

Microsoft Compiler–
A Review (S–80)

by James Garon

A DREAM COME TRUE?

It has long been the dream of many TRS-80 programmers to escape the speed limitations of BASIC. BASIC is slow because it is an interpreted language. That means that each time BASIC comes to a line of your program, it spends most of its time looking up the words in that line. Only a small percentage of BASIC's attention actually goes to performing your instructions. The dreamers among us - the speed-seekers - have wished for a way to eliminate all this repetitive interpreting; a way to let the potentially fast Z80 chip spend ALL its time doing our bidding.

What we've been dreaming about is called a "compiler". A compiler is a special program which converts a BASIC program directly into machine language. The original BASIC program, called the "source" program, is compiled into a machine language "object program." The object program should theoretically perform exactly the same as the BASIC source program with one important exception: speed. Since the compiled object program is already in the computer's natural language (Z80 machine code), no extra time is needed for the task of converting words such as PRINT, FOR, GOTO, etc. The result should be a much faster program.

Is Microsoft's new compiler the answer to our dreams? I ran some tests to find out. The preliminary manual states that execution speed can normally be increased from 3 to 10 times by compiling. However, increases of up to 30 times are promised if one uses integer variables exclusively (either by placing a "%" after each variable or by using DEFINT). Since I am interested in graphics, I used the following test program to paint the screen white four ways:

```
3 CLS
5 PRINT"(1) SINGLE PRECISION":PRINT"(2) INTEGER PRECISION":PRINT"(3)
SINGLE PRECISION POKE":PRINT"(4) INTEGER PRECISION POKE";:INPUTP
6 ON P GOTO 10,100,1000,10000
7 GOTO 5
10 CLS:FOR Y=0 TO 47
20 FOR X=0 TO 127
30 SET(X,Y)
40 NEXT X
50 NEXT Y
60 PRINT@0,;
70 GOTO5
100 CLS:FOR Y%=0 TO 47
110 FOR X%=0 TO 127
120 SET(X%,Y%)
130 NEXT X%
140 NEXT Y%
150 GOTO60
1000 CLS:VIDEO=15360
1010 FINISH=16383
1020 FOR I=VIDEO TO FINISH
1030 POKE I,191
1040 NEXT I
1050 GOTO 60
10000 CLS:VIDEO%=15360
10010 FINISH%=16383
10020 FOR I%=VIDEO% TO FINISH%
10030 POKE I%,191
10040 NEXT I%
10050 GOTO 60
```

continued on page 82

APPLETHINGS

APPLE III UPDATE

By now, you should be able to see the Apple III at your dealers. As we said last month, fear not, it doesn't replace the Apple II. It is directed at the $4000 and up business market and comes packaged as The Information Analyst, a business system containing a Mail List Manager, Visicalc III, and Apple Business BASIC. It will also be available in the future as a word processing package.

There are few additions to last month's preview worth noting. Of the peripheral capabilities we mentioned, there are only four expansion slots available to the user. However, the Apple III already comes with an RS-232C compatible serial port, a printer interface for the Apple Silentype printer, a clock, audio output through a 6 bit digital to analog converter, video modulator, and the disk drive. The Languages already announced are Apple Business BASIC, Pascal, and FORTRAN, which will be available in the fall.

Of specific interest to us at Softside, the Apple III will be able to use Apple II software. In the Apple II emulator mode you will not be able to use the full 96K or add any of the Apple III's additional power; the emulator is limited to 48K and Apple II's features. This does mean, however, that Apple III owners will be able to take advantage of the multitude of software already developed on the Apple II, not having to wait like owners of most new systems for good software to be developed.

Applesoft vs. Integer BASIC

The period of time that it takes for significant software to be developed on a new system or in a new language seems to be at least a year. Atari owners will take heart in that statement because they are still in that waiting period. There are a few good items available, but software for them is still not overly abundant. (A Catch-22 in this cycle is that a system itself will seldom catch on well until the consumer sees that there is a quantity of good software available, but the software doesn't usually come until there are a lot of systems sold.) Anyway, such had been the case with Applesoft, which is in most ways a much better language than Integer BASIC. (Integer BASIC does offer a little more if you like programming in machine language, but you could also buy an assembler that would go with Applesoft.) I've seen people buy the Apple II with Integer BASIC rather than with Applesoft for the same price because the dealer told them that all the software was in Integer BASIC.

Such is not the case anymore. Integer BASIC had a good head start, but now that Applesoft has been around a while, we see that most of the new programs are either in Applesoft or machine language, which is usualy compatible with either version of the machine. Although we are still receiving returns from our June SoftSide: Apple survey (which has had a great response), our tally so far shows that over 94% of those responding have Applesoft, or both Applesoft and Integer BASIC. Although we have been running Applesoft and Integer BASIC programs in similar proportions, we will eventually try to phase out Integer in favor of Applesoft in this magazine. We are still looking for good programs in either language, but we expect that the current trend will show itself in our submissions as more and more people use Applesoft in their programming.

Other Things

Another interesting fact from our survey so far is that about 90 % of the returns showed systems with at least 48K of memory, and there were quite a few with the Pascal language system. The most often-mentioned program in the section that asked you to name your five favorites was Invaders. The generic name 'Adventures' ran second, although when named most mentioned Scott Adams' or Microsoft's, and Sargon was third choice. Other favorites were Star Trek, Visicalc, SubLogic's Flight Simulator, and Three Mile Island. Most mentions of Star Trek were for the Apple version. You'll notice that TSE has added Super Invasion, by far the best version of the arcade game, to its product line. Your voices do not go unnoticed. I might add that if you see an excellent piece of software that TSE does not carry, let us know. Sometimes products don't find their way to us, and we'd like to keep our readers informed of the best.

For those of you contemplating purchase of Apple's language system: Keep your eye on Microsoft's Z-80 Softcard. It looks like it might offer much more power, and the cost is a couple hundred dollars less. We'll review it as soon as possible.

BUGS, WORMS AND UNDESIRABLES

by Kay Pasa

There are only a couple of bugs to report this month, both minor problems with Apple programs.

In the high resolution drawing program in the June issue, add the line:

```
851 NS=1
```

The program was condensed from a larger version, and that line was inadvertently omitted.

In the Horse Race program from July, there is a minor technicality. The program will occasionally give you a one to one payoff, meaning that you get back the money you put down as a bet. This is not a very good way to make money, so make the following change and the lowest payoff will be 2 to 1:

```
130 FOR L=1 TO 5: S =
INT(29*RND(1)) + 2: R(L)=s:
NEXT
```

Also, in the Pork Barrel program in the same issue, line number 520 inexplicably came out of our printer as 544. Only the electrons know why.

The Microsoft Z-80 SoftCard. New Power for Your Apple II!

Plug the new Microsoft Z-80 SoftCard into your Apple II™ and start using all of the system and application software written for Z-80 based computers. Software that you could never use before on your Apple II.

The SoftCard actually contains a Z-80 processor and lets you switch between the Apple's 6502 and the Z-80 with simple commands, so you can use software written for either processor.

Starting with Two Software Standards. Versatile CP/M, the most widely used microcomputer operating system ever, is included on diskette in the SoftCard package, ready to run on your Apple II.

You get Microsoft's 5.0 BASIC, too, the most powerful version to-date of our famous BASIC interpreter.

PRINT USING, 16-digit precision, CALL, and CHAIN and COMMON are just some of the major BASIC features you'll add. Applesoft's graphics extensions are still included.

More Power Down the Line. You can get even more programming power and versatility by adding Microsoft's FORTRAN, COBOL, BASIC Compiler and Assembly Language Development System. All are available separately to run with the SoftCard system.

And the whole host of CP/M-based business, scientific and educational applications can be easily transferred to your Apple with SoftCard.

The Microsoft Z-80 SoftCard is compatible with most every Apple product from the Apple II to the Apple II Plus, Language Card and peripherals. Independent peripherals for the Apple are supported as well. The SoftCard package requires a system with 48K and a disk drive.

$349 for SoftCard, CP/M, Microsoft BASIC and complete documentation.

To Order TOLL FREE
1-800-258-1790
(In NH call 673-5144)

TM Apple II is a trademark of Apple Computer, Inc.
CP/M is a registered trademark of Digital Research.

MICROSOFT
CONSUMER PRODUCTS

ATARI doubles product line

Atari has just announced 34 new software packages, a light pen, thermal printer, 80 column dot matrix printer, a dual disk drive, an RS-232 serial and parallel interface, and a modem.

With the new Telelink I ™ ROM cartridge, interface module, and modem, owners of either Atari computer may access the Source or other time sharing services.

The Atari 815 ™ is a dual density, single-sided dual disk drive using standard 5 1/4" disks. Each diskette holds 160K of data, and up to four units can be chained to the computer for 1.3 megabytes of storage. That is four times the capacity of competing computers like Radio Shack's model I.

The new printers are the 822, 40 column, 37 character per second thermal printer and the 825, 80 column, 50 to 83 character per second dot matrix printer. The 822, apparently from Trendcom, weighs less than 6 pounds and prints upper and lower case letters at a density of 10 characters per inch. It requires thermal paper. The 825 is identical to the Centronics 737 and gives you a choice of proportional or monospaced characters in condensed, normal, or elongated form. The character set includes lower case with true descenders.

The Atari 850 ™ Interface Module includes four serial RS-

continued on page 57

ATARI EDITOR/ASSEMBLER

Atari 400/800 ROM cartridge
Reviewed by George Blank

The Atari Editor/Assembler, due to be released in the near future, is an excellent and very usable Assembly language development system.

My own experience is with a preliminary version, recorded on EPROM and documented with a photocopy of an early draft of the manual, so it is likely that some features will change before the final version is released.

The manual is a guide to the Atari computers and the particular features of the Editor/Assembler, not an introduction to assembly language programming. If you are not already familiar with 6502 assembly language, you will need some other resources. My personal bias is that the Heathkit ET-3400 microprocessor trainer and course is one of the best introductions to Assembly language. This course teaches 6800 assembly language, but the instruction sets are similar and programming methods are about the same in most assembly languages. However, for those who don't want to spend $300 to learn, and who are willing to work somewhat harder to understand programming, Rodnay Zaks "Programming the 6502", published by Sybex, is a good text.

The ROM cartridge actually contains two programs, EDIT and DEBUG. EDIT is used to write, edit, and assemble programs, and DEBUG is a monitor, tracer, disassembler, and mini-assembler. The cartridge interfaces directly with the operating system and disk operating system to avoid repeating code and to save memory.

While the assembler can be used in an Atari 400 or 800 with no other peripherals, the manual recommends an Atari 800 with 16K of memory, disk drive, and printer for maximum utility. The final version will support disk storage, but may or may not support cassette input/output.

Using the Assembler

Programs are entered in five fields, as follows: STATEMENT NUMBER/ LABEL/ OPERATION CODE/ OPERAND/ COMMENT. The requirements for spacing are not critical except that a label must be separated from the statement number by exactly one space, the operation code mnemonic must be at least two spaces from the statement number, and other fields require a separation of at least one

continued on page 57

The Apple™ Stand

TO ORDER
TOLL FREE (orders only)
1-800-258-1790
(In NH call 673-5144)

KEY

- **I -** Integer Basic
- **M -** Machine Language (Apple II or Apple II plus
- **A -** Applesoft
- **ROM -** Apple II Plus or Applesoft card only

Your Apple II* software market from The Software Exchange

ADVENTURELAND
Adventure International
Cassette/24K/M $14.95

ADVENTURE SAMPLER
Adventure International
Cassette/24K/M $5.95

AIR TRAFFIC CONTROLLER
Sensational Software
Cassette/16K/M $9.95

ANDROID NIM
Disk/24K/A ROM $17.95

APPEN I TEXT EDITOR
Muse
Cassette/16K/I $17.95

APPILOT EDU-DISK
Muse
Disk/48K/I $49.95

APPLE 21
Softape
Cassette/24K/I $15.95

APPLELISTNER
Softape
Cassette/16K/I $15.95

APPLETALKER
Softape
Cassette/16K/I $15.95

BASEBALL
Muse
Cassette/16K/A ROM $14.95

BEST OF MUSE
Disk/32K/I $39.95

BISMARCK
Disk 48K/A ROM $59.95

BRIDGE CHALLENGER
Personal Software
Cassette/16K/M $14.95

CCA DATA BASE MANAGER
(Works in conjunction with Visicalc)
Personal Software
Disk/32K/A ..... $99.50 + $2

COMPU-READ
Edu-ware
Disk/48K/A $24.95

COUNT ADVENTURE
Adventure International
Cassette/24K/M $14.95

DAN'S DISK UTILITIES
Periph. Unl.
Disk/16K/I $39.95

DATE STONES OF RYN
Tape 32K/A ROM/48K/A $14.95
Disk/48K/A ROM $19.95

DR. MEMORY
Muse
Disk/32K/I $49.95

DUNGEON CAMPAIGN
Synergistic
Cassette/16K/I $12.50
Disk/32K/I $14.95

ELECTRIC CRAYON
Muse
Cassette/16K/M $17.95

ENGINEERING MATH TAPE I
Hayden
Cassette/24K A/16K A ROM .. $14.95

ESCAPE
Muse
Cassette/16K/I $12.95

FASTGAMMON
Quality Software
Cassette/16K/M $19.95

FORM LETTER
Periph. Unl.
Disk/48K/A ROM $24.95

FORTE
Softape
Cassette/16K/I $19.95

GAME PLAYING TAPE 1
Hayden
Cassette/16K/A $9.95

GAME PLAYING TAPE 2
Hayden
Cassette/16K/A $9.95

GAME PLAYING TAPE 3
Hayden
Cassette/16K/A $9.95

GENERAL MATH TAPE 1
Hayden
Cassette/24K A/16K A ROM .. $14.95

GHOST TOWN ADVENTURE
Adventure International
Tape/24K/M $14.95

GLOBAL WAR
Muse
Tape/32K/A ROM $17.95
Disk/48K/A ROM $24.95

HIGHER GRAPHICS
Synergistic
Disk/32K/I $24.95

HIGHER TEXT
Synergistic Software
Disk/24K/M $35.00

HI-RES MYSTERY HOUSE
On Line Systems
Disk/48K/M $24.95

INTERLUDE
Disk/32K/I $17.95

INVASION OF ORION
Tape 32K/A ROM/48KA $19.95
Disk 48K/A ROM $24.95

MAGIC PAINT BRUSH
MP Software
Disk 32K/A ROM $29.95

MAILING LIST
Periph. Unl.
Disk/48K/M $39.95

MAILING LIST DATABASE
Synergistic
Disk/48K/A $34.50

MAZE GAME
Muse
Cassette/16K/I $12.95

MEMO CALENDAR
Periph. Unl.
Disk/32K/A ROM $29.95

MICRO INFORMATION SYSTEM
Muse
Disk/48K/M $99.95

MICROSOFT ADVENTURE
Microsoft
Disk 32K/M $29.95

MISSION IMPOSSIBLE ADVENTURE
Adventure International
Cassette/24K/M $14.95

MONTY PLAYS MONOPOLY
Ritam
Cassette/16K/M $24.95
Disk/32K/M $27.95

MORLOC'S TOWER
Tape/32K/A ROM/48K A $14.95
Disk/48K/A ROM $19.95

MUSIC BOX
Muse
Cassette/16K/M $12.95

MYSTERY FUN HOUSE ADVENTURE
Adventure International
Cassette/24K/M $14.95

PIRATES COVE ADVENTURE
Adventure International
Cassette/24K/M $14.95

PROGRAM LINE EDITOR
Synergistic Software
Disk/24K/M $40.00

PYRAMID OF DOOM ADVENTURE
Adventure International
Cassette/24K/M $14.95

RESCUE AT RIGEL
Tape/32K A ROM/48K A $19.95
Disk 48K/A ROM $24.95

SARGON
Hayden
Cassette/24K/M $19.95

SARGON II
Hayden
Cassette/24K/M $29.95
Disk/24/K/M $34.95

SIDE SHOWS
Muse
Cassette/16K/I $12.95

STARFLEET OF ORION
Tape 16K/I $19.95
Disk 32K/I $24.95

STRANGE ODYSSEY ADVENTURE
Adventure International
Cassette/24K/M $14.95

SUPER INVASION
Sensational Software
Tape 32K/M $19.95

SUPER-TEXT
Muse
Disk/48K/M $99.95

TANK WAR
Muse
Cassette/16K/M $12.95

TEMPLE OF APSHAI
Tape 32K/A ROM/48K/A $24.95
Disk 48K/A ROM $29.95

TEXT EDITOR
Periph Unl.
Disk/48K/A ROM $64.95

THREE-D
MP Software
Disk/48K/A ROM $29.95

THREE MILE ISLAND
Muse
Disk/48K/I $39.95

U-DRAW I
Muse
Cassette/16K/M $17.95

U-DRAW II
Muse
Disk/32K/I $39.95

UNCLE SAM'S JIGSAW
Muse
Cassette/32K/A ROM $12.95

VISICALC
Personal Software
Disk/32K/A ..... $150.00 + $2

VOODOO CASTLE ADVENTURE
Adventure International
Cassette/24K/M $14.95

WILDERNESS CAMPAIGN
Synergistic
Cassette/48K/I $14.95
Disk/48K/I $17.50

WILDERNESS & DUNGEON CAMPAIGNS
Disk/48K/I $29.95

WINDFALL
Edu-Ware
Tape/32K/A $14.95
Disk/32K/A $19.95

BOOKS

BASIC HANDBOOK
by David Lien ..... $14.95 + $1

6502 GOURMET GUIDE & COOK BOOK ..... $10.95 + $1

INTRODUCTION TO LOW RESOLUTIONS ..... $9.95 + $1

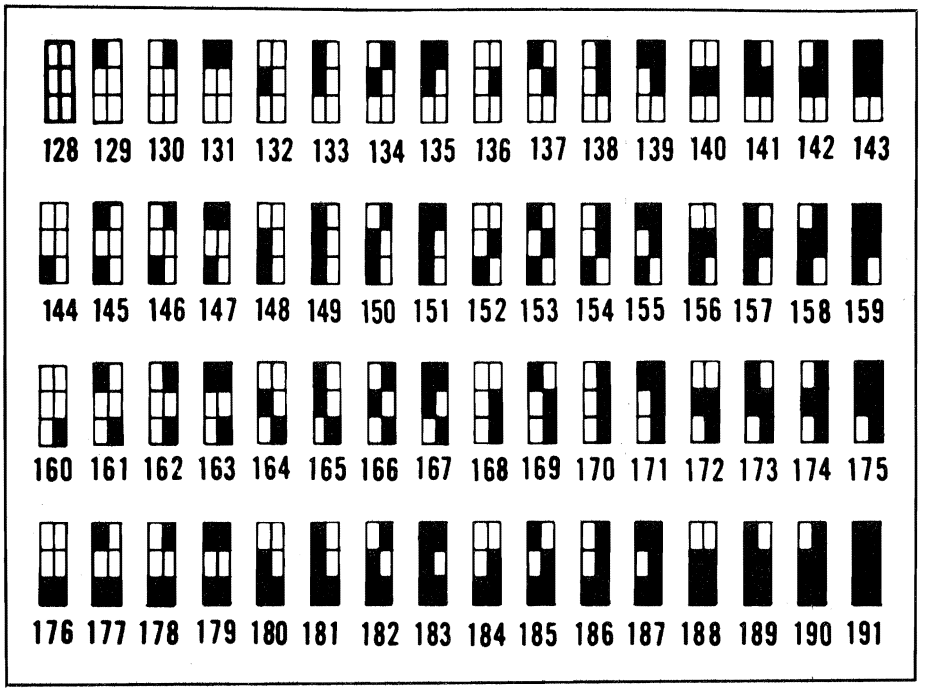
LITTLE BOOK OF BASIC STYLE
by John Nevison ..... $5.95 + $1

CONVERTING GRAPHICS

From One Computer To Another

A comparative look at the graphics commands of the S-80, Apple, and Atari computers.

by George Blank,
Mark Pelczarski
James Garon
and Rich Bouchard



One of the most confusing parts of converting programs from one computer to another is converting the graphics routines. At SoftSide, we seek to solve much of that problem with appropriate pictures, but there are a few techniques that can make life simpler. Part of the problem is the variety of graphics modes available.

The TRS-80 is fairly easy. It has only low resolution graphics with a 128 by 48 display. You can SET (turn on) or RESET (turn off) using a command of the form SET (a,b) where a is the horizontal position from 0 to 127 and b is the vertical position from 0 to 47. It is also possible to treat the TRS-80 screen as a collection of 1024 blocks of six squares, 2 wide and 3 high. You can print graphic characters or letters using the command PRINT @ x, where x is a number from 0 to 1027. There are 16 lines of 64 columns, with 0 to 63 at the top of the screen and 960 to 1023 at the bottom. A third way is to POKE these characters directly into video memory. Memory locations 15360 (3C00 Hex) to 16383 (3FFF).

The graphics characters range from CHR$(128) (space) to CHR$(191) (all six squares lit). (Refer to the chart above)

You can also include control characters such as CHR$(24) (Backspace cursor) and CHR$(26) (Downward linefeed). These characters can be placed on the screen individually or combined in a string. Yet another trick is to leave a string full of spaces in a program, locate that string with the VARPTR (variable pointer) function, and then POKE the graphics variables into the string. Radio Shack Level II Basic also includes a STRING$(a,b) for indicating several characters, where a is the number of characters and b is the particular character. For example, STRING$(20,140) would print a line 20 characters long with the two center blocks of each character turned on.

Here are some sample programs to put a small box in the center of the screen.

```
10 FOR X = 50 TO 71: SET(X,19) : SET(X,28) : NEXT X
20 FOR Y = 20 TO 27: SET(50,Y) : SET(51,Y) : SET (70,Y)
   SET (71,Y) : NEXT Y

10 PRINT@ 409, CHR$(188);STRING$(9,140);CHR$(188)
20 PRINTTAB(25) CHR$(191); TAB(35) CHR$(191)
30 PRINTTAB(25) CHR$(191); TAB(35) CHR$(191)
40 PRINTTAB(25) CHR$(143)+STRING$(9,140)+CHR$(143)

10 POKE 15769,188
20 FOR X = 15770 TO 15778
30  POKE X, 140 : POKE X + 192, 140
40 NEXT X
50 FOR X = 15833 TO 15897 STEP 64
60  POKE X,191 : POKE X + 10,191
70 NEXT X
80 POKE 15779,188 : POKE 15961,143 : POKE 15971,143

10 CLEAR 500 : N$=CHR$(26)+STRING$(11,24)
20 A$ = CHR$(188)+STRING$(9,140)+CHR$(188)+N$+CHR$(191)
   +"          "+CHR$(191)+N$+CHR$(191)+"          "+CHR$
   (191)+N$+CHR$(143)+STRING$(9,140)+CHR$(143)
30 PRINT@ 409, A$
```

Apple Graphics are somewhat more complex, with two graphics modes and colors. Low resolution has a 40 by 40 display in 16 colors and high resolution has a 280 by 160 display with 6 colors.

To enter low resolution graphics, you use the command GR and specify the position of the point to be plotted in a range of 0-39, both horizontally and vertically. Colors are chosen with the command COLOR =x, with the following 16 possibilities for x:

0 Black	4 Dark Green
1 Magenta	5 Light Grey
2 Dark Blue	6 Medium Blue
3 Purple	7 Light Blue
8 Brown	12 Green
9 Orange	13 Yellow
10 Grey	14 Aqua
11 Pink	15 White

The lo-res graphics commands are:

PLOT X, Y which sets the point X, Y to the most recent color.

HLIN X1, X2 AT Y which draws a horizontal line from (X1, Y) to (X2, Y).

VLIN Y1, Y2, AT X which draws a vertical line from (X, Y1) to (X, Y2).

SCRN (X, Y) returns the number of the current color at point X, Y.

To enter the high resolution mode, you use the command HGR and enter the point to be plotted in a range of 0 to 279 horizontally and 0 to 159 vertically. Colors are selected with the command HCOLOR = x with x selecting the following colors:

0 Black	4 Black
1 Green	5 Orange
2 Violet	6 Blue
3 White	7 White

The hi-res graphics commands are:

HPLOT X, Y which sets the point at X, Y to the current HCOLOR.

HPLOT X1, Y1 TO X2, Y2 which draws a colored line from point (X1, Y1) to (X2, Y2)

Both high resolution and low resolution graphics modes allow four lines of text at the bottom of the screen. It is possible to override the text window and extend the vertical resolution to 48 lines in lo-res and 192 lines in hi-res, but you lose all text, since text and graphics cannot be mixed. To eliminate the text window, POKE-16302, 0. To restore the window, POKE-16301, 0.

In addition, the Apple also has commands that allow you to use a prepared "shape table" of previously specified patterns. The shape commands are:

DRAW N AT X, Y which draws shape N at the given point on the screen.

XDRAW N AT X, Y which draws the same shape, but reverses the colors in those positions currently on the screen.

ROT=X sets the rotation for shapes to follow.

SCALE=X sets the scale for shapes to follow.

Creating shape tables is a complex process beyond the scope of the present discussion. There are programs such as the Magic Paintbrush graphics package which allow you to create shape tables by simply drawing on the screen with the game paddles.

These programs will draw our box on the screen of the Apple:

```
10 GR : COLOR = 9 : HLIN 15,25 AT 15
20 HLIN 15,25 AT 25
30 VLIN 15,25 AT 15
40 VLIN 15,25 AT 25
```

```
10 HGR : HCOLOR = 5 : HPLOT 15,15 TO 15,25
20 HPLOT 25,15 TO 25,25
30 HPLOT 15,15 TO 25,15
40 HPLOT 25,15 TO 15,25
```

```
10 HGR : HCOLOR = 5
20 HPLOT 15,15 TO 15,25 TO 25,25,TO 25,15 TO 15,15
```

The ATARI computer has considerably more complex graphics. There are basically 3 text and 6 graphics modes. Actually, there are three more graphics modes not easily accessible to the Basic programmer, and the three text modes can contain graphics. In addition, there are three variations on each graphics mode. Here is a table of the basic modes.

Mode	Type	Horizontal	Vertical (Full Screen)	Colors
0	Text	40	- 40	2
1	Text	20	20 24	5
2	Text	20	10 12	5
3	Graphics	40	20 24	4
4	Graphics	80	40 48	2
5	Graphics	80	40 48	4
6	Graphics	160	80 96	2
7	Graphics	160	80 96	4
8	Graphics	320	160 192	(2)

The difference between the first and second vertical column is a four line text window at the bottom of the screen. If you do not want the text window, add 16 to the graphics mode number to override the split screen format. Thus GRAPHICS 22 is a full screen version of GR.6 with a 160 by 96 display, 0-159 horizontal and 0-95 vertical. Actually, that overstates the size slightly, as the machine initializes with 39 characters across and you have to POKE in the 40 character wide screen. This feature is to allow for overscan on some TV sets. You can also narrow the screen with a similar POKE. The reason the 2 for GR.8 is in parentheses is that you really are only entitled to two shades of one color. Graphics 9, 10, 11, and a few special tricks can overcome that problem, but they are beyond the reach of the current discussion.

The ATARI has 16 colors available, each of them in 8 hues from dark to almost white, using the command SETCOLOR a, b, c with a representing one of 5 color registers, b representing the color from the table below, and c an even number from 0 (dark) to 14 (light). You can change colors on the screen without disturbing the drawing by simply changing the color register for that point.

0 Grey	4 Pink
1 Gold	5 Pink Purple
2 Orange	6 Purple
3 Red Orange	7 Blue
8 Blue	12 Green
9 Light Blue	13 Yellow Green
10 Turquoise	14 Orange Green
11 Green Blue	15 Light Orange

continued on page 88

ONE LINERS

In last month's SoftSide: S-80 edition in the programming hints column, James printed several unique graphic programs. What made them special is the fact that the programs were each only one line long. Being a rather fascinating idea, we thought it would be worthwhile to share the concept with the rest of our readers this month and see how creative you can be.

The rules are as follows:

1. The program must be written as a single line in BASIC.

2. The program must be self-contained. Do not make any assumptions about string storage available, graphics modes being in effect, etc.

3. The program should provide a continuously changing graphics display, as impressive as possible.

```
1 CLS:FORT=1TO2STEP0:A=PEEK(14400):E=SGN(((AAND24)-14)*(AAND24))
:F=SGN(((AAND96)-50)*(AAND96)):IFX+F<00RX+F>127ORY+E<00RY+E>47NE
XTELSEX=X+F:Y=Y+E:RESET(X,Y):SET(127-X,Y):SET(X,47-Y):SET(127-X,
47-Y):SET(X,Y):IFINKEY$<>CHR$(31)NEXTELSERUN
```

Here's a creative example for the S-80 by Rich Bouchard:

This program allows the user to create graphic displays. The program starts by turning on the four graphic points at the corners of the screen. The dot in the upper left will start to flash. You can move the dot with the arrow keys, and all four dots will move also, forming a symmetrical pattern. Diagonal movement is created by holding two keys down; for example, the right and down arrows. When your masterpiece is finished, it may be erased by hitting the CLEAR key.

And here's an example in Applesoft by a not-so-creative editor.

```
 LIST
1  HOME : GR : FOR J = 1 TO 2 STEP
     0: FOR K = 0 TO 1: FOR I = 0
      TO 39: COLOR=  RND (1) * 16
     : HLIN 0,39 AT I: COLOR=  RND
     (¶) * 16: VLIN 0,39 AT 39 *
     K - 2 * I * K + I: NEXT I: NEXT
     K: NEXT J
]
```

If you create an excellent one-liner, send it in and we might print it. Send them to:

One Liners, c/o SoftSide
PO Box 68,
Milford, NH 03055

MODPROG

MODPROG is an S-80 program for Level II 4K and up, and an Apple program for those with Applesoft.

by Thomas Hanlin III

Like to be able to change PRINTs to LPRINTs, use one-key abbreviations for statements, use Super Graphics, and quickly find every REMark in your program with one short BASIC program? Modprog will do all this and more in just three lines (Four for Apple users.)

How to use it: load your program into memory and type in Modprog. (The lowest line number in your program must be 4 or higher.) Modprog must be the first four lines in memory, so don't change the line numbers! RUN Modprog. To change one command to another throughout the program (PRINT to LPRINT, SET to RESET, etc.), look up the values of the commands in the chart and enter them when prompted. Modprog takes it from there. To find a character/statement (REM, etc.) without changing anything, just enter the appropriate code twice. Line numbers are listed automatically. To change "N" to "NEXT", etc., use the ASCII value of the letter (see the chart on page C2 of your LEVEL II BASIC USER'S MANUAL) as the number to change. Note that this letter will be changed throughout the program --including, if you're not careful, REMs, PRINTs, and variables! To get Super Graphics, use a character in the PRINT statement that isn't used anywhere else in the program. Search with the ASCII value of that character and change it to the appropriate graphics character code. When Modprog's done, DELETE0-3 and RUN, CSAVE or SAVE your program.

How it works: BASIC program lines are stored in memory starting at location PEEK (16548) + 256*PEEK(16549) in the format: X - Y - A - B - instructions - 0 - X . . . where X + Y*256 is the memory location of the two locations preceding the next line number and A + B*256 is the present line number. When X and Y both equal zero then there is no next line - you're at the end of the program. So . . . Modprog line 0 starts at the beginning of program memory, advances the line counter four lines so it doesn't try to modify itself, and INPUTs the character/statement to change and what to change it to. Line 1 finds the number of the line being looked at and the location of the two locations preceding the next line number (which is also the first location past the end of the line), and search-loops from the beginning to the end of the line. Line 2 continues that loop. Line 3 sets things to search the next program line and determines

continued on page 70

SAILPLANE DERBY

SAILPLANE is an S-80 program for Level II 16K and up.

by David J.T. Nunn

The Sailplane Derby program is a 'True to Life' simulation of a cross-country sailplane race.

Competition gliding is carried out by enthusiasts in most countries of the world using sailplanes which are typically of 50-foot wingspan, constructed of fiberglass reinforced plastic with extensive streamlining and costing upward of $30,000. They are towed aloft by small, powered airplanes and released: from then on the pilots must rely on rising upcurrents of warm air called 'thermals' to stay aloft. Pilots guide their machines across country seeking the summer cumulus clouds that often mark the top of an active thermal and then turn in tight circles to stay within the confines of the narrow columns of rising air, enabling their sailplanes to be lifted upward. When all possible height has been gained (sometimes by entering the cloud capping the thermal), they dart off in search of the next active thermal on the course line.

In competition, the task is usually to fly from the take-off point, around two turning points and back home. This triangular task is timed; the pilot completing the task in the shortest time is the winner.

SAILPLANE PERFORMANCE

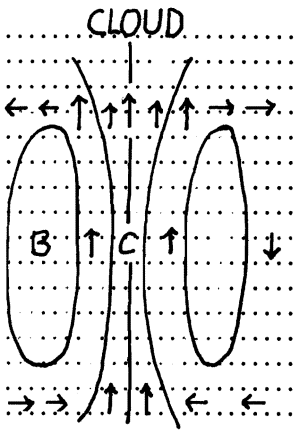
A sailplane's performance is usually specified as its maximum glide angle, typically 40 to 1 @ 50 mph: for every foot in height lost it glides forward 40 feet. However, if the sailplane flies faster, then the glide angle deteriorates due to increased aero-dynamic drag. (Flying slower than 50 mph worsens the glide angle.)

SPEED	40	50	60	70	80	90	100	110	120
G.ANGLE	23:1	40:1	34:1	30:1	26:1	24:1	21:1	19:1	18:1

The table above shows that flying at 50 mph will produce maximum glide distance. However, if the thermals are strong and you have sufficient altitude, then faster progress can be made at 60 to 80 mph without too large an altitude loss.

THERMAL STRUCTURE

The diagram shows a vertical cross section thru a typical thermal. The arrows indicate relative air movements. It will be seen that a sailplane entering the thermal at (A) will encounter strong sinking air; then at (B) neutral air and finally strong lifting air at (C). The shape of the thermal is sometimes called a 'vortex ring' — similar to the smoke rings produced by extrovert smokers!



CORRECT 'SPEED TO FLY'

When a sailplane flies through a sinking or rising air mass, its airspeed should be adjusted in accordance with the following general rule:

1. If the sailplane is in sinking air, its speed should be increased.
2. If in rising air, the speed should be decreased (usually to a minimum of 50 mph).

At first glance, this rule seems illogical. However, the reason is to ensure that, in the case of sink, the sailplane is subject to the sink effect for the minimum 'period of time', thus reducing the 'total' loss of height; and the converse for lifting air. It should be noted that the exact quantity of each speed adjustment can only be learned by experience, although mathematical treatments for the problem have been derived by several experts including Dr. Paul MacReady (lately of Man Powered Flight fame).

It may be necessary to increase the airspeed of the sailplane if you are attempting to 'penetrate' into a strong wind. Consider the table set out below:

SPEED	40	50	60	70	80	90	100	110	120
G.ANGLE	9:1	20:1	21:1	20:1	18:1	17:1	16:1	15:1	14:1

The table above shows the dramatic effect of glide angle (over the ground) of a sailplane heading directly into a 24 mph wind - the strongest the simulation is likely to inflict on you. You must remember that, although the angle through the air is still the same as shown in the first glide angle chart, the distance covered 'over the ground' is greatly reduced. You can also see that it pays to fly faster, say at 60 or 70 mph, to achieve maximum distance over the ground.

FLYING IN CLOUD

It is possible to climb much higher in a thermal by allowing your sailplane to be drawn up, while still circling, into the cumulus cloud above. When in cloud, your turn indicator and compass are invaluable in maintaining control and ensuring that the exit from the cloud is made on the correct heading.

It is important to note that both thermal strengths and cloud base heights rise during the day to a maximum in the early afternoon. So it is in the simulation; therefore, better progress may be made later in the day in the better weather conditions (see later section on launch times).

DESCRIPTION OF OPERATION

The simulation program runs as follows:

1. Either you or the computer sets the weather
2. Input number of players
3. Input players names
4. A meterological forecast will now be printed. You should particularly note the thermal strength and wind speed and direction as weak thermals coupled with a strong wind make progress into wind very difficult.
5. The players are now invited to either accept a launch half an hour after the first thermals have developed or request a deferment of the launch for one or two hours (enter 0,1, or 2 as required).

If either a long distance flight is anticipated or the period of time thermals will be produced is short, then acceptance of an immediate launch is advisable. If not, a faster flight can be achieved by deferring the launch and thereby using the strongest thermals only available during the middle of the day.

(Note: During the last two hours of the day the thermal strengths start to diminish before disappearing altogether.)

6. The program now requests coordinates of the two turnpoints around which the triangular task is set. You are launched at position X=20 and Y=20, and the location of the turnpoints are to be in the range X=10 to 70, Y=10 to 70.

It is suggested that as a beginners task the first turn be at X=50 and Y=50; the second turn to be at X=50 and Y = 30.

7. The program now produces the cockpit graphic display and requests the first player to commence the task.

YOUR SAILPLANE'S INSTRUMENTS

Your sailplane is equipped with the following instruments:

1. ASI (located top left): Stands for air speed indicator—your speed is limited to the range 40 to 120 mph.
2. ALT (located bottom left): Displays your altitude above ground level in feet.
3. COM (located top center): Compass gives standard 360° compass readout (360°=North, 270°=West, 180°=South, 90°=East).
4. V-V (located center): This is the most important instrument you have. It is the variometer, the instrument that shows how fast the glider is sinking or rising. An analog (needle type display) is provided. However, you will find that the digital display below is more accurate. A reading of +1 indicates a climb rate of 100 feet per minute and -3 indicates a descent rate of 300 feet per minute.
5. T&S (located to the right of the variometer): This is the turn indicator which indicates, by both analog and digital output, the rate of turn (in degrees).
6. Stopwatch (extreme right): This times your flight. The watch starts automatically at the beginning of the game. However, it will be reset if a relaunch (see section on commands) is executed. The watch is stopped immediately after a valid crossing of the finish line occurs.

CONTROL COMMANDS

Take special note that operation of any of the following control commands can only take place while the symbol (***) is seen immediately to the right of the compass. During the period that control commands are allowed, any combination of commands are allowed.

If your computer does not have a numeric keypad, it is suggested that the following alterations are made to the simulation program:

Change Line 2000 to read — 2000 IFA$="Q"THENS(B)=S(B)+10

Change Line 2010 to read — 2010 IFA$="Z"THENS(B)=S(B)-10

You have made a note of these alterations, haven't you!!

FLIGHT CONTROLS (FOR TRS-80 WITHOUT NUMERIC KEYPAD)

4 = Left turn - is incremented in 3 degree units for as long as key is depressed (up to a total of 90°)
6 = Right turn - (incremented as above)
5 = Neutralizes turn control to the 'straight ahead' position
Q = Each press increases the sailplane's speed by 10 mph
Z = Each press decreases the sailplane's speed by 10 mph

FLIGHT CONTROLS (FOR TRS-80 WITH NUMERIC KEYPAD)

4 = Left turn - is incremented in 3° units for as long as key is depressed (up to a total of 90°)
6 = Right turn - (incremented as above)
5 = Neutralizes turn control to the 'straight ahead' position
8 = Each press increases the sailplane's speed by 10 mph
2 = Each press decreases the sailplane's speed by 10 mph

ADDITIONAL COMMANDS

Ⓢ = START BUTTON — Once you have been launched the stopwatch timing your flight starts to run. If, however, you wish, for some tactical reason, to take another start, you can fly back to behind the start line (so that X and Y are both less than 20), and then, provided you are below the height of 3,281 feet, pressing the start button will reset your stopwatch to zero. Why should you want to go to this trouble! Well, the experts, always interested in saving minutes or even seconds, like to climb high in a thermal near to the start line and then dive across the line at 120 mph, pressing the start button at just the right moment. This allows them to decelerate, gaining height by converting speed into altitude, thereby saving the time taken to climb, say, 300 feet in the next thermal. That's the theory anyway. Oh! Why 3281 feet? Well, for the uninitiated, that's equal to 1000 meters, the height the FAI, the international aviation ruling body, determines is the maximum start height.

(T) = This is the turnpoint button. In real life the competitors carry sealed cameras with which to photograph the turns, which are selected to be some well-known,easily-recognizable landmark. In the simulation, all you have to do is maneuver your sailplane to the correct location near the turn and press 'T'. If you are out of position, an error message will be flashed up. Correct locations for turning at each turn are as follows:
Turn One: You must be at a higher value for Y (than the turn) and within 2 units of the value for X.
Turn Two: You must be at a lower value for Y (than the turn) and within 2 units of the value for X.
Example: If the first turn of the task is at X = 50 Y = 50 then to turn correctly you must be at Y = 50.1 (or higher) and X = 48 to X = 51.9. Also note that as soon as you get a 'good turn' message at turn one the program shifts all further analysis to refer to turn two.

(F) = This is the finish button. When you have completed the task, having rounded both turnpoints, you head for home and the finish line at coordinates X = 20 and Y = 20 (your original take-off point). When you cross the finish line - that is when both your X and Y coordinates are less than twenty - press 'F'. This will stop the timer and display your landed position back at the airfield.

(R) = This is your relaunch button. If, when you are initially launched,you cannot find a thermal to keep you airborne, the relaunch facility will place you back at the original launch position and height. This will then give you a second try at the task. Remember that you can only take 'one' relaunch and that this must be taken within the first hour of flight.

(K) = Stands for kill control input. If you press this control (you should hold it down for around 8 seconds), the control input routine, marked thus ***, ceases to run, ensuring a faster overall program execution. Of course, you would only use K if certain that control inputs would not be required for a while - to cancel, just hold down K again.

Unlike most so-called simulation programs available today "Sailplane Derby" is a fairly close model of actual cross-country gliding; (the author is an experienced glider pilot). Therefore,a certain level of skill is required for efficient operation of the program. It is suggested that the following training procedure is followed during the first few program runs.

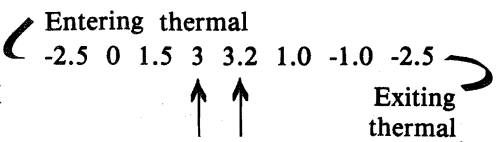
1. Run program.
2. Define weather with thermal strength set at 3 and wind strength set at 1.
3. Defer launch 2 hours.
4. Enter any valid turn-point co-ordinates (as you will not require turn-points for training).

You will now be launched into a favorable meterological environment with strong thermals and little wind drift.

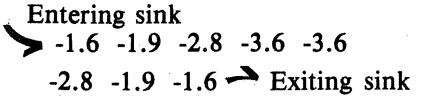
You should note that the program always initialises one strong thermal close to your launch position. Therefore, as soon as the cockpit display is printed, you should commence a gentle turn, say to the right, at a turn rate of around 25. Also, it is a good idea to slow down to 50 m.p.h. You will now see a panorama of the sky moving across your field of view. When you locate a nearby cloud, that is to say one that is both wide and high in your field of vision, straighten out the sailplane and fly in the direction of the cloud. Try to point your sailplane slightly to one side of this cloud, let us say to the left of the center.

It is essential to understand that when you approach the cloud it will disappear prior to your arrival at the location of the thermal. The disappearance is NOT an indication that you have reached the thermal: therefore, continue to glide in a straight line. DO NOT TURN AT THIS TIME.

If the cloud has a thermal under it (one in every five has sink!), you should notice first a rise in the sink rate and then your variometer should follow approximately the following progression. (The actual numbers will vary with the strength and height of the thermal).

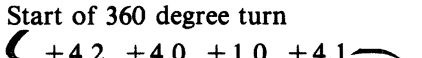


When the progression has reached almost the peak climb rate (between the two arrows) apply full right turn (90) and you should have your glider approximately central in the thermal. If you were unlucky enough to hit sink, the progression of variometer readings will be different, similar to those shown below.



The only thing to do here is to steer away and try another nearby cloud.

Looking on the bright side, let us assume that you are climbing in a thermal at the maximum turn rate of 90 which indicates a complete 360 degree turn in four program loops. It is possible in this situation that your climb rate may be varying (see example of variometer readings below).



This occurrence indicates that you are not correctly centered in the thermal. There are many techniques used for centering, the most common being the so-called 'worst heading' method. To use this technique wait until the least favorable indication of climb appears on the variometer, in our example +1.0, then wait one program loop until you get the next variometer reading which is + 4.1. Quickly straighten out, and in the very next program loop,apply full turn in the same direction as previously. The effect of this procedure is to move the center of your circles into a position nearer the center of the thermal.

It is suggested that you fly the sailplane in various thermals,practicing centering the glider and training yourself in general flying skills prior to actually attempting to fly a set cross-country task. You should soon gain experience in the handling of the sailplane. This may take a little time but, once mastered,the program will provide interest which will not wane as do some of the more trivial simulations on the market.

SUGGESTED PRACTICE TECHNIQUE

10-30 Titles and Copyright statement.

```
10 CLS:PRINT@462,CHR$(23);"SAILPLANE DERBY"
20 PRINT@650,"BY DAVID J.T. NUNN.":FORN=1TO2700:NEXT
30 CLS:PRINT@393,"(D.J.T.NUNN. - LONDON,ENGLAND. COPYRIGHT 1979)
":PRINT@594,"( VERSION 3.4 / 1.77 MHZ ).";:FORN=1TO2700:NEXT
```

40 Dimension arrays; define integer variables.

```
40 CLEAR160:DIMTH(30,4),XY(3),WW(3),VV(3),X(3),Y(3),H(3),S(3),R(
3),W(3),D(3),K(3),B$(3),L(5):DEFINTD,N,Q,M,O,F,S,W,U,B,C,P:L(4)=
1:L(5)=1
42 RANDOM
```

44-60 Define user selected weather conditions.

```
44 CLS:PRINT"DO YOU WISH TO DEFINE YOUR OWN WEATHER CONDITIONS O
R ACCEPT A  RANDOMISED WEATHER PATTERN
1 = OWN
2 = RANDOM"
46 PRINT"(1 OR 2)"
48 W$=INKEY$:IFW$=""THEN48
50 IFW$="2"THEN70
51 IFW$="1"THEN52ELSE46
52 CLS:PRINT"ENTER THERMAL STRENGTHS
1 = WEAK
2 = MEDIUM
3 = STRONG"
53 INPUT"(1,2 OR 3 THEN ENTER)";A1:IFA1<1ORA1>3THEN53
54 CLS:PRINT"ENTER WIND SPEED
1 = 3-5 MPH.
2 = 7-13 MPH.
3 = 19-25 MPH."
55 INPUT"(1,2 OR 3 THEN ENTER)";A3:IFA3<1ORA3>3THEN55
57 IF A3=3 THEN A3=5
60 A4=(9-A1)/10:A5=.6:A2=RND(6):GOTO100
```

70-80 Define randomised weather conditions.

```
70 CLS:PRINT"ONE MOMENT...":A1=RND(3):A4=(9-A1)/10:A5=.6
80 A2=RND(6):A3=RND(6)
90 '
```

100-130 Randomiser selects values for thermals (X & Y coordinates, strengths).

```
100 FORN=1TO30:TH(N,1)=RND(425)+130:NEXT
110 FORN=1TO30:TH(N,2)=RND(475)+175:NEXT
120 FORN=1TO30:TH(N,3)=(RND(5)*A1)+7:NEXT
130 Q5=0:F2=0
```

140-160 Turns one thermal in every five into sinking air on a random basis.

```
140 FORN=1TO30:T1=RND(5)
150 IFT1=1THENTH(N,3)=INT(TH(N,3)*-.6)
160 NEXTN
```

170 Ensures there is at least one thermal within gliding range of the start by loading a thermal into front of array, using a very limited set of conditions for the randomiser.

```
170 TH(1,1)=RND(120)+140:TH(1,2)=RND(120)+140:TH(1,3)=(RND(5)*A1
)+7
```

180-200 Sets the vertical centroid of the thermal above ground level into the last line of the thermal matrix.

```
180 FORN=1TO30:IFTH(N,3)<0THEN200
190 TH(N,4)=RND(A1*600)+(A1*500)
200 NEXTN
210 IFF9=1THEN630
220 F9=1
```

230-270 Enter player's names and set player counter.

```
230 CLS:INPUT"NUMBER OF PLAYERS ";P1
240 IFP1<1ORP1>3THEN230
250 INPUT"TYPE NAMES OF PLAYER(S)-AFTER EACH PRESS ENTER ";B$(1)
260 IFP1=1THEN280ELSEINPUTB$(2)
270 IFP1=2THEN280ELSEINPUTB$(3)
```

280-450 Print meteorological report.

```
280 CLS:PRINT"METEOROLOGICAL DATA"
290 PRINT"THERMAL STRENGTH: ";
300 IFA1=1PRINT"WEAK"
310 IFA1=2PRINT"MODERATE"
320 IFA1=3PRINT"STRONG"
330 PRINT"CLOUD DEVELOPMENT EXPECTED :  ";
340 PRINT" ISOLATED SUMMER CUMULUS."
350 PRINT"WIND VELOCITY (FROM THE WEST)";
360 IFA3=<2PRINT":  3-5 MPH."
370 IFA3<5ANDA3>2PRINT":  7-13 MPH."
380 IFA3>4PRINT":  19-25 MPH."
390 IFA1=1THENPRINT"FOUR HOURS OF CONVECTION EXPECTED"
400 IFA1=2THENPRINT"FIVE HOURS OF CONVECTION EXPECTED"
410 IFA1=3THENPRINT"SIX HOURS OF CONVECTION EXPECTED"
420 T1=15-((1-A4)*10)
430 PRINT"THERMALS EXPECTED TO START AT ";T1;"00 HOURS."
440 PRINT"VISIBILITY IS EXPECTED TO BE 20 MILES."
450 PRINT@788,"PRESS ANY KEY TO CONTINUE"
460 IFA3>4THENT2=RND(399)+100:T3=RND(500)+150:T4=RND(3)*A1+6:FOR
N=27TO30:TH(N,1)=T2:T2=T2+40:NEXT:FORN=27TO30:TH(N,2)=T3:NEXT:FO
RN=27TO30:TH(N,3)=TH(27,3):NEXT
470 A$=INKEY$:IFA$=""THEN470ELSECLS
```

480-510 Is launch deferment required if so increment A4 and A5.

```
480 PRINT"YOU WILL BE LAUNCHED HALF AN HOUR AFTER
CONDITIONS HAVE BECOME SOARABLE, AT ";T1;"30 HOURS.":A4=A4+.05
490 INPUT"COMPETITORS MAY REQUIRE A LATER LAUNCH. THEREFORE IF A
 LAUNCH  POSTPONMENT IS REQUIRED ENTER NUMBER OF HOURS OF POSTP
ONMENT  (1 OR 2), IF NONE REQUIRED ENTER 0";T2
500 IFT2<0ORT2>2THENPRINT"ENTER 0,1 OR 2":GOTO490
510 T2=T2/10:A4=A4+T2:A5=A5+T2
520 '
```

530 Initialise variables for launch of first sailplane.

```
530 H(1)=3000:H(2)=3000:H(3)=3000:S(1)=60:S(2)=60:S(3)=60:W(1)=6
0:W(2)=60:W(3)=60:D(1)=45:D(2)=45:D(3)=45:K(1)=3000:K(2)=3000:K(
3)=3000:X(1)=200:X(2)=200:X(3)=200:Y(1)=200:Y(2)=200:Y(3)=200:R(
1)=0:R(2)=0:R(3)=0
540 B=1:T1=0:T2=0:T3=0:T4=0
```

550-600 Request positions of turnpoints and load into memory.

```
550 PRINT"ENTER TURNPOINTS (TURN ONE MUST BE NORTH OF TURN TWO)"
560 INPUT"TASKSETTER TO TYPE X CO-ORDINATE OF FIRST TURNPOINT(10
-70)";U1
570 INPUT"Y CO-ORDINATE";U2
580 INPUT"TASKSETTER TO TYPE X CO-ORDINATE OF SECOND TURNPOINT(1
0-70)";U3
590 INPUT"Y CO-ORDINATE";U4
600 IF(U1<10ORU1>70)OR(U2<10ORU2>70)OR(U3<10ORU3>70)OR(U4<10ORU4
>70)THENPRINT"ENTER COORDINATES BETWEEN 10 & 70":GOTO560
```

602 Call up map S/R to indicate positions of turnpoints.

```
602 CLS:PRINT"TURNPOINTS ARE DISPLAYED AS CROSSES ON THE FOLLOWI
NG MAP.":FORN=1TO4200:NEXT:CLS:GOSUB3060:FORN=1TO10000:NEXT:CLS
610 IFP1=1THENL(2)=1:L(3)=1
620 IFP1=2THENL(3)=1
```

continued on page 59

A Sailing Simulation

CARIBBEAN CRUISING

by Mark Pelczarski and Jim Klink

CARIBBEAN CRUISING for the Apple works with 16K memory and requires Applesoft in ROM

With this simulation, you won't have to worry about flashing screens, obtrusive sound effects, being shot at, or blasting an opponent out of the water. Just sit back and maneuver your sailboat around the islands. As the winds vary, you'll occasionally have to tack as you travel into the wind, or change your foresail to take maximum advantage of conditions. You won't have to worry about fatal mistakes; you cannot run aground, crash into another boat, or rip out your mast or sails. The worst that will happen is your boat will be drawn in red and sit still if you have the wrong sail for the conditions. Once you have some sailing skill, you may even want to try challenging someone to a race.

Sailing a Boat:

Your boat is equipped with a main sail and three different foresails: a jib, a genoa (sometimes called a jenny), and a spinnaker. When you start, the jib will be in place. Your control over the boat is through the game paddle. The knob controls your rudder, and the button allows you to change your foresail. At the lower left corner of the screen (and at the lower right for a second boat, if there are two), you will see your direction and which sail you have in place. Zero degrees is north, 90 degrees is east, 180 degrees is south, and 270 degrees is west. At the bottom center of the screen will be the exact wind direction and speed. There is also a wind vane in the upper right corner.

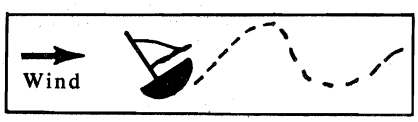
A jib is a relatively small foresail, and its maximum speed is attained when you are travelling almost at a right angle to the wind. It is still effective with the wind behind you, but its capabilities decrease considerably as you begin to head into the wind. It is the safest sail at higher wind speeds, and the only one that will work in this program if the wind speed is over 35 mph.

A genoa is actually a large jib. It works the same as a jib, but with high wind speeds it is too large and should be replaced with a jib.

A spinnaker is a very large balloon-like sail. It is most effective with the wind directly behind you. As the wind approaches you from a right angle, the spinnaker's effectiveness rapidly decreases. Because of its size, you should not use this sail when the wind speed is over 30 mph.

When sailing, your boat will be drawn in the approximate direction that you are heading, but you should rely heavily on your compass reading at the bottom of the screen. The screen is drawn so that north is at the top. The computer will not let you go off the edge of the screen, nor will hitting the other boat or an island affect you.

When heading into the wind, you must tack, or zigzag, back and forth to get from one point to another. This means you may have to change headings several times to avoid going directly into the wind.



('Pathways through the Rum'?)

To keep in the spirit of things, it wouldn't be a bad idea to fire up your blender to create some concoctions that would add to the atmosphere of this simulation. I just happen to have a couple of ideas here:

The Pina Colada

The Basics: Pineapple Juice, Cream of Coconut (available in cans), and Dark Rum. The proportions aren't too important - usually more pineapple juice and less coconut. For a great improvement, add vanilla ice cream. To make Strawberry Coladas, add frozen strawberries; for Banana Coladas, add a banana or two. In general, throw in whatever you want. Blend well and serve over ice. A cherry or some fresh pineapple adds a nice touch.

The Landfall

Throw all of the following (or reasonable fascimiles) into a blender: Orange Juice, Pineapple Juice, Sweet and Sour Mix (alternately, Whiskey Sour Mix, or Lemon, Lime, and a little sugar), Dark Creme de Cacao, Cream of Coconut, and Dark Rum. Vanilla ice cream also does wonders. I usually go heavier on the orange juice, and lighter on the sweet and sour mix and coconut. Blend well, of course, and serve over ice.

The Morning Sun

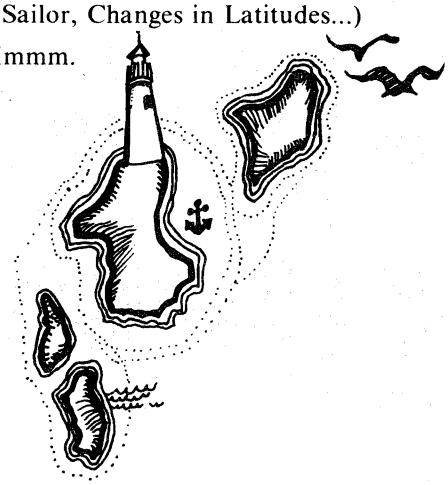
Fill a blender 2/3 with Orange Juice. Throw in a Banana or two, a couple scoops of Cream of Coconut, Vanilla Ice Cream (if available), and Dark Rum. Cover the blender and turn it on. It also goes great with breakfast, minus the rum.

It should be noted that all of the above taste just fine without rum, so if you have an aversion to fermented sugar cane, or are under-age, you may still partake in these fine offerings.

Suggested Listening

To complete the mood, you may want to take a run down to your local record store to see if they have any Steel Band albums. Nonesuch Records has one called Sounds of the Sun, and I've seen others around. An alternative would be any album by Jimmy Buffett (Volcano, Son of a Son of a Sailor, Changes in Latitudes...)

Mmmm.



Sailboat's Documentation:

Variables, in alphabetical order:(Note: all subscripted variables, except S$, will have subscripts of 0 or 1, corresponding to boat 0 or boat 1.)

BC(0-1): Boat Color. BC(0) equals 7 (white), and BC(1) equals 4 (black).

BD(0-1): Boat Direction, in degress off North, as explained in the accompanying article.

BN: Number of boats (players). Value will be 1 or 2.

BR(0-1): Boat Rotation, and BS(0-1): Boat Shape.

These 2 are closely related. There are 4 boat shapes (so that as you turn, your boat will never appear upside-down). Each shape will be used in two possible directions, giving 8 possible views of the boat. Each boat shape is drawn so that its X, Y coordinates will determine the bow position on the screen. The rotations correspond to those necessary for the 2 views that are used with each shape. Apple rotations of 0, 16, 32, 48 correspond to actual clockwise rotations of 0, 90, 180, or 270 degrees.

Shape 1 Shape 2 Shape 3 Shape 4

DD: Used in two different ways to compute angles. In one case, it's a difference of 2 measures. In the other case, it's used to determine boat rotation.

FS(0-1): Foresail. Possible values: 1-Jib, 2-Genoa, 3-Spinnaker, 0-Sail is down.

L: Location in which shape subroutine will put data.

PF: Paddle flag. 0 if waiting for boat 0 to put up sail, 1 for boat 1, and 2 if both boats are under sail. Because the sail choice is put in through the keyboard, only one boat's sail may be down at any given time.

S$: Used by initial shape subroutine when reading.

S$(1-3): Sail name. 1-Jib, 2-Genoa, 3-Spinnaker.

SP: Speed, used to compute new position.

T: Whose turn. Counts 0,1,0,1...

U: Unit distance, set to 4. Changing this will cause the sailboats to move around the screen in larger or smaller steps.

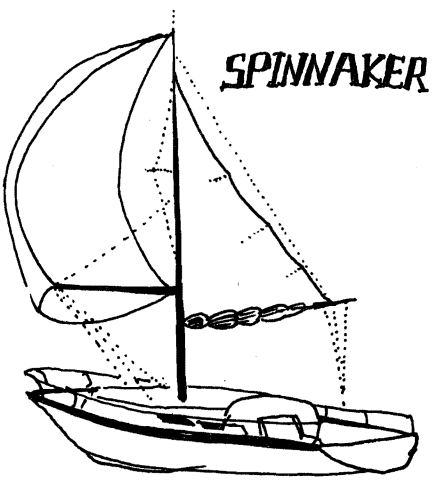
V: Value poked into memory by subroutine that reads the shape tables.

WD: Wind direction.

WS: Wind speed, varies between 10 and 40 mph.

X(0-1), Y(0-1): X,Y coordinates of each boat on the screen. X can be 6 to 273, left to right; Y can be 6 to 153, top to bottom.

XN, YN: New X, Y coordinates, as they are computed.




```
 LIST
5  REM  C.1980 MARK PELCZARSKI
```

Lines 10,20 cause shape table initialization.

```
10  GOSUB 900
20  POKE 232,0: POKE 233,28
```

Lines 25-45 initialize variables and screen.

```
25 U = 4
30  DIM BS(1),BR(1),FS(1),BD(1),X
    (1),Y(1),BC(1)
35  DIM S$(3):S$(1) = "JIB      "
    :S$(2) = "GENOA    ":S$(3) =
    "SPINNAKER"
40 PF = 2:WD =  INT ( RND (1) * 3
    60):WS =  INT ( RND (1) * 11
    ) + 20
42  HOME : VTAB 21: PRINT "1 OR 2
     SAILBOATS? ";: GET S$: IF S
    $ < "1" OR S$ > "2" THEN 42
43 BN =  VAL (S$)
45  SCALE= 1: HGR
```

Line 46 creates blue background.

```
46  HCOLOR= 6: HPLOT 0,0: CALL 62
    454
```

Line 50-80 initialize more variables and draw the boats at starting positions.

```
50  FOR T = 0 TO BN - 1
60 FS(T) = 1:BS(T) = 1:BR(T) = 32
    :Y(T) = 130 + T * 20:X(T) =
    40 + T * 20:BD(T) = 90:BC(T)
     = 7 - T * 3
70  HCOLOR= BC(T): ROT= 32: DRAW
    1 AT X(T),Y(T)
80  NEXT
```

Line 90 draws the wind vane.

```
90  SCALE= 3: ROT= WD / 5.625: DRAW
    5 AT 261,20
```

Line 100 prints initial text information.

```
100  HOME : VTAB 21: PRINT "DIR:"
    ;BD(0); TAB( 14);"WIND DIR:"
    ;WD; TAB( 32);"DIR:";BD(1): PRINT
    "JIB"; TAB( 14);"WIND SPD:";
    WS; TAB( 32);"JIB"
```

Line 105 checks if ESC has been pressed. If so, go back to start.

```
105  IF PF = 2 AND  PEEK ( - 1638
    4) = 155 THEN  POKE  - 16368
    ,0: CLEAR : GOTO 25
```

Lines 110,112 change wind speed, plus or minus 1 mph.

```
110 WS = WS +  INT ( RND (1) * 3)
     - 1: IF WS < 10 THEN WS = 10
112  IF WS > 40 THEN WS = 40
```

Lines 120-150 erase the wind vane, choose a new direction in the range plus or minus 2 degrees, and redraw vane.

```
120  HCOLOR= 6: SCALE= 3: ROT= WD
     / 5.625: DRAW 5 AT 261,20
130 WD = WD +  INT ( RND (1) * 5)
     - 2: IF WD > 360 THEN WD =
    WD - 360
140  IF WD < 0 THEN WD = WD + 360
150  HCOLOR= 7: ROT= WD / 5.625: DRAW
    5 AT 261,20
```

Line 160 updates the text.

```
160  VTAB 21: HTAB 23: PRINT WD;"
     ": HTAB 23: PRINT WS
```

Line 165 redraws the islands.

```
165  ROT= 0: SCALE= 1: HCOLOR= 1:
     DRAW 6 AT 55,100: DRAW 6 AT
    200,75: DRAW 6 AT 80,50
```

Lines 170-820 repeat for each player's turn.

```
170  FOR T = 0 TO BN - 1
```

Lines 180-200 change the boat's direction according to the paddle setting.

```
180 DD =  PDL (T) / 25 - 5
```

```
190 BD(T) = BD(T) + DD: IF BD(T) >
    360 THEN BD(T) = BD(T) - 360
200 IF BD(T) < 0 THEN BD(T) = BD
    (T) + 360
```

Lines 210-270 check if the sail is down, and if so, has a new one been selected? The ASCII values referred to in lines 230-250 correspond to J, G, and S, respectively.

```
210 IF PF < > T THEN 300
220 IF PEEK ( - 16384) < 128 THEN
    400
225 A = PEEK ( - 16384): POKE -
    16368,0
230 IF A = 202 THEN FS(T) = 1: GOTO
    260
240 IF A = 199 THEN FS(T) = 2: GOTO
    260
250 IF A = 211 THEN FS(T) = 3: GOTO
    260
255 GOTO 400
260 VTAB 23: HTAB 12: PRINT "
        ";
270 PF = 2: VTAB 22: HTAB T * 31 +
    1: PRINT S$(FS(T)): GOTO 400
```

Lines 300-350 check if the paddle button is pushed, and if so, take down the sail. If PF is already 0 or 1, this is skipped. PEEK(T-16287) is > 127 if the appropriate button is being pressed.

```
300 IF PF < > 2 OR PEEK (T - 1
    6287) < 128 THEN 400
310 FS(T) = 0:PF = T: VTAB 22: HTAB
    T * 31 + 1: PRINT "DOWN
    ":BC(T) = 7 - T * 3
320 VTAB 23: HTAB 15: PRINT "CHO
    OSE SAIL"
330 PRINT "  (J) JIB, (G) GENOA
      (S) SPINNAKER";
340 VTAB 23: IF T = 0 THEN HTAB
    12: PRINT "<--": GOTO 400
350 HTAB 26: PRINT "-->": GOTO 4
    00
```

Line 400 computes the difference in degrees between the wind and boat directions.

```
400 DD = ABS (WD - BD(T)): IF DD
    > 180 THEN DD = 360 - DD
```

Line 410 computes a base speed.

```
410 SP = (WS / 35) * U
```

Line 420 checks if foresail is down; if so, the speed is zero.

```
420 IF FS(T) = 0 THEN SP = 0: GOTO
    600
```

Lines 430-470 compute the speed if the spinnaker is up.

```
430 IF FS(T) < > 3 THEN 500
440 IF WS > 30 THEN SP = 0:BC(T)
    = 5: GOTO 600
450 IF DD > 90 THEN SP = 0: GOTO
    600
460 IF DD < 45 THEN SP = .9 * SP
    : GOTO 600
470 SP = 1800 / DD ^ 2 * SP: GOTO
    600
```

Lines 500-550 compute the speed if the jib or genoa are up.

```
500 IF FS(T) < > 2 THEN 520
510 IF WS > 35 THEN SP = 0:BC(T)
    = 5: GOTO 600
520 IF DD > 140 THEN SP = 0: GOTO
    600
530 IF DD > 90 THEN SP = 8100 /
    DD ^ 2 * SP: GOTO 600
540 SP = (DD / 300 + .7) * SP
550 IF FS(T) = 1 THEN SP = SP *
    .8
```

Lines 600-630 convert the directions to normal trigonometric standards and compute vectors for change in X and Y.

```
600 DD = BD(T) + 90: IF DD > 360 THEN
    DD = DD - 360
610 DD = DD * 3.14 / 180
620 XN = X(T) - COS (DD) * SP
630 YN = Y(T) - SIN (DD) * SP
```

Lines 640-670 check the range of the new coordinates.

```
640 IF YN > 153 THEN YN = 153
650 IF YN < 6 THEN YN = 6
660 IF XN > 273 THEN XN = 273
670 IF XN < 6 THEN XN = 6
```

Line 680 erases the boat.

```
680 HCOLOR= 6: ROT= BR(T): SCALE=
    1: DRAW BS(T) AT INT (X(T))
    , INT (Y(T))
```

Lines 690-770 compute the shape and rotation that should be used in drawing the boat with the given direction.

```
690 DD = BD(T) + 22.5
700 IF DD > 360 OR DD < = 45 THEN
    BS(T) = 3:BR(T) = 0: GOTO 80
    0
710 IF DD > 45 AND DD < = 90 THEN
    BS(T) = 2:BR(T) = 16: GOTO 8
    00
720 IF DD > 90 AND DD < = 135 THEN
    BS(T) = 1:BR(T) = 32: GOTO 8
    00
730 IF DD > 135 AND DD < = 180 THEN
    BS(T) = 2:BR(T) = 32: GOTO 8
    00
740 IF DD > 180 AND DD < = 225 THEN
    BS(T) = 1:BR(T) = 48: GOTO 8
    00
750 IF DD > 225 AND DD < = 270 THEN
    BS(T) = 4:BR(T) = 48: GOTO 8
    00
760 IF DD > 270 AND DD < = 315 THEN
    BS(T) = 3:BR(T) = 48: GOTO 8
    00
770 BS(T) = 4:BR(T) = 0
```

Lines 800,810 draw the boat at its new location and update the coordinates.

```
800 ROT= BR(T): HCOLOR= BC(T): DRAW
    BS(T) AT INT (XN), INT (YN)
810 X(T) = XN:Y(T) = YN
```

Line 815 updates the text.

```
815 VTAB 21: HTAB 5 + T * 31: PRINT
     INT (BD(T));" "
```

Lines 820 and 830 loop back.

```
820 NEXT T
830 GOTO 105
```

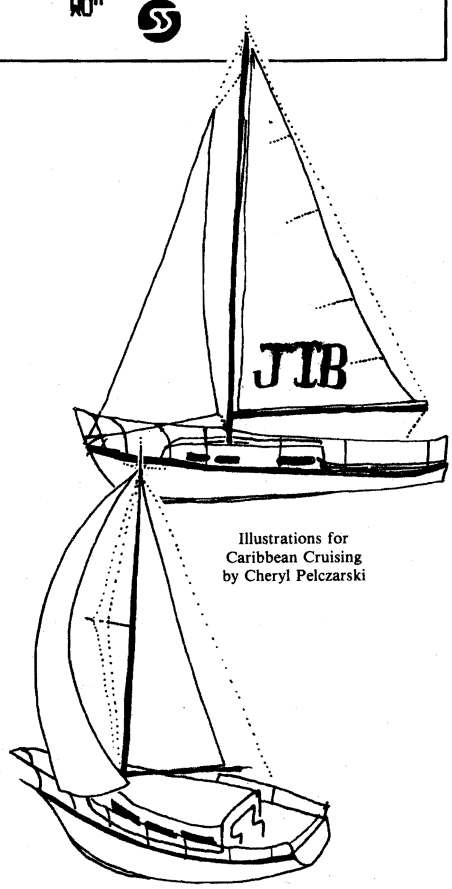
Lines 900-990 are instructions, which may be omitted if line 10 is changed to GOSUB 1000.

```
900 HOME : PRINT "  C A R I B B
    E A N   C R U I S I N G"
905 PRINT : PRINT "    (OR 'PATH
    WAYS THROUGH THE RUM')"
910 PRINT : PRINT "     DON'T WO
    RRY ABOUT FLASHING SCREENS,"

915 PRINT "OBTRUSIVE SOUND EFFEC
    TS, OR BEING SHIP-"
920 PRINT "WRECKED.  YOU DON'T H
    AVE TO BLAST ANY"
925 PRINT "OPPONENTS OUT OF THE
    WATER;  JUST SIT"
930 PRINT "BACK AND SAIL. OR IF
    YOU WISH, CHALLENGE";
935 PRINT "A FRIEND TO A RACE AR
    OUND THE ISLANDS."
940 PRINT : PRINT "     THE INFO
    RMATION AT THE LOWER LEFT"
945 PRINT "WILL BE FOR THE WHITE
     SAILBOAT AND"
950 PRINT "PADDLE 0.  THE INFORM
    ATION AT THE LOWER"
955 PRINT "RIGHT IS FOR THE BLAC
    K SAILBOAT AND"
960 PRINT "PADDLE 1.  THE DIAL C
    ONTROLS THE RUDDER,";
965 PRINT "AND THE BUTTON ALLOWS
     YOU TO CHANGE YOUR";
970 PRINT "SAIL.  IF YOUR BOAT I
    S SHOWN IN RED,"
```

```
975  PRINT "YOUR SAIL IS INAPPROP
     RIATE FOR THE WIND"
980  PRINT "CONDITIONS AND WOULD
     NORMALLY CAUSE"
985  PRINT "TROUBLE.  THE 'ESC' K
     EY LETS YOU MOVE"
990  PRINT "BOTH BOATS BACK TO TH
     E START."
```

Lines 1000-1200 put the shape table into memory. Discovering that most of this data was in the range 9-63, I put the initial information in decimal into the DATA statement at 1030. The rest of the data I was able to condense by adding 24 to each value and using an ASCII character (see pp. 138-9 in the Applesoft manual). The net result was half as much data in lines 1150-1200, where each character replaces 2 hexadecimal digits. (The same technique could have been used in the Invaders program in May, had I been observant earlier.) Anticipating possible questions, the following characters appear in lines 1150-1200: '0', the letter, not the number; '%', not a 'Z'; and the strange blurbs are consecutive 'W's. Other characters that may appear fuzzy are the number '3' and the letter 'S'.

```
1000  FOR L = 7168 TO 7181
1010  READ V: POKE L,V
1020  NEXT L
1030  DATA    6,0,14,0,177,0,84,1
     ,247,1,154,2,160,2
1040  FOR T = 1 TO 6
1050  GOSUB 1100
1060  NEXT T
1070  VTAB 24: HTAB 12: PRINT "<P
     RESS ANY KEY>";: GET S$: RETURN
1100  READ S$
1110  FOR V = 1 TO  LEN (S$)
1120  POKE L, ASC ( MID$ (S$,V,1)
     ) - 24
1130 L = L + 1: NEXT V
1140  POKE L,0:L = L + 1: RETURN
1150  DATA "!AEEEEEE)WWWWWWW+EEE
     EEEEEM33333373+!!A!!!!!)WWWW
     WW73+!!AEEEEE-3WWWW73+!!AEE
     EE%)33WWW73+!!AEEE%!)333WW
     73+!!AEE%!!)3333WW73+!!AE%!!
     !)33333W73+!!A%!!!!)33333373
     +"
1160  DATA  "!!!!!!!!)33333333+!!
     !!!!!!)33333333+!!AEE!!!)333
     SWW33+!!!AEE!!)33SWW333+!!!!
     AEE!)3SWWW333+!!!AEEEE)SWW7S
     W33+!!AE%!EE-WW73WWW3+!AEEE%
     !EMW73WWWWW+AEEEEEE%!M73WWWW
     WO"
1170  DATA "!%!!!!!!)3333333S+A%!
     !!!!!)3333333SOE%!!!!!!)WWWW
     WWWWOE%AEEEEE-3WWWW7SOE%AEE
     EE%)33WWW7SOE%AEEE%!)333WW
     7SOE%AEE%!!)3333WW7SOE%AE%!!
     !)33333W7SOA%A%!!!!)3333337S
     +"
1180  DATA  "!!!!!!!!IW3333333+!!
     !!!!!AMWW333333+!!!!!!AEMWWW
     33373+!!A%!AEEMWWW7W73+!!AE
     EEEMWW73WW73+!!AEE%!EMW73WW
     W33+!!!AEE%!M73WWW333+!!!!AE
     E%)3WWW3333+!!!!!AEE-WWW3333
     3+"
1190  DATA   "<<,MW"
1200  DATA  "!EEAE%%!)3SWWWW7W/A%
     AEEEEE)SW7SWWW3+AEEEE%!A-33W
     WWW7WOE!AEEEE%)SWWSWWW30!!EA
     %E!E-S73WS7WW+!EEA%E!!)3333S
     733+!!!A%!!!)333WWW73+!AEEEE
     E!)3WWWWWW/EEEEEEEEMWWWWWW
     WO"
```



Illustrations for Caribbean Cruising by Cheryl Pelczarski

GEMA

PROTOUR 80

PROTOUR 80 is an S-80 program for Level II 16K and up.

by David Bohlke

It is no longer necessary to sit in front of your TV screen on weekends and drool over the pro golfers' pars and birdies. Now you can face the same challenge on your TRS with PROTOUR 80. And with a lot of practice and a little skill you can become a scratch golfer!

This links-simulation will generate a different random nine hole course each time you play. Your ball and the tee will be displayed on the left of the screen; and the fairway and green will be constructed towards the right. Each player, in turn, will hit from the tee to the green. After everyone has reached the green, each player will putt out. At the completion of every hole a current scorecard will be displayed.

Accuracy and club selection is very important when you are shooting for the green. If your ball wanders off the fairway or into the sandtrap — you can expect an erratic second shot. And if you are wild enough to hit the ball off the screen, you will be penalized one stroke and distance; so you will need to hit the ball from your original location.

The direction of your tee and fairway shots must be a number between 10 and 49. In this representation, 10 is due north, 20 is east, 30 is south, and 40 is west (screen reference). For more accurate shots, select a direction between these cardinal directions. For example, to shoot north-east, use 15 for direction; and to shoot east-north-east, you can try 17.

Club selection for distance is also important. The woods (1-4) will hit from about 260-200 yards, respectively. The irons (1-9) will travel from 200-90 yards, respectively. And the wedge will give a distance of approximately 30-70 yards.

For both direction and distance there is always a slight random variation. After all, there aren't any golfers who hit a perfect shot everytime. But with a little practice, you should be able to reach the green in the regulation number of strokes for par fairly easily - and better players will be able to get close to the cup (+), allowing a shorter putt.

As all golfers know, most rounds are won or lost on the putting green. PROTOUR 80 is no exception. You will find the contour greens more challenging. The green representation will be unique for every player since you are shooting at the cup from different angles. To putt, you need only enter the distance of your shot. As a rough estimate, each SET position is one unit of length - so the entire green is 127 units (feet) long. But, in addition to the straight line distance, you'll also have to take into account the contour of the green. A ball going down hill, for instance, will speed up which means that it needs a shorter distance entered as compared to an equivalent level putt.

Well, enough of my jibberish -it's time you tee-up and learn for yourself. As a final note, the more experienced players should enter their names first at the start of the game. Then you will always tee off first, and the other players can get a fairly good idea of direction and distance. You can consider this 'honor' as your handicap. At least you won't waste any time looking for lost balls!

```
{5-30 Initialization

5 REM  xxx  DAVID BOHLKE    COGGON, IA     16 JULY 79  xxx
6 REM  xxx  PRO TOUR '80  xxx
7 DEFINTA-Y:CLEAR300
20 GOSUB7000
30 GOTO9000

{7000-7170 Directions

6999 REM  xxx  DIRECTIONS  xxx
7000 CLS:PRINTTAB(19)"P R O   T O U R   8 0":PRINT
7010 PRINT"     THIS IS A SIMULATED GOLF GAME FOR 1-4 PLAYERS, U
SING A"
7020 PRINT"RANDOMLY GENERATED 9 HOLE COURSE.  EACH HOLE HAS TWO
INPUT"
7030 PRINT"CYCLES.  EACH PLAYER IN TURN WILL HIT THE BALL FROM T
HE TEE"
7040 PRINT"TO THE GREEN.  AFTER ALL PLAYERS HAVE REACHED THE GRE
EN, EACH"
7050 PRINT"GOLFER WILL AGAIN IN TURN PUTT OUT.  WHEN EACH HOLE I
S COM-"
7060 PRINT"PLETED, A CURRENT SCORECARD WILL BE DISPLAYED."
7100 PRINT"     THE BETTER PLAYERS SHOULD <ENTER> THEIR NAMES FI
RST, SINCE"
7110 PRINT"THEY WILL ALWAYS BE SHOOTING FIRST.  THIS WILL ENABLE
 THE"
7120 PRINT"OTHER PLAYERS TO GET A BETTER IDEA OF DIRECTION AND D
ISTANCE."
7130 PRINT"IF YOUR FAIRWAY SHOT MISSES THE FAIRWAY OR LANDS IN T
HE SAND-"
7140 PRINT"TRAP, THERE IS A POSSIBILITY OF AN ERRATIC SECOND SHO
T.  ALSO,"
7150 PRINT"WHEN PUTTING ON THE CONTOUR GREENS; BE SURE TO ADD A
LITTLE"
7160 PRINT"TO DISTANCE WHEN GOING UPHILL, AND SUBTRACT A LITTLE
WHEN GOING"
7165 PRINT"DOWNHILL.              <ENTER> TO CONTINUE . . .";
7170 INPUTA$:RETURN

{8000-8400 Displays scorecard
     P$(I) contains the names of the players
     TT(I) total score for each player (accumulated)
     Z is the number of holes played

7999 REM  xxx  SCORECARD  xxx
```

```
{8000-8146 Prints scorecard lines, hole#, etc.

8000 CLS:PRINT"O P E N    G O L F"
8010 FORI=1TO9:PRINT@I*5+136,I;:NEXT
8050 PRINT@260,"P A R";:PRINT@133,"HOLE #";
8060 PRINT@186,"TOTAL"
8080 FORI=1TOPL:PRINT@I*128+256,P$(I);:TT(I)=0:NEXT
8100 FORI=1TO9:FORJ=1TO2*PL+3STEP2
8110 PRINT@I*5+J*64+71,CHR$(149);:NEXTJ,I
8120 FORJ=1TO2*PL+3STEP2
8130 PRINT@J*64+121,CHR$(191);:NEXT
8140 FORI=1TOPL+2:PRINT@I*128+64,STRING$(63,140);
8142 IFI=1,PRINT@I*128+64,STRING$(63,188);
8144 IFI=2,PRINT@I*128+64,STRING$(63,143);
8146 NEXT

{8160-8165 Prints Par for each hole

8160 X=0:FORI=1TOZ:PRINT@I*5+264,TP(I);:X=X+TP(I)
8165 PRINT@314,X;:NEXT

{8210-8290 Prints players' scores on scorecard

8210 FORI=1TOPL
8220 FORJ=1TOZ
8230 TT(I)=TT(I)+P(I,J):PRINT@J*5+I*128+264,P(I,J);
8260 PRINT@I*128+314,TT(I);
8270 NEXTJ
8290 NEXTI
8400 PRINT@960,"<ENTER> TO CONTINUE . . .";:INPUTA$:RETURN

{9000-9140 Main game loop

8999 REM   xxx   MAIN GAME LOOP   xxx
9000 CLS:GOSUB9200:CLS:GOSUB9900

{9030 loop for nine holes
     TP(Z) is par for each of nine holes

9030 FORZ=1TO9
9032 GOSUB9800:TP(Z)=P

{9050 loop for tee-to-green for each player
     M,N is location of ball
     PL is the number of players
     If B=1 then tee is at the top of the screen

9050 FORZZ=1TOPL
9055 M=1:IFB=1,N=RND(6)+5ELSEN=RND(6)+35

{9060-9070 P(ZZ) records each player's score for each
     hole

9060 S=0:GOSUB9600
9070 P(ZZ,Z)=P(ZZ,Z)+S
9080 NEXT

{9090-9100 Loop for each player to putt out on green

9090 FORZZ=1TOPL:S=0:DC=B(ZZ):GOSUB9400
9092 P(ZZ,Z)=P(ZZ,Z)+S
9100 NEXT
9110 GOSUB8000
9120 NEXT Z
9130 PRINT@980,"<ENTER> FOR ANOTHER GAME . . .";
9140 INPUTA$:RUN

{9200-9225 Inputs the number of players and their names

9199 REM   xxx   INITIALIZATION   xxx
9200 CLS:PRINTCHR$(23):PRINT"G O L F":PRINT:PRINT
9205 INPUT"HOW MANY PLAYERS (1-4) ";PL:PRINT
9206 IFPL<1ORPL>4,9200
9210 FORI=1TOPL:PRINT"NAME OF PLAYER # ";I;:INPUTP$(I)
9215 P$(I)=LEFT$(P$(I),11):NEXT:PRINT
9225 RETURN

{9400-9590 Putting Sequence

9399 REM   xxx   DIRECTION AND DISTANCE ON FAIRWAY SHOTS   xxx
9400 CLS

{9404-9420 SETs contour green

9404 UD=RND(3)-2:IFUD=0,9404
9405 YS=26:HL=RND(6)+6:HT=0
9410 FORX=0TO126STEP2:SET(X,YS):SET(X,YS+1):SET(X,YS+2)
9411 SET(X+1,YS):SET(X+1,YS+1):SET(X+1,YS+2)
9412 IFX>75ANDX<100,9420
9414 IFHT=0ANDX>20 AND X<70 ANDRND(12)=1,UD=-UD:HT=1
9415 IFRND(HL)=HL,YS=YS+UD

{9420-9426 Computes location of cup (FX and FY)

9420 NEXT:FX=RND(13)+83:FY=2
9425 IFPOINT(FX,FY),9430
9426 FY=FY+1:GOTO9425

{9430-9436 SETs flag and pole

9430 FORJ=0TOFY-3:SET(FX,J):NEXT
9432 FORJ=FXTOFX+14:SET(J,1):SET(J,7):NEXT
9434 FORJ=1TO7:SET(FX+13,J):SET(FX+14,J):NEXT
9436 PRINT@FX/2+66,Z;

{9438 Cup location

9438 RESET(FX-1,FY):RESET(FX,FY):RESET(FX+1,FY):RESET(FX-1,FY+1)
:RESET(FX,FY+1):RESET(FX+1,FY+1)

{9440-9442 no putt necessary (DC is distance to cup)

9440 IFDC>5,9450ELSEPRINT@960,"IT'S IN THE CUP !! ??";
9442 SET(FX,FY+1):SET(FX+1,FY+1):FORI=1TO2000:NEXT:RETURN

{9450-9460 SETs ball (variables B,C)

9450 C=0:B=FX-DC:IFB<0,B=RND(4)
9452 IFPOINT(B,C+2),9460
9454 C=C+1:GOTO9452
9460 SET(B,C):SET(B+1,C)

{9465-9480 INPUT length of putt

9464 REM   xxx   PUTTING ROUTINE   xxx
9465 PRINT@896,CHR$(31);
9470 PRINT@960,"<ENTER> LENGTH OF PUTT (GREEN IS 127 FEET WIDE)"
;
9475 PRINT@896,P$(ZZ);"'S PUTT . . .";:INPUTPP
9480 IFPP<1ORPP>111,9465

{9490 Move ball left or right
```

```
9490 IFB<FX,MM=1ELSEMM=-1
9492 PM=0:S=S+1

 {9500-9540 Loop to move ball on green

9500 RESET(B,C):RESET(B+1,C)
9502 IFB>80ANDB<110,9520
9503 IFMM=-1,9510
9505 IFPOINT(B+2,C+1),C=C-1:PP=PP-RND(3)
9506 IFPOINT(B,C+2)=0,C=C+1:PP=PP+RND(3)
9507 GOTO9520
9510 IFPOINT(B-1,C+1),C=C-1:PP=PP-RND(3)
9515 IFPOINT(B+1,C+2)=0,C=C+1:PP=PP+RND(3)
9520 B=B+MM:IFB>124,B=124ELSEIFB<3,B=3
9530 PM=PM+1:SET(B,C):SET(B+1,C)
9535 IFMM=-1ANDB<3,9460
9536 IFMM=1ANDB>120,9460
9540 IFPM<>PP,9500

 {9550-9590 Get another putt if necessary, else return
     with the number of strokes (S)

9550 IFB=FX OR B=FX-1,9560ELSE9465
9560 RESET(B,C):RESET(B+1,C):SET(B,FY+1):SET(B+1,FY+1)
9590 RETURN

 {9600-9684 Move shots from tee to green
     M,N is the location of the ball
     S is the number of strokes

9599 REM   xxx   MOVE BALL DOWN FAIRWAY   xxx
9600 GOSUB9700
9602 X=0:M1=M:N1=N:S=S+1
9605 ST=0:IFPOINT(M,N-1)ORPOINT(M,N+1),ST=RND(4)
9606 GOSUB9950

 {9610-9655 loop to move ball

9610 IFR=0ORC=0,MM=R:NN=C:GOTO9615
9612 IFRND(R)=R,MM=1ELSEMM=0
9613 IFRND(C)=C,NN=1ELSENN=0
9615 IFDR>30,MM=-MM
9616 IFDR>20ANDDR<40,NN=-NN
9620 RESET(M,N):RESET(M+1,N)
9622 PRINT@G,G$;:PRINT@PS,S$;
9630 M=M+MM:N=N-NN
9635 IFMM=0,D=D-9-ST:GOTO9650
9637 IFNN=0,D=D-5-ST:GOTO9650
9638 D=D-10-ST
9650 IFM<20RM>125ORN<20RN>45,9680
9652 SET(M,N):SET(M+1,N)
9655 IFD>0,9610

 {9660-9661 Check if on green

9660 MG=(G-INT(G/64)*64)*2:NG=INT(G/64)*3-1
9661 IFM-MG<0 OR N-NG<0 OR M-MG>18 OR N-NG>9,9600

 {9662 B(ZZ) is each player's distance to cup
9662 DC=ABS(10-(M-MG))*5+ABS(5-(N-NG))*12+RND(2):B(ZZ)=DC
9664 FORJ=1TO3:PRINT@PA,CHR$(30);:FORII=1TO300:NEXT
9665 PRINT@PA,P$(ZZ);" IS ON THE GREEN";:FORII=1TO500:NEXT:NEXT
9670 RETURN
9680 PRINT@PA,"OUT OF BOUNDS - PENALTY STROKE";
9682 RESET(M,N):RESET(M+1,N):M=M1:N=N1:FORII=1TO2000:NEXT
9684 S=S+1:GOTO9600

 {9700-9790 Inputs player's club choice
     D is the distance the ball will travel
     DR is the direction of the hit

9699 REM   xxx   GET FAIRWAY SHOT   xxx
9700 GOSUB9790:PRINT@PA,P$(ZZ);" <ENTER> CLUB CHOICE":C=0
9705 PRINT@PA+64,"WOOD(W) OR IRON(I) OR SAND WEDGE(S)";:PRINT@PA
+50,"";
9706 SET(M,N):SET(M+1,N):RESET(M-1,N):RESET(M+2,N)
9710 INPUTC$:IFC$="S",D=40+RND(30):GOTO9730
9712 IFC$<>"I"ANDC$<>"W",9700ELSEGOSUB9790
9714 IFC$="I"PRINT@PA,"WHICH IRON (1-9) ";:INPUTC:GOTO9720
9716 PRINT@PA,"WHICH WOOD (1-4) ";:INPUTC
9720 C=INT(C):IFC<1,9700
9722 IFC$="I"ANDC>9,9700
9724 IFC$="W"ANDC>4,9700
9726 IFC$="I",D=(10-C)*13+80+RND(20):GOTO9730
9728 D=200+(4-C)*10+RND(20)
9730 GOSUB9790
9732 PRINT@PA,"   10";:PRINT@PA+64,"40 ";CHR$(136);CHR$(132);" 2
0";
9734 PRINT@PA+128,"   30";
9740 PRINT@PA+76,"DIRECTION";:PRINT@PA+12,"<ENTER>    ";
9745 INPUTDR:DR=INT(DR):IFDR<10ORDR>49,9730
9750 GOSUB9790:RETURN
9790 FORI=PATOPA+128STEP64:PRINT@I,CHR$(30);:NEXT:RETURN

 {9800-9882 SETs fairway and green
     Y is the yards to cup
     P is the par

9799 REM   xxx   CONSTRUCT FAIRWAY   xxx
9800 Y=260+RND(200):P=4

 {9805 White screen

9805 FORI=0TO960STEP64:PRINT@I,STRING$(63,191);:NEXT:B=RND(2)

 {9806 PA is the PRINT@ locations for inputs

9806 IFB=1,PA=832ELSEPA=0
9810 X=RND(6):IFX=1,P=3:Y=150+RND(90)
9812 IFX=6,P=5:Y=490+RND(100)
9815 T=RND(4)+3:D=12:R=0:IFB=1,C=3ELSEC=44
9816 PRINT@448,"HOLE #";Z;" ";:PRINT@512,"YARDS";Y;:PRINT@576,"P
AR";P;
9818 E=3*P+RND(5):F=RND(3)+2

 {9820-9832 Blacks out fairway

9820 IFB=2,9822
9821 FORI=CTOC+D:RESET(R,I):RESET(R+1,I):RESET(R+2,I):NEXT:R=R+3
:GOTO9823
9822 FORI=CTOC-DSTEP-1:RESET(R,I):RESET(R+1,I):RESET(R+2,I):NEXT
:R=R+3
9823 IFRND(8)=1,C=C+RND(3)-2:IFC<1,C=1ELSEIFC>46,C=46
9824 IFRND(4)=4,D=D+1ELSEIFR>100,9835
9825 IFR<EORRND(T)<T,9820
9826 IFB=1,C=C+1ELSEC=C-1
9827 IFD>18,D=18
9828 IFB=1ANDC>25,C=25
9829 IFB=2ANDC<22,C=22
9830 IFP=3ANDR>40,9835ELSEIFP=4ANDR>65,9835
9832 GOTO9820
```

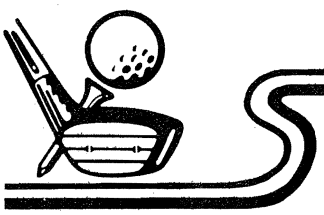
```
{9835-9860 picks location of green and sandtrap
     G is the green PRINT @; G$ is PRINT STRING$
     PS is sandtrap PRINT @; S$ is the PRINT STRING$

9835 G=INT(Y/10)-6:X=R-3
9836 IFG>50,G=G-1:GOTO9836
9840 FORJ=13TO34STEP2:IFPOINT(X,J),9844
9842 YY=J
9844 NEXT
9850 GG=INT(YY/3)*64+(RND(3)-2)*64-192:IFGG<64ORGG>831,9850
9853 G=G+GG
9857 IFB=1,PS=G-RND(8)+(RND(2)-3)*64-1
9858 IFB=2,PS=G-RND(8)+RND(2)*64-1
9860 PRINT@G,G$;:PRINT@PS,S$;
9880 IFB=1PRINT@64,"TEE";ELSEPRINT@896,"TEE";
9882 RETURN

{9900-9979 Common routines

9899 REM   xxx   PRINTING STRINGS   xxx
9900 G$="   G   G   "+CHR$(26)+STRING$(11,24)+" G   +   G "+CHR$
(26)+STRING$(11,24)+"   G   G   "
9905 S$=CHR$(184)+CHR$(191)+CHR$(180)+CHR$(26)+STRING$(3,24)+CHR
$(139)+CHR$(191)+CHR$(135):RETURN

{9940-9979 Direction indicators (R and C) for adjusting the
     movement of the ball in the tee-to-green sequence

9940 IFST>0,DR=DR+(RND(5)-3):IFDR<0,DR=49ELSEIFDR>49,DR=1
9950 IFDR=10ORDR=30,R=0:C=1:RETURN
9952 IFDR=20ORDR=40,R=1:C=0:RETURN
9960 IFDR<20,DZ=DR-10:GOTO9970
9962 IFDR<30,DZ=30-DR:GOTO9970
9964 IFDR<40,DZ=DR-30:GOTO9970
9966 DZ=50-DR
9970 ON DZ GOTO9971,9972,9973,9974,9975,9976,9977,9978,9979
9971 R=3:C=1:RETURN
9972 R=2:C=1:RETURN
9973 R=1:C=1:RETURN
9974 R=1:C=2:RETURN
9975 R=1:C=3:RETURN
9976 R=1:C=4:RETURN
9977 R=1:C=7:RETURN
9978 R=1:C=10:RETURN
9979 R=1:C=15:RETURN
```

NEW RADIO SHACK COMPUTERS

Radio Shack announced three new computers on August 1. The TRS-80 Color Computer, part 26-3001, has 4K at $399; part 26-3002 has 16K (maximum), at $599, has an extended Basic in ROM, and uses plug-in ROM software cartridges. The processor is a 6809, has an RS-232C interface, and an RF modulator for direct connection to your TV. Cassette recorder (1500 baud), joysticks, and modems are among the options available.

The Model III is an upgrade of the Model I. Part 26-1061 has 4K at $699, part 26-1062 has 16K at $999, and part 26-1063 has 32K and two forty-track double density disk drives for $2495. The unit is one piece construction with room for two drives internal, and possibility of two more external. Lower case and real time clock are standard with 16K and up. Most Model I software will run, and it has both a 500 baud and a 1500 baud tape interface. It also has a built in parallel printer interface.

The third computer is a six ounce, seven inch long, hand-held computer that sells for $299. It is programmable in BASIC, has 1.9K RAM, and 57 alphanumeric keys. It also has an optional cassette interface for $49.



MASTER'S GOLF

MASTER'S GOLF — an Atari program requiring at least 16K of memory and one joystick

by David Bohlke

This version of simulated golf is a modification of PROTOUR 80, which was written on a TRS-80, and is described elsewhere in this issue. Since both games are basically the same, it would be a good idea to read the PROTOUR 80 article first as it describes general game play.

Some of the major differences between the two games are the color, sound, graphics, and input routines utilized on the ATARI. You will need to insert a joystick into paddle slot No. 1. To pick your club (wood, iron, wedge), push the stick to get the different selections; then press the fire button to lock in your choice. To select your club number and direction, push the joystick until the value is reached and then press the fire button. Direction values are the same as on PROTOUR 80: North equals 10, East equals 20, South equals 30, and West equals 40. Any number from 10-49 is a legal direction entry. The order of tee-off will vary with respect to the lowest score on the previous hole.

When everyone has reached the green, the putting sequence will begin. The player with the longest straight line putt will go first. However, don't always judge successive putts strictly by this distance as the contour of the green will distort the real distance needed to sink the putt. As with club selection, just push the joystick to input your distance, then press the fire button. For a rough estimate, the width of the green is eighty units (feet).

The line listing description and variable definitions should give you a pretty good idea of program flow for MASTERS' GOLF. Instead of commenting on the program specifics, I would like to offer some general impressions of writing on the TRS-80 and ATARI.

For those who haven't had a chance to program in color and with sound -- there is no comparison possible to a mute black and white monitor -- my TRS-80 has become a dust collector since the ATARI arrived. Usually it takes about 50% more time to write a program and get the colors and sounds just right. But the end result is well worth the effort.

Graphics on the ATARI are much more impressive, thanks to the color and resolution (320 x 192 max). The PLOT X, Y command is similar to SET(X,Y) on the TRS-80. Instead of a RESET command, it is only necessary to change the COLOR register on the ATARI, then use the same routine as in PLOT. To test a plotted point, the TRS-80 uses a POINT(X,Y) command. On the ATARI, the command is LOCATE X,Y,Z. Here, X and Y denote the location and Z will contain the value of the COLOR register. This makes it possible to distinguish between the various colors used. The ATARI also has a DRAWTO command for connecting any two points with a straight line. There is also a command which will 'fill in' a pre-defined shape with a specific color.

String handling on the ATARI is a bit more complicated than on the TRS-80. It is not possible to INPUT strings into an array. Rather, the string value must be INPUT into a scratch string and then packed into the array. Check the name input routines from both listings for a comparison.

Cassette save and load operations are superior on the ATARI. There isn't a volume setting to fool with or any plugs to constantly pull in and out (as with the earlier 80's). I've probably had only two or three bad loads, but I'm not sure if it was the recorder, the tape, or the operator at fault.

The keyboard and edit features on the ATARI also have a pleasant advantage. There is upper/lower case, reverse video, auto repeat, no keybounce, and about 30 special characters. Editing is convenient with versatile insert/delete commands. It is even possible to alter line numbers without changing the line statement.

There are several minor differences in the general BASIC statements. The ATARI does not support an ELSE clause on the IF...THEN statement. But it does have GOSUB var. and GOTO var., which can be handy. Also, the ATARI does allow the abbreviation of statement commands. Another nice feature is that a program can be CONTINUED after a BREAK and an error correction.

It is probably too early to judge the manufacturers' support. However, I did receive in two separate mailings a Reference Manual and an Introduction to Basic Programming tape which were not ready for distribution when I bought the machine. That seems like a good indication.

Perhaps my earlier comment is the best recommendation I have for the ATARI -- my TRS 80 is getting dusty (but hopefully not rusty).

```
1 REM  MASTERS' GOLF
2 REM by David Bohlke
20-50, 66
initialization
G(40) holds contour of green
B$(7) scratch string
N$(20) holds names of four players
N(4) length of players names
P(9) par for nine holes
S(4,9) each players score for each
   of 9 holes
PU(4) lengths of each players putt
T(4),H(4) retains player order of hit
   for tees and greens
20 DIM A$(22):A$=" WOOD   IRON   WEDGE
"
22 DIM G(40),B$(7),N$(28),N(4),P(9),S(4,
9),PU(4),T(4),H(4)
50 GRAPHICS 0:SETCOLOR 2,11,6:SETCOLOR 4
,15,6

inputs number of players and names
52 ? :? "      M A S T E R S'   G O L F
":?
54 ? ,"by  David Bohlke":? :?
56 ? " ENTER   number of golfers (1-4) "
;:INPUT N$:PL=ASC(N$)-48:IF PL<1 OR PL>4
 THEN RUN
```

```
60 ? :? :FOR I=1 TO PL
62 PRINT " ENTER  name of golfer # ";I;"
 ";:INPUT B$:?
64 NK(I)=LEN(B$)+NK(I-1):N$(NK(I-1)+1,NK(I))
=B$:NEXT I
66 FOR I=1 TO 4:HK(I)=I:T(I)=I:NEXT I
```

200-350 Main Game Loop

loop for 9 holes

```
200 FOR HL=1 TO 9
```

plots fairway, green, tee, yardage, sandtrap

```
210 GOSUB 900:POKE 752,1
```

sets player order of hit from tee to green

```
212 FOR I=1 TO 5:FOR J=1 TO PL-1
214 X=S(T(J),HL-1):Y=S(T(J+1),HL-1)
216 IF X>Y THEN Z=T(J):T(J)=T(J+1):T(J+1
)=Z
218 NEXT J:NEXT I
```

loop for each player to hit from tee to green. S is the number of shots taken.

```
220 FOR TX=1 TO PL:S=0
225 PRINT "HOLE # ";HL;"     PAR ";P(HL)
;"     YARDS ";YA
230 TU=T(TX):IF PL=1 THEN TU=1
240 GOSUB 700:S(TU,HL)=S
242 FOR I=1 TO 4:PRINT " IT'S  ON  THE
 GREEN "
244 FOR J=70 TO 1 STEP -1:SOUND 0,J,10,4
:NEXT J:NEXT I
245 NEXT TX:IF PL=1 THEN 250
```

sets player order to putt-out

```
246 FOR I=1 TO 4:HK(I)=I:NEXT I:FOR I=1 T
O 5:FOR J=1 TO PL-1
247 IF PUK(HK(J))<PUK(HK(J+1)) THEN Z=HK(J):H
(J)=HK(J+1):HK(J+1)=Z
248 NEXT J:NEXT I
```

loop for each player to putt

```
250 FOR TX=1 TO PL:S=0
252 IF PL=1 THEN TU=1:GOTO 260
254 TU=HK(TX)
260 GOSUB 500:FOR I=1 TO 400:NEXT I
280 S(TU,HL)=S(TU,HL)+S:NEXT TX
```

scorecard routine

```
300 GOSUB 400
```

for 9 holes

```
350 NEXT HL
```

then end

```
360 PRINT "END":END
```

400-490 prints scorecard

colors and prints

```
400 GRAPHICS 0:SETCOLOR 2,15,4:SETCOLOR
4,RND(0)*16,8:POKE 752,1
402 ? :? "      M A S T E R S'   G O L F
 ":? :?
404 PRINT "HOLE":? :PRINT "PAR"
406 POSITION 36,4:PRINT "TOT";
```

prints hole numbers

```
408 FOR I=1 TO 9:POSITION I*3+6,4:PRINT
I;:NEXT I
```

lines in scorecard

```
410 FOR I=1 TO 38:FOR J=1 TO PL+1
411 POSITION I,J*3+5:PRINT " ";:NEXT J:N
EXT I
```

prints par for each hole and total

```
412 TP=0:FOR I=1 TO HL:POSITION I*3+6,6:
PRINT P(I);:TP=TP+P(I):NEXT I
413 POSITION 36,6:PRINT TP;
```

lines in scorecard

```
414 FOR I=1 TO 10:FOR J=4 TO PL*3+8
416 POSITION I*3+5,J:PRINT " ";:NEXT J:N
EXT I
```

prints players names and score for each

```
420 FOR I=1 TO PL:POSITION 1,I*3+7
425 PRINT N$(NK(I-1)+1,NK(I));
430 T=0:FOR J=1 TO HL:T=T+S(I,J)
434 POSITION J*3+6,I*3+7:PRINT S(I,J);
440 NEXT J
445 POSITION 36,I*3+7:PRINT T;
450 NEXT I
```

check for end of game

```
470 IF HL=9 THEN POSITION 2,23:PRINT " E
ND     MASTERS'  GOLF  ":END
```

else return

```
480 POSITION 2,23:PRINT "Press  FIRE  to
 continue  .  .  .";
482 IF STRIG(0)=0 THEN RETURN
484 SOUND 0,RND(0)*200,10,6:GOTO 482
490 RETURN
```

500-660 putting routine

colors

```
500 GRAPHICS 5:COLOR 1:SETCOLOR 0,12,6:S
ETCOLOR 4,14,8:SETCOLOR 2,0,12:POKE 752,
1:SETCOLOR 1,12,10
502 COLOR 1
```

DC is the distance to the cup

```
510 DC=INT(PUK(TU))
```

plots contour of the green

F is the location of the flag

```
530 G=30:C=1:IF 0.5>RND(1) THEN C=-1
532 D=RND(1)*20+10:R=0.7
534 E=RND(1)*20+40
535 F=INT(RND(1)*20)+45
540 FOR I=0 TO 78 STEP 2
545 PLOT I,G:PLOT I+1,G+G*(I/2)=G
546 PLOT I,G+1:PLOT I+1,G+1
548 IF 5>ABS(F-I) THEN 570
550 IF I>E THEN R=0.95
552 IF I>D THEN C=-C:D=100
560 IF R<RND(1) THEN G=G+C
568 IF G>39 THEN G=39
570 NEXT I
```

B is the location of the ball

```
580 B=INT(F-DC)
```

cup impression in the green

```
585 COLOR 4:PLOT F,G(F/2):PLOT F,G(F/2)+
1
```

plots for flag and pole

```
590 COLOR 2:PLOT F,0:DRAWTO F,G(F/2)-3
591 PLOT F+8,4:DRAWTO F,0
592 PLOT F+8,4:DRAWTO F,8
```

no putt is necessary!

```
595 IF PUK(TU)<5 THEN PRINT " IT'S  IN  T
HE  CUP  !  !  !  ":COLOR 3:PLOT F,G(F/2)+1:
SOUND 0,40,0,8:RETURN
```

plots ball

```
596 COLOR 2:PLOT B,G(B/2)-1
```

inputs length of putt

```
600 L=20:S=S+1
602 PRINT "Hole # ";HL;"  ♦♦♦♦♦  Laying
 ";S(TU,HL)+S-1
603 PRINT N$(NK(TU-1)+1,NK(TU));" 'S  PUT
T ":PRINT " PUTT  LENGTH ——————— ? ? "
;L
604 IF STRIG(0)=0 THEN 610
605 IF STICK(0)=15 THEN 604
606 ? :IF STICK(0)=14 THEN L=L+1:IF L>80
 THEN L=1
607 IF STICK(0)=13 THEN L=L-1:IF L<1 THE
N L=80
609 SOUND 0,250-L*3,10,4:GOTO 603
```

moves ball across green

```
610 C=0:N=1:IF B>F THEN N=-1
620 COLOR 4:PLOT B,G(B/2)-1
622 B=B+N:COLOR 2:PLOT B,G(B/2)-1
625 IF B>77 THEN B=77:GOTO 600
626 IF B<1 THEN B=1:GOTO 600
627 IF G(B/2)<G(B/2+1) THEN L=L+INT(RND(
1)*2)
628 IF G(B/2)>G(B/2+1) THEN L=L-INT(RND(
1)*2)
630 C=C+1:IF C>=L THEN 650
635 SOUND 0,B*2,10,RND(1)*10
640 GOTO 620
```

another putt attempt is needed

```
650 IF B<>F THEN 600
```

putt is good

```
655 COLOR 4:PLOT B,G(B/2)-1
656 COLOR 2:PLOT B,G(B/2)+1
660 RETURN
```

700-850 moves ball from the tee to the green

B1,B2 is plot location of the ball

```
700 B2=T1-2+INT(RND(1)*5):B1=3:COLOR 1:P
LOT B1,B2:G=0:C=1:Z=2
```

input routines for club choice

(D is distance)

```
701 C1=C:IF C>1 THEN C1=C*8-8
702 PRINT N$(NK(TU-1)+1,NK(TU));" 'S  TURN
 :":PRINT "SELECT CLUB ?  ";A$(C1,C1+6)
703 IF STRIG(0)=0 THEN 707
704 IF STICK(0)=15 THEN SOUND 0,RND(0)*2
00,10,4:GOTO 703
705 C=C+1:IF C>3 THEN C=1
706 ? :GOTO 701
707 PRINT "COMPUTED":FOR I=1 TO 100:NEXT
 I
708 IF C=3 THEN 750
709 I1=4:I=1:IF C=2 THEN I1=9
710 PRINT A$(C1,C1+6);" #  ? ";I:? :?
712 IF STRIG(0)=0 THEN GOTO 725+C*5
714 IF STICK(0)=15 THEN SOUND 0,RND(0)*2
00,10,4:GOTO 712
716 I=I+1:IF I>I1 THEN I=1
718 GOTO 710
730 D=(4-INT(I))*10+200+RND(1)*40:GOTO 7
55
735 D=(9-INT(I))*12+90+RND(1)*20:GOTO 75
5
750 D=30+RND(1)*20
```

inputs for direction (DR)

```
755 D1=1:D2=1:DR=20
756 PRINT "COMPUTED":FOR I=1 TO 100:NEXT
 I
758 PRINT "DIRECTION  ?? ";DR:? :?
759 IF STRIG(0)=0 THEN 764
760 IF STICK(0)=15 THEN SOUND 0,RND(0)*2
00,10,4:GOTO 759
761 IF STICK(0)=14 THEN DR=DR+1:IF DR>49
 THEN DR=10
762 IF STICK(0)=13 THEN DR=DR-1:IF DR<10
 THEN DR=49
763 GOTO 758
```

X,Y are the movement offsets for direction D4, D5 are modifications for hits from sand, etc.

```
764 D4=DR:D3=0:IF Z=0 THEN D3=INT(RND(0)
*7)-3:D4=DR+D3:IF D4<10 OR D4>49 THEN 76
4
766 DR=D4:IF Z=0 THEN D5=D/(RND(0)+2):D=
D-D5
767 IF DR<21 THEN X=(DR-10)/10:Y=(20-DR)
/10:D2=-1:GOTO 780
768 IF DR<31 THEN X=(30-DR)/10:Y=(DR-20)
/10:GOTO 780
769 IF DR<41 THEN X=(DR-30)/10:Y=(40-DR)
/10:D1=-1:GOTO 780
770 X=(50-DR)/10:Y=(DR-40)/10:D1=-1:D2=-
1
780 D5=D-RND(0)*D-10:IF D5<20 THEN D5=20

782 IF Z=3 THEN D=D5
```

moves ball down the fairway

```
800 S=S+1:Q=Q+1:IF Y=0 THEN Y=1.0E-03
802 IF X=0 THEN X=1.0E-03
803 COLOR Z:PLOT B1,B2
807 IF X>RND(1) AND Y>RND(1) THEN B1=B1+
D1:D=D-8:B2=B2+D2:GOTO 820
810 IF X>RND(1) THEN B1=B1+D1:D=D-8
812 IF Y>RND(1) THEN B2=B2+D2:D=D-8
820 IF (B1>1) AND (B1<78) AND (B2>1) AND
 (B2<38) THEN 830
826 COLOR 3:PLOT B1,B2:C=1:GOTO 701
830 LOCATE B1,B2,Z:COLOR 3:PLOT B1,B2
832 SOUND 0,B1+B2,12,RND(1)*4+3
835 IF D>0 THEN 803
```

not on the green; get another hit

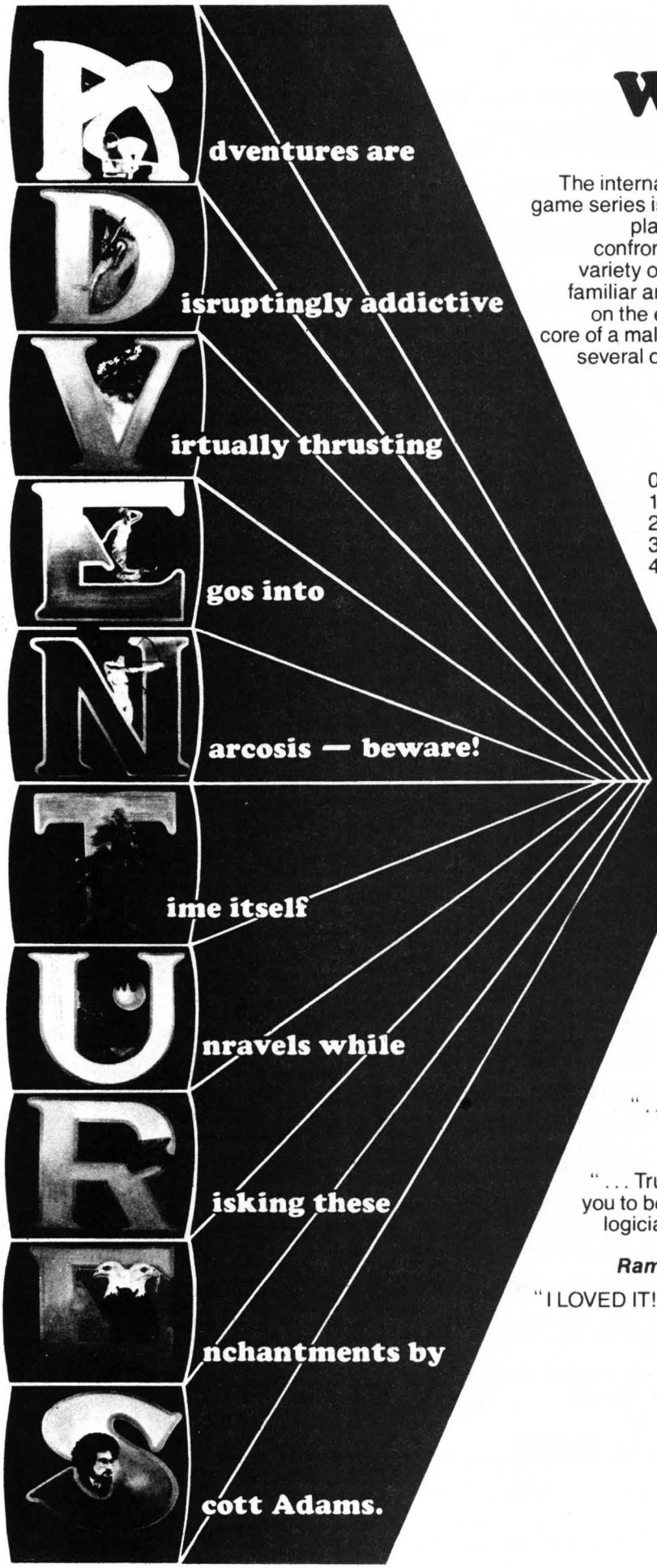
```
840 IF R1<ABS(B1-G1) THEN C=1:GOTO 701
842 IF R1<ABS(B2-G2) THEN C=1:GOTO 701
```

compute length of putt, return

```
845 PUK(TU)=ABS(B1-G1)*7+ABS(B2-G2)*7+RND
(1)*5
850 RETURN
```

900-983 plots fairway, green, tee, sand, gets yardage

continued on page 77



Great Scott! What have you created?

The international reception of Scott Adams' ADVENTURE game series is phenomenal! Each of the ten games takes the player on odyssies of intrigue and danger, through confrontations with various villains, in pursuit of a wide variety of mysteries and treasures, in surroundings both familiar and alien. You may find yourself in a voodoo den, on the edge of a strange and threatening galaxy, in the core of a malfunctioning nuclear reactor, or in any one of the several other settings; ranging from mundane to bizarre.

Challenge your genius! Order an ADVENTURE

Master these ADVENTURES!

0. Special Sampler
1. Adventureland
2. Pirates Adventure
3. Mission Impossible
4. Voodoo Castle
5. The Count
6. Strange Odyssey
7. Mystery Fun House
8. Pyramid of Doom
9. Ghost Town

ADVENTURES are available for Pet™, Sorcerer™, Apple™, TRS-80™ Microcomputer, and other major personal computers.

adventure international
Box 3435 • Longwood, Fl 32750
(305) 862-6917

Look who's raving about ADVENTURE

Out of 50 programs reviewed, Adventure was rated No. 1! "Highly recommended."
80 Software Critique

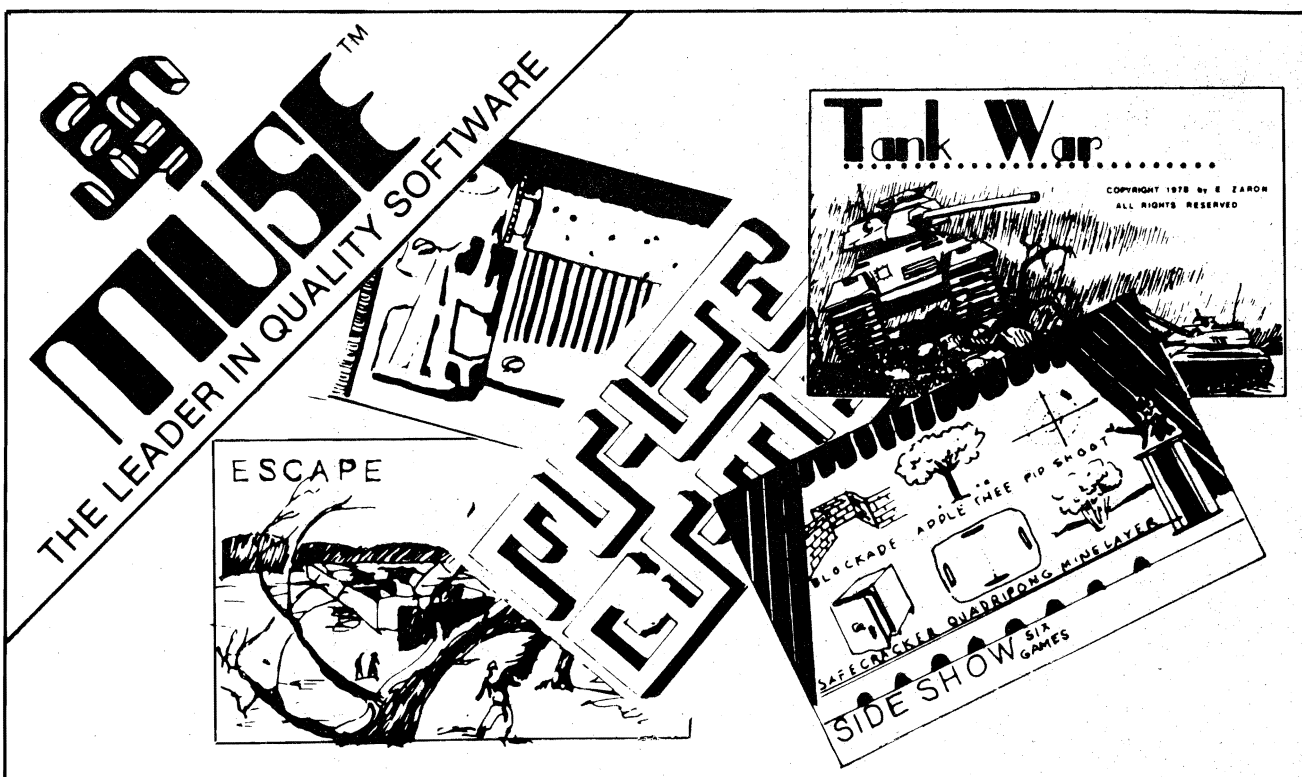
"Adams' Adventure is exquisite. It is a true tour-de-force . . . "
Recreational Computing

" . . . It is worth the money to buy a computer just to play Adventure."
Software Directory Winter 1980

" . . . Truly absorbing simulations. These games require you to be inquisitive, innovative, a thinker, a risk taker, a logician, a warrior, and a lover of real challenges — in short, an Adventurer."
Ramon Zamora, Recreational Computing Issue 4

"I LOVED IT! There was a real challenge to this program . . . "
Captain 80, 80 Microcomputing Issue 1

TO ORDER: CALL TOLL FREE
1-800-258-1790
(In NH call 673-5144)



THE BEST OF MUSE

A COLLECTION OF OUR FIVE MOST POPULAR CASSETTE PROGRAMS NOW ON DISK

MAZE GAME, the original and still the best maze game for Apple II users. Set the size and difficulty level of your maze and watch as it is randomly generated. Special graphics place you inside a colorful maze, with a blue floor, green and yellow walls, and a pink sky above. Player options include getting a maze map, having a compass available, and leaving retraceable footprints on the maze floor. 16K Integer

ESCAPE is an exciting 3-D adventure with special color graphics that place you inside a maze of hallways. Find your way out by following the clues of people you meet inside. But be careful . . . the liars will not hesitate to give you false maps and compasses. It's hard to quit this game! Rated as the number 1 Apple program by the Marin Computer Center. 16K Integer.

SIDE SHOWS is a collection of six exciting games selected for their originality, ease of operation and imaginative use of the Apple. Includes the Apple Tree, Safecracker, Pip Shoot, Minelayer, Blockade and Quadripong. In the Apple Tree two players use the paddles to catch randomly falling apples. In Blockade and Minelayer try to surround your opponent with a moving wall or mine field. Pip Shoot can be played by two or more and the object is to shoot down as many "pips" as possible. Quadripong is four wall pong with varying ball speed and sound. Use the paddle carefully to be the first to open the safe in Safecracker. 16K Integer.

TANK WAR is a state-of-the-art shootout for two players with dramatic sound and graphics. Players choose the style and size of their tank. Steer your tank with the game paddles to pursue and outmaneuver your opponent. Fire two types of missiles to disable and destroy the enemy. 16K Machine Language.

THE MUSIC BOX - start the MUSIC BOX and your Apple comes alive with sound and color. Play it as a three-octave instrument or type in a song and the music box will play it for you! A brilliant full color light show accompanies the music. Standard musical notation includes sharps and flats, note time, rests, dotted notes and tempo. Your songs can be saved on disk and recalled later. All notes can be easily retuned from the keyboard to produce special sound effects. Complete with two songs and documentation that explains basic musical notation. 16K Machine Language.

All on one disk. A $64.75 value for only $39.95

TO ORDER
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

6 South St.,
Milford, NH 03055

The above programs are also available on cassette for $12.95 each.

ROM the ROBOT

ROM THE ROBOT is an Integer BASIC program for the Apple and requires at least 16K.

by Bill Smith

This is the third part of Bill Smith's ROM the ROBOT series, which is an animation tutorial using low resolution graphics on the Apple II. Parts 1 and 2 appear in the June and July issues of Softside: Apple. Part 1 describes the instructions that create the robot's figure and let you turn him in place, make him nod, shake his head, or blink. Part 2 describes the instructions that allow ROM to walk around the screen without falling off. This month's addition combines ROM's abilities into a simple game.

There are a few features of Apple's Integer BASIC that should be noted with regards to this program. In line 20, the PEEK statement checks the keyboard strobe. If a key has been pressed, its ASCII code is put in the variable KEY. If no key has been pressed, the value will be less than 128. Apple's ASCII codes repeat after 128, so both 65 and 193 are codes for the letter A (therefore, line 30 tests for character 194, or a B, etc.).

Two sets of POKE's should also be mentioned. The Apple really has 2 low-res screens, and Bill has set up his program to draw ROM on page 1, do a memory move to page 2, and display page 2. With this method you never see ROM actually being redrawn. Line 70 does the memory move, and lines 80, 9010, and 9020 cause page 2 to be displayed.

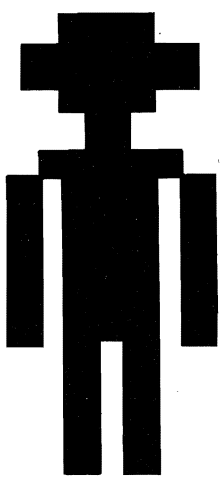
One last item pertains to Integer BASIC. In an IF-THEN statement, if the condition is true, only the first instruction after THEN is affected. The false condition resumes immediately, not at the next program line. (See line 90 for an example. In most other BASIC's the second IF and the GOTO would be meaningless.)

ROM ON PARADE - PART 3

This month, we are going to teach ROM a game to demonstrate some of his many attributes. This will, together with what we have from parts 1 and 2, make a complete game program, so we had better give it a title. How does ROM ON PARADE sound?

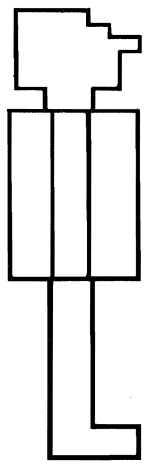
The instructions to play the game are simple, but a certain amount of skill is required to be a successful Drill Sergeant. The idea is to march ROM around the screen using the keyboard commands:

"F" - Front face
"L" - Left face
"R" - Right face
"M" - March forward
"C" - Climb
"D" - Descend
"H" - Halt



When the heel of ROM's foot is placed above one of the three white spots on the screen, it disappears. Cover them all in turn and the game is over. A counter keeps track of the time, so the appropriate commands must be issued promptly.

For those readers who have been typing in ROM from the previous parts of the series, line 30 needs changing to add a new COMMAND subroutine P (parade game). Lines 2080 to 2083 are the subroutine itself. Lines 26 and 65 in the MAIN LOOP are new, as are the 4000 series.



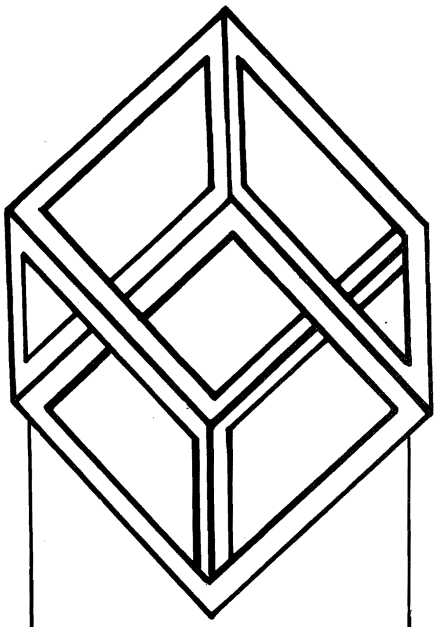
Lines 2080-2083. The new COMMAND subroutine "P" introduces another flag FP which controls the visits to the 4000 set of lines from the MAIN LOOP. The counter C and spot flags (S1, S2 and S3) are set to zero. Short instructions for the game are also given.

Lines 4010 checks if all the spots have been covered. If so we branch to line 4140 to print the score and a comment before resetting the FP flag and ending the game.

If the spots have not all been covered, we increment the counter C in line 4020, rewrite the instructions with the updated score, then in lines 4040 to 4090 redraw each spot, checking if ROM's heel is over it before returning to the MAIN LOOP for another go-around.

To operate, after RUNning the program type "P". Read the instructions, type "F" or some other relevant command and you are off . . .

```
 1 REM  *ROM THE ROBOT*
 2 REM   BILL SMITH
 3 REM   GAMEIER ISLAND
 4 REM   GIBSONS B.C.
 5 REM   CANADA
10 GOTO 9000
20 KEY= PEEK (-16384): IF KEY<
   128 THEN 60
25 PRINT : PRINT : PRINT
26 IF FP=1 THEN FP=2
30 IF KEY$194 AND KEY$198 AND
   KEY$204 AND KEY$206 AND KEY$
   210 AND KEY$217 AND KEY$195
    AND KEY$196 AND KEY$205 AND
   KEY$200 AND KEY$209 AND KEY$
   208 THEN 50
40 A=KEY
50 POKE -16368,0
60 GOSUB A*10
65 IF FP=2 THEN GOSUB 4000
70 POKE 60,0: POKE 61,4: POKE
   62,255: POKE 63,7: POKE 66,
   0: POKE 67,8: CALL -468
80 POKE -16299,0: GR
90 IF X<6 OR X>33 THEN 92: IF
   Y<5 OR Y>24 THEN 94: GOTO 20

92 GOSUB 8010: GOSUB 8000: IF
   X<6 THEN X=X+1: IF X>33 THEN
   X=X-1:A=194: GOTO 20
94 IF Y<5 THEN Y=Y+1: IF Y>24 THEN
   Y=Y-1: GOTO 20
100 COLOR=4: REM  BODY (BU)
110 VLIN Y+1,Y+7 AT X: VLIN Y+1
    ,Y+7 AT X-1: VLIN Y+1,Y+7 AT
    X+1
120 RETURN
200 COLOR=15: REM  HEAD FRONT(HF)
210 PLOT X,Y: HLIN X-1,X+1 AT Y-
    1: PLOT X-2,Y-2: PLOT X-1,Y-
    2: PLOT X+1,Y-2: PLOT X+2,Y-
    2: PLOT X-2,Y-3: PLOT X,Y-3
    : PLOT X+2,Y-3
220 HLIN X-1,X+1 AT Y-4
230 RETURN
250 COLOR=15: REM  EYES BLANK(HB)
260 PLOT X,Y: HLIN X-1,X+1 AT Y-
    1: HLIN X-2,X-1 AT Y-2: HLIN
    X-2,X+2 AT Y-3: HLIN X-1,X+
    1 AT Y-4: HLIN X+1,X+2 AT Y-
    2
270 RETURN
300 COLOR=15: REM  HEAD FRONT LOOKIN
    G DOWN (HFD)
310 HLIN X-1,X+1 AT Y: PLOT X-1
    ,Y-1: PLOT X+1,Y-1: PLOT X-
    2,Y-2: PLOT X,Y-2: PLOT X+2
    ,Y-2: HLIN X-1,X+1 AT Y-3: HLIN
    X-1,X+1 AT Y-4
320 RETURN
400 COLOR=15: REM  HEAD LEFT(HL)
410 PLOT X,Y: HLIN X-1,X+1 AT Y-
    1: HLIN X-2,X+1 AT Y-2: PLOT
    X+1,Y-3: HLIN X,X+1 AT Y-4:
     PLOT X-1,Y-3
420 RETURN
500 COLOR=15: REM  HEAD FACING RIGHT
    (HR)
510 PLOT X,Y: HLIN X-1,X+1 AT Y-
    1: HLIN X-1,X+2 AT Y-2: PLOT
    X-1,Y-3: HLIN X-1,X AT Y-4:
     PLOT X+1,Y-3
520 RETURN
900 COLOR=11: REM  LEG & ARMS FOR RI
    GHT FACE(LR)
910 VLIN Y+8,Y+14 AT X: VLIN Y+
    1,Y+7 AT X: PLOT X+1,Y+14
920 RETURN
950 COLOR=11: REM  LEG & ARMS FOR LE
    FT FACE (LL)
960 VLIN Y+8,Y+14 AT X: VLIN Y+
    1,Y+7 AT X: PLOT X-1,Y+14
970 RETURN
1000 COLOR=11: REM  LEGS AND ARMS FOR
     FRONT FACE (LF)
1005 COLOR=11
1010 PLOT X-2,Y+1: PLOT X+2,Y+1
1015 VLIN Y+2,Y+7 AT X+3: VLIN Y+
     2,Y+7 AT X-3
1020 VLIN Y+8,Y+14 AT X-1: VLIN
     Y+8,Y+14 AT X+1
1050 RETURN
1100 REM  LEGS & ARMS MARCH LEFT (LML
     )
1110 COLOR=11
1120 PLOT X,Y+1: PLOT X-1,Y+2: PLOT
     X-2,Y+3: PLOT X-3,Y+4: PLOT
     X-4,Y+5: PLOT X+2,Y+3
1130 VLIN Y+8,Y+14 AT X: VLIN Y+
     10,Y+13 AT X-2: PLOT X-1,Y+
     9: PLOT X-3,Y+13: PLOT X-1,
     Y+14
1140 RETURN
1150 REM  LEG & ARMS MARCH LEFT 1 (LM
     L1)
1160 COLOR=11
1170 PLOT X,Y+1: PLOT X+1,Y+2: PLOT
     X+2,Y+3: PLOT X+1,Y+4: PLOT
     X,Y+5: PLOT X-2,Y+3: PLOT X-
     3,Y+4: PLOT X-4,Y+5
1180 VLIN Y+8,Y+14 AT X: PLOT X+
     1,Y+11: PLOT X-1,Y+14: PLOT
     X+1,Y+13: PLOT X+2,Y+12: PLOT
     X-1,Y+10
1190 RETURN
1200 REM  LEGS & ARMS MARCH RIGHT (LM
     R)
1210 COLOR=11
1220 PLOT X,Y+1: PLOT X+1,Y+2: PLOT
     X+2,Y+3: PLOT X+3,Y+4: PLOT
     X+4,Y+5: PLOT X-2,Y+3
1230 VLIN Y+8,Y+14 AT X: VLIN Y+
     10,Y+13 AT X+2: PLOT X+1,Y+
     9: PLOT X+3,Y+13: PLOT X+1,
     Y+14
1240 RETURN
1250 REM  LEG & ARMS MARCH RIGHT (LMR
     1)
1260 COLOR=11
1270 PLOT X,Y+1: PLOT X-1,Y+2: PLOT
     X-2,Y+3: PLOT X-1,Y+4: PLOT
     X,Y+5: PLOT X+2,Y+3: PLOT X+
     3,Y+4: PLOT X+4,Y+5
1280 VLIN Y+8,Y+14 AT X: PLOT X-
     1,Y+11: PLOT X+1,Y+14: PLOT
     X-1,Y+13: PLOT X-2,Y+12: PLOT
     X+1,Y+10
1290 RETURN
1940 D=0: REM  BLINK
1941 IF F>1 THEN F=0: GOSUB BU: GOSUB
     LF: IF F=0 THEN 1942: GOSUB
     HB:F=0: RETURN
1942 GOSUB HF:F=1: RETURN
1950 REM  CLIMB
1951 Y=Y-1: GOTO 2051
1960 REM  DESCEND
1961 Y=Y+1: GOTO 2051
1980 D=0: REM  FRONT FACE
1981 GOSUB HF: GOSUB BU: GOSUB LF:
      RETURN
2000 REM  HALT
2001 IF D=1 THEN A=210: IF D=2 THEN
     A=204: POP : GOTO 20
2040 D=2: REM  RIGHT FACE
2041 GOSUB HL: GOSUB BU: GOSUB LL:
      RETURN
2050 REM  FORWARD MARCH
2051 IF D=1 THEN 2052: IF D=2 THEN
     2055: POP :A=206: GOTO 20
2052 X=X+1: GOSUB HR: GOSUB BU: IF
     F=0 THEN 2053: IF F=2 THEN
     2054:F=F1: GOSUB LR: RETURN

2053 F=1:F1=2: GOSUB LMR: RETURN

2054 F=1:F1=0: GOSUB LMR1: RETURN
```



THREE D

Create 3-dimensional graphics!

by Mark Pelczarski

3-dimensional figures can be rotated, shifted, scaled, or distorted. Each figure can be saved on disk and later assembled into larger figures, with each part capable of being manipulated. Screen images may be saved and used with other programs.

48K Applesoft ROM
$29.95 on diskette

TO ORDER
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

```
2055 X=X-1: GOSUB HL: GOSUB BU: IF
     F=0 THEN 2056: IF F=2 THEN
     2057:F=F1: GOSUB LL: RETURN

2056 F=1:F1=2: GOSUB LML: RETURN

2057 F=1:F1=0: GOSUB LML1: RETURN

2060 D=0: REM  SHAKE HEAD
2061 GOSUB BU: GOSUB LF: IF F=0 THEN
     2062: IF F=2 THEN 2063:F=F1:
      GOSUB HF: RETURN
2062 F=1:F1=2: GOSUB HL: RETURN

2063 F=1:F1=0: GOSUB HR: RETURN

2080 GOSUB BU: GOSUB HF: GOSUB LF:
      COLOR=15: PLOT 10,26: PLOT
     20,38: PLOT 30,24
2081 PRINT : PRINT "MARCH ROM AROUND
     SO THAT HIS HEEL IS   OVER EACH
      OF THE THREE WHITE SQUARES IN"

2082 PRINT "TURN.YOU WILL BE SCORED O
     N SPEED AND THEACCURACY OF THE S
     POTS COVERED.";
2083 C=0:S1=0:S2=0:S3=0:FP=1: RETURN

2090 REM  QUIT ROUTINE
2092 POKE -16300,0: TEXT : CALL
     -936
2094 VTAB 10: TAB 18: PRINT "BYE!"

2096 POKE 75,8: POKE 205,8: END

2100 D=1: REM  LEFT FACE
2101 GOSUB HR: GOSUB BU: GOSUB LR:
      RETURN
2170 D=0: REM  NOD
2171 IF F>1 THEN F=0: GOSUB BU:GOSUB
     LF: IF F=0 THEN 2172: GOSUB
     HF:F=0: RETURN
2172 GOSUB HD:F=1: RETURN
4000 REM   PARADE GAME
4010 IF S1 AND S2 AND S3=1 THEN
     4140
4020 COLOR=15:C=C+1: IF C=999 THEN
     4140
4029 PRINT : PRINT "F=FACE FRONT.R=RI
     GHT FACE.L=LEFT FACE M=FWD MAR
     CH. C=CLIMB.      D=DESCEND. H
     =HALT"
4030 VTAB 23: TAB 29: PRINT "SCORE "
     ;C
4040 IF S1=1 THEN 4060
4050 PLOT 10,26: IF X=10 AND Y=11
      THEN GOSUB 4110
4060 IF S2=1 THEN 4080
4070 PLOT 20,38: IF X=20 AND Y=23
      THEN GOSUB 4120
4080 IF S3=1 THEN 4100
4090 PLOT 30,24: IF X=30 AND Y=9
      THEN GOSUB 4130
4100 RETURN
4110 S1=1: RETURN
4120 S2=1: RETURN
4130 S3=1: RETURN
4140 PRINT : PRINT : PRINT "YOUR SCOR
     E OF ";C;: IF C>160 THEN 4240
     : IF C>120 THEN 4220: IF C>
     80 THEN 4200: IF C>40 THEN
     4180
4160 PRINT " WAS EXCELLENT ";B$:
      PRINT "YOUR COUNTRY NEEDS YOU A
     S A DRILL       INSTRUCTOR"
     ;: GOTO 4250
4180 PRINT " WAS GOOD ";B$: PRINT
     "BUT IT IS POSSIBLE TO DO BETTER
     ": GOTO 4250
4200 PRINT " WAS AVERAGE ";B$: PRINT
     "KEEP PRACTICING": GOTO 4250

4220 PRINT " WAS POOR ";B$: PRINT
     "TRY AGAIN": GOTO 4250
4240 PRINT " WAS TERRIBLE ";B$: PRINT
     "YOU JUST HAVE TO BE ABLE TO DO
     BETTER   THAN THIS!!!"
4245 FP=0
4250 PRINT "TO PLAY AGAIN HIT 'P',IF
     FINISHED 'Q'";
4260 RETURN
8000 REM  GRUNT SUBROUTINE
8002 FOR Z=1 TO 10:ZZ= PEEK (-16336
     ): FOR ZZ=1 TO 2: NEXT ZZ: NEXT
     Z: RETURN
8010 PRINT "OUCH! xxxx THAT HURT MY N
     OSE xxxx"
8011 RETURN
9000 REM  INITIALIZATION
9010 POKE 74,0: POKE 75,12
9020 POKE 204,0: POKE 205,12
9030 X=19:Y=19
9040 BU=100:HF=200:HD=300:HL=400
     :HR=500:HB=250:LR=900:LL=950
     :LF=1000
9041 LML=1100:LML1=1150:LMR=1200
     :LMR1=1250
9050 FOR A=20 TO 24: VTAB A: PRINT
     "
             ": NEXT A
9060 A=198: GR
9070 GOTO 20
9999 END

>
```

New products from Scelbi Publications . . . .

Calculating with BASIC
Use your computer to calculate home mortgage payments, interest rates, payback periods and more. Complete routines already worked out for problem solving using BASIC language. Also includes mathematics, finance and statistics, mechanical engineering and electronics. $7.95

Learn Micro-Computers
A new multimedia information package. Includes text ("Understanding Microcomputers") plus high-quality cassette. For the beginner just starting in microcomputers. Covers all the basics quickly, easily, and enjoyably. Companion tape includes chapter-by-chapter synopsis of the book. Great new idea for self-study. $14.95

Personal Information Management System
Increase your personal capabilities . . . save money . . . improve your ability to plan . . . locate important facts quickly . . . eliminate the drudgery of routine chores. These all add up to a better life style for you! A new way of living which can be more pleasant, bring you more happiness and success. Information. That's the key. Your command of information is what gives you the power to succeed. PIMS is a ready-to-use data base management program for computers like the TRS-80, PET, etc. Without any knowledge of programming you can have your computer manage things such as income tax deductions, department store charges, plan your day, vacation, education and important life events. Use it to keep your personal mailing list, schedule your TV viewing time, keep track of personal disbursements, and more. Much more! There's practically no limit to the type of applications. Includes easy-to-read manual and source listing. Get your copy now! $9.95

Microcomputer Potpourri
Data on all the popular chips. Pin connections, diagrams, distinguishing features. All the pertinent information is presented clearly and concisely. Includes glossary. $2.95

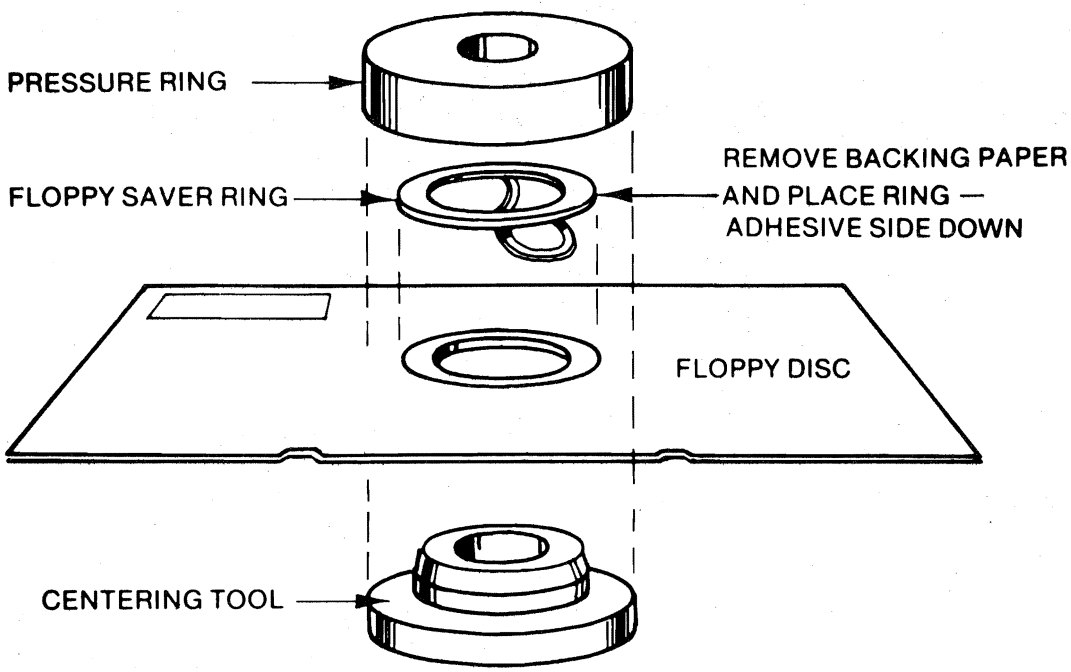
Personal Information Management System
When you're in business you've got a personal stake in how information is managed. That's because information is your key to success. PIMS will allow you to unleash the power of a microcomputer - and you don't have to become a programmer first. Use a computer for accounts receivable . . . accounts payable . . . maintain inventory records . . . run a mailing list . . . keep track of credit charges. These are just a few of the many things you can do with PIMS. It is a ready-to-use data base management program for computers like the Commodore PET and Radio Shack TRS-80. You can define and construct your own data bases. Each records can contain up to ten fields. You can search, list and sort. There's even a command that lets you sum columns of numbers. PIMS provides a valuable introduction to data base management. Complete source listing. $9.95

Calculating with BASIC
Here's a variety of programs in BASIC language to help the businessman, scientist and engineer. Shows how to apply the language to practical problems and equations. Formulas cover calculations of interest, payback periods, mortgage schedules, techniques for extending number of useful digits in monetary calculations using limited BASICs. A variety of electronic applied formulas are programmed. The mechanics chapter covers resultant-force calculations, attractive forces due to gravity, projectile motion prediction and graphing, moments of inertia for T-section, I-section and channel sections. Mathematics chapter includes programs to solve the quadratic formula, general summation formulas such as sum of geometric progression, number of conversion program, algorithms to compute sine, cosine, tangent, log e. $7.95

Understanding Microcomputers
How do you select a microcomputer system for your business? This book is for the person wanting a full understanding of which system is best. It tells how to select a small computer system. Introduces BASIC language programming. Basic instructions for almost every class of microprocessor are illustrated. Written in simple English. Convenient glossary covers all key terms. $9.95

Take My Computer . . . Please!
An uproariously funny book about the true-life misadventures of author Steve Ciarcia and his computer's inability to cooperate. Hardcover. $5.95

TO ORDER TOLL-FREE
1-800-258-1790
(In NH call 673-5144)



Tri-Star Corporation

FLOPPY DISC SAVER

FLOPPY SAVER PREVENTS:

Computer drive's clamping hub from tearing disk's center hole.

Coating removal, scuffing and dimpling when the disk hits the rotating spindle.

Data loss caused by improper rotation of a floppy disk.

Any damage to Floppy Disk's center hole.

FLOPPY SAVER KIT consists of:

Plastic centering tool and pressure ring.

25 7-mil mylar rings with paper protected adhesive backing.

EASILY installed on disks in seconds. Only one side needs the ring.

for 5 1/4" diskettes • Complete Kit $14.95 • Refill $7.95

TO ORDER: CALL TOLL FREE

1-800-258-1790

(In NH call 673-5144)

HARDSIDE

SYNERGISTIC SOFTWARE

PROGRAM LINE EDITOR

Stronger than Integer! More powerful than Applesoft! Faster than a speeding cursor! Able to leap the longest statements in a single move! The Program Line Editor is the most sophisticated program development tool yet for the Apple II computer.
Program Line Editor includes full feature line editors for both Integer and Applesoft programs that render all other techniques obsolete. Program development and modification can now be completed in a tenth the time previously required. The package includes a set of versatile user-definable escape functions which enable you to type commands or sets of commands with only two keystrokes!

The Program Line Editor is available on disk and is compatible with any Apple II or Apple II Plus system with disk drive. **$40.00**

HIGHER TEXT

Higher Text provides the display capabilities you need when the Apple's text output just isn't sufficient. With Higher Text you can print characters in upper case, lower case, upside down, sideways and in any color. If standard characters still aren't adequate, try printing in script, Old English, foreign languages, or special scientific symbology. Define your own specialized or decorative character sets, and easily create professionally customized displays without any hardware modifications.

Higher Text comes on disk and will run on any Apple II or Apple II PLUS system with disk drive. The text generator is in machine language and links easily to the user's application program. **$35.00**

WILDERNESS CAMPAIGN

Free the kingdom of Draconia from the Evil Necromancer! Your party must overcome obstacles, defeat hostile inhabitants, survive natural hazards, explore tombs, temples, castles, and ruins, in search of gold and magical devices. **48K Integer; Cassette $14.95, Disk $17.50.**

Both adventures on disk, 48K Integer, $29.95

DUNGEON CAMPAIGN

Direct an expeditionary force into an underground labyrinth. Catacombs filled with treasures and hazards, poisonous vapors, evil sorcerers, stairways, pitfalls and an incredible assortment of MONSTERS. **16K Integer; Cassette $12.50, Disk $14.95.**

MAILING LIST DATABASE

Entry and storage of a list of names with associated addresses, cities, states, zip codes, phone numbers, comments, and code designations. List, search, edit, print labels; convert prior mailing lists. Up to 225 records can be loaded on line at any time. **48K AppleSoft, Disk $34.50.**

HIGHER GRAPHICS

A complete set of shape generation, manipulation and utilization tools for the programmer. Quickly and easily, you add sophisticated graphic displays and effects to your programs. Shape Maker, Table Combiner, Screen Creator, 4 shape tables, 3 high-res displays, and the machine language graphics routines required to produce high resolution graphics on the Apple II. **48K, Integer Basic, Disk $24.95.**

To order call TOLL-FREE 1-800-258-1790

(In NH call 673-5144)

MAZE SEARCH

MAZE SEARCH is for an 8K Atari with a joystick.

by David Bohlke

Has your mind been uncontrollably wandering lately? Then try putting it back in line with a game that demands intense concentration. In MAZE SEARCH you not only need to control your moving cursor to intercept the 24 targets, but you must also diligently plan ahead to be able to negotiate in and out of the maze. Only the very best players will be able to capture all the targets.

Each game begins with your ATARI generating a random maze. Every section within the maze has only one entrance/exit to the outside corridor – which is your means of accessing other sections of the maze. After the maze is completed, 24 blue targets (blocs) will be randomly placed in the maze. The object of MAZE SEARCH is to move your green cursor and run over the blue targets. Several players can in turn compete to determine which player can intercept the most targets.

There is a time limit to each game which you can adjust in line 600 (variable CT). The time remaining in the game is indicated by a green graphics bar to the right of the screen. Time is most efficiently used when you keep your cursor moving! Each time you intercept a target it will be displayed to the left of the maze. When the game is over, press the fire button on your stick control for another game (use paddle slot 1).

```
1 GOTO 100
2 REM  MAZE SEARCH
3 REM by David Bohlke
```

direction changes for the STICK(0) commands. See page 60 of the Reference manual.

```
5 GOTO 7
6 GOTO 14
7 X=X+1:RETURN
9 GOTO 13
10 GOTO 11
11 X=X-1:RETURN
13 Y=Y+1:RETURN
14 Y=Y-1:RETURN
```

full screen graphics in mode 3

```
100 GRAPHICS 3+16
```

outline of maze

```
120 COLOR 1:PLOT 4,0:DRAWTO 36,0:DRAWTO
36,22:DRAWTO 4,22:DRAWTO 4,0
```

M(200),N(200) will hold the location of every intersection in the maze

```
200 DIM M(200),N(200)
```

K is the intersection counter. M,N is the screen location of the intersection

```
210 K=0:M=18:N=10
```

plot the initial intersection

```
220 PLOT M,N
```

240-390

Loop to construct maze.

Test if next intersection is filled in.

```
240 LOCATE M+2,N,X:IF X=0 THEN 250
242 LOCATE M-2,N,X:IF X=0 THEN 250
244 LOCATE M,N+2,X:IF X=0 THEN 250
246 LOCATE M,N-2,X:IF X=0 THEN 250
```

next intersection is used, so goto 370 to get the previous intersection (M and N) and decrement counter. When K = 0 we'll be back at the beginning and the maze will be completed.

```
248 GOTO 370
```

there is at least one adjacent open intersection, so pick a random direction (D)

```
250 D=INT(RND(0)*4)+2:GOTO D
```

offsets for specific directions (M1 and N1)

```
251 M1=-1:N1=0:GOTO 300
252 M1=0:N1=1:GOTO 300
253 M1=1:N1=0:GOTO 300
254 M1=0:N1=-1
```

check if the intersection at the random direction is open; goto 250 if not

```
300 LOCATE M+M1*2,N+N1*2,X:IF X=1 THEN 2
50
```

plot to the open intersection

```
310 PLOT M+M1,N+N1:PLOT M+M1*2,N+N1*2
```

sound and random color

```
320 SOUND 0,M+N,10,4:SETCOLOR 0,RND(0)*1
5,10:SETCOLOR 4,RND(0)*15,8
```

adjust M,N to the new intersection. Increment K (counter) and M(K),N(K) as the new intersection location; branch to start process again.

```
330 M=M+M1*2:N=N+N1*2:K=K+1:M(K)=M:N(K)=
N:GOTO 240
```

all directions are blocked, so back up to the previous intersection

```
370 M=M(K):N=N(K):K=K-1
```

check for maze completion

```
380 IF K=0 THEN 400
```

not finished, branch to check previous intersection

```
390 GOTO 240
```

clears dimensions (no longer needed)

```
400 CLR
```

player starting location

```
405 M=11:N=11
```

set colors

```
410 SETCOLOR 0,1,10:SETCOLOR 4,4,6
420 SETCOLOR 2,8,8:COLOR 2:PLOT M,N
```

plot 24 target blocks

```
430 SETCOLOR 1,12,8:FOR I=1 TO 24:GOSUB
900:NEXT I
```

500-700 Main Game Loop

check stick; if not pressed branch to 600

```
500 S=STICK(0):IF S=15 THEN 600
```

save position M,N with scratch variables X,Y GOSUB S will be 5-14 for direction offset

```
510 X=M:Y=N:GOSUB S
```

blank players block, locate new position X,Y

```
520 COLOR 0:PLOT M,N:LOCATE X,Y,Z
```

if no move possible (wall), then goto 580

```
540 IF Z=1 THEN 580
```

if hit a target block,then increment counter (HT) and check for end of game

```
550 IF Z=3 THEN HT=HT+1:FOR I=1 TO 20:SO
UND 0,I+50,12,15:NEXT I:IF HT=24 THEN 80
0
```

up until first hit branch to 560

```
551 IF HT=0 THEN 560
```

plot hits on left of maze

```
552 COLOR 3:IF HT/2=INT(HT/2) THEN PLOT
1,23-HT
554 IF HT/2<>INT(HT/2) THEN PLOT 2,23-HT
```

set M,N to new location X,Y

```
560 M=X:N=Y
```

plot player's block

```
580 COLOR 2:PLOT M,N
```

increment time, plot time block on the right of the screen, check for end of game

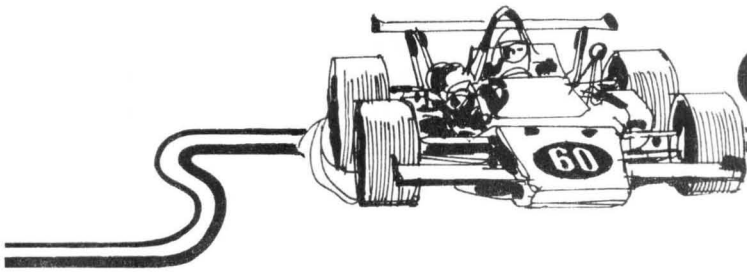
```
600 CT=CT+0.02:COLOR 2:PLOT 38,22-CT:PLO
T 39,22-CT:IF CT>21 THEN PLOT 38,0:PLOT
39,0:GOTO 800
```

a little sound, branch to beginning of game loop

```
700 SOUND 0,2*M+2*N,10,4:GOTO 500
```

game is over; check for the fire button to be pressed to start next game

```
800 IF STRIG(0)=0 THEN RUN
810 SOUND 0,RND(0)*100,10,4:GOTO 800
900 COLOR 3:X=INT(RND(0)*16)*2+5:Y=INT(R
ND(0)*11)*2+1:LOCATE X,Y,Z
ND(0)*11)*2+1:LOCATE X,Y,Z
904 IF Z<>0 THEN 900
906 PLOT X,Y:RETURN
```



GRAND PRIX

GRAND PRIX is an S-80 program for Level II 16K and up.

by Max Chauvet

This Arcade-type game simulates a qualifying session of an international Grand Prix race.

You are in the driver's seat of a powerful Formula One car and you will try to complete one lap of the circuit in the shortest possible time.

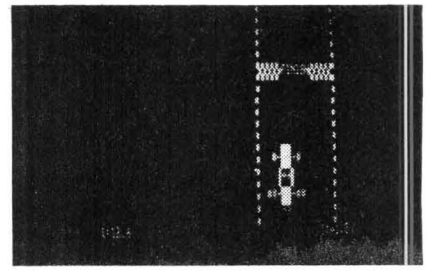
The only controls you have on the car are left and right steering using the arrow keys ← and →

On the straight sections of track you will be going at top speed, but in the curves, whenever you steer, your car will slow down. Therefore, to break the lap record, you will need to avoid all unnecessary steering—driving tightly into the curves. But be careful; if you go over the edge of the track you will hit the guard rail and lose precious seconds.

When you are approaching the finish line, the program will display the time to beat on the left side of the screen and your current time on the right side.

The program will then allow you to re-run on the same circuit or will randomly generate a new circuit.

Note that the car will continue turning as long as an arrow key is held down.



```
10000 ' GRAND PRIX RACING
10010 ' 1980. M.CHAUVET
10020 CLEAR500:DEFINTA-Y:CLS
10025 RPS=0:R0=0:RD=0:EC=0:B=0:KB=0:LC=0:RC=0:TW=2:Z=0:T=0:QZ=17
:UN=1:NL=128
10026 SN=0:LP=0:DX=10:SC=600:T1=0:T2=0
10040 M$=STRING$(255,0):M0$=STRING$(75,0)
10050 ADDR=VARPTR(M0$):IFPEEK(16396)=201 THENPOKE16526,PEEK(AD+1
):POKE16527,PEEK(AD+2)
10052 FST=AD
10054 ZD=PEEK(AD+1)+PEEK(AD+2)*256
10056 IF ZD>32767 ZD=ZD-65536
10058 AD=ZD
10060 IFPEEK(16396)<>201THEN DEFUSR0=AD:CMD"T"
10070 CAR=ADDR+182
10080 EC=16: DIM LAP(50)
10090 KBOARD=14400 : TWO=2 : BL=32
10100 B$=STRING$(8,24) : C$=CHR$(26)
10110 A$=STRING$(2,176)+LEFT$(B$,4)+C$+"$-"+STRING$(2,191)+"-$"+
      LEFT$(B$,5)+C$+CHR$(170)+CHR$(93)+CHR$(94)+CHR$(149)+LEFT$
      (B$,6)+C$+"##-"+STRING$(2,191)+"-##"+LEFT$(B$,5)+C$+CHR$
      (34)+CHR$(34)
10200 ' SCROLLING ROUTINE
10210 DATA CD7F0A7D00018E40C5FD21BD00FD09DD21B600DD092600DD6E000
1
10220 DATA 003C09C1B72802FDE9114003E519E5D101400009EB0E10ED
10230 DATA B0110003E1E519E5D101400009EB0E38EDB011BF02E1E519E5D1
10240 DATA 01400009EB0E39EDB0117F02E1E519E5D101400009EB0E3AEDB0
10250 DATA 113D02E1E519E5D101400009EB0E3CEDB011EE02FDE1FDE5FD19
10260 DATA FD7E00FD7710FD7E01FD7711FD7E18FD7708FD7E19FD7709113A
10270 DATA 022600DD6E0019E5FDE1E119E5D101400009EBFDE5C1EDB83E00
10280 DATA C900000000000000
10290 '  MOVE CAR
10300 DATA DD213101DD093D20313E05117C0219E5D11313E50600DD4E00EDB
823E5FDE1FD4610
10310 DATA FD7000FD4611FD7001E13DC80600DD4E0109DD23DD2318D53E051
10A00DD19117B02
10320 DATA 19E5D11B1BE50600DD4E00EDB0011200ED42E5FDE1FD4600FD701
0FD
10330 DATA 4601FD7011E13DC80600DD4E0109DD23DD2318D1
10340 DATA 0242063F044208400500
10350 DATA 023E0641043E08400500
10360 DATA END
10370 READ D$ : IF D$="END" THEN 10440 ELSE GOSUB 15000
10380 FORI=1TOLEN(D$)STEP2
10390 D=ASC(MID$(D$,I,1)):D1=ASC(MID$(D$,I+1,1))
10400 IFD>57THEND=D-7
10410 IFD1>57THEND1=D1-7
10420 D=(D-48)*16+D1-48 : POKE ADDR,D : AD=AD+1
10430 NEXT I : GOTO 10370
10440 DATA 0,1,1,2,3,3,2,1,1,0
10450 FORI=1TO10:READC(I):NEXTI
10460 POKE ZD+6,PEEK(FST+1)
10470 POKE ZD+7,PEEK(FST+2)
10500 '  CIRCUIT SET UP
10510 TM=30+RND(20): TB=TM*10.1: TK=TM-5: TL=TM-1: CN=0
10520 FOR I=2 TO TM-2: DIR=RND(3)-2:CN=CN+DIR:IFABS(CN)>1THENCN=
CN-2*DIR:DIR=-DIR
10524 IF PRV=0 PRV=DIR
10526 IF DIR THEN IF DIR=PRV THEN TB=TB+14:PRV=DIR ELSE TB=TB+6:
PRV=DIR
10530 LAP(I)=DIR:NEXTI
10540 LAP(1)=0:LAP(TM)=0:LAP(TM-1)=0:CLS
10542 PRINT@466,"THE CIRCUIT IS ";:PRINTUSING"#.## MILES LONG";T
M/20;
10545 PRINT@530,"THE LAP RECORD IS ";:T1=TB/600:T2=TB-T1*600:PRI
NTUSING"##:##.#";T1,T2/10;:TB$=""
10547 FORI=548TO554:TB$=TB$+CHR$(PEEK(15360+I)):NEXT
10548 ME$=TB$:M=548:GOSUB21000:FORI=1TO200:NEXTI:CLS
10550 CH$=CHR$(28)+CHR$(255)
10570 CLS: PRINT@671,A$;
10580 POKE CAR,36: RCRASH=15360+733+EC+1: LCRASH=RC+5-2*EC-1
10590 RPS=15384: ROAD=132: RD=13
10600 FOR LP=1 TO TM
10610 SN=LAP(LP): IF SN THEN RO=132:RD=0 ELSE RD=13
10620 FORI=1TO10:RPS=RPS+C(I)*SN:Z=USR(0):PRINTCH$;:POKE RPS,RO:
POKE RPS+EC,RO:RO=RO+RD:RD=-RD: B1=B:B=PEEK(KB):IF SN POKE RPS-U
N,NL:POKE RPS+QZ,NL
10625 IFB=0THEN10800
10630 T=T+2 :IFB=32THEN10700
10650 Z=USR(1):POKE CAR,PEEK(CAR)+TWO:LC=LC+TWO:RC=RC+TWO:GOTO10
710
10700 Z=USR(2):POKE CAR,PEEK(CAR)-TWO:LC=LC-TWO:RC=RC-TWO
10710 IFB1>0THENIFB1<>BTHEN12000
```

continued on page 78

GRAND PRIX Take-A-Part

Let's first look at the statement line numbers in the program: you will notice that they are all 5 digits long starting at 10000. This is not due to a special programming technique; actually I wrote this program on top of an earlier version in order to be able to refer some of the original statements while entering the new ones. The original version was eventually deleted.

The optical effect of the car advancing is created by simulating the track receding. This is done by the first part of the machine language routine (Basic being too slow for video games) which scrolls the screen down around the car. The second part of this routine shifts 2 positions right or left every character that forms the car drawing, whenever the arrow keys are pressed. All other functions are in straightforward BASIC.

The car leaving the track is detected by PEEKing the video memory 16 positions left of the right frontwheel and 16 positions right of the left front wheel. In this case 16 is the width of the track. Because track format is identical on both sides of the car, if the right front wheel runs over a dot, then the "crash" spot 16 positions to its left will also contain a dot. The same holds for the left wheel in the opposite direction. This was the tricky part. The rest is easy. Score counter is incremented by 1 with every program cycle, by 2 with every car shift and by 20 with every crash. Final score is compared with a theoretical record score calculated when generating the circuit layout, i.e. the succession of curves and straight lines.

The structure of the main program cycle may be summarized as follows:

(1) Scroll the screen around the car down one line.
(2) Insert a new line at the top -Curves are formed by progressively shifting the position of dots on this line.
(3) Check the arrow keys. If pressed, shift car and update score.
(4) Check crash spots. If not blank, display crash effects and update score.
(5) Return to step (1)

Let's now look in more detail at each of the sections in the program.

LINES 10000-10110 -VARIABLE DEFINITION

In order to speed up execution, an initial value is assigned to the most frequently used variables. Because of this, the BASIC interpreter places them at the beginning of the variable names list, thus reducing search time when referenced later in the program. This is why in statement 10025 several variables are set to zero. Other variables used here are:

M$,MO$
: contain the machine language routine.

ADDR
: absolute starting address of the routine.

EC
: track width

BL
: blank character 32

NL
: null character 128

LAP(50)
: sequence of curves and straights forming the circuit. Each element represents 10 lines on the screen and may take the following values:
-1 for a left curve
0 for a straight line
1 for a right curve
Note that the screen width limits to 2 the number of successive curves in the same direction

KBOARD
: Keyboard address. When pressing the arrow keys, PEEKing this location will return the value 32 for the left key and 64 for the right key.

A$
: Car drawing. It lays over 5 lines by use of the cursor back (decimal 24) and cursor down (decimal 26) characters.

LINES 10200-10430 - MACHINE LANGUAGE ROUTINE

The DATA statements contain the hexadecimal form of the routine. Lines 10370-10430 convert every 2 characters to their decimal equivalent value which is then POKEd in the reserved string variable space. In order to execute this code, the interpreter must be given a starting address. This is done under Level II BASIC by poking this address into memory locations 16526 and 16527. Under DOS, a DEFUSR statement is required.

If the available memory exceeds 16K, the routine will be located past the 32767 address because string area is reserved in high RAM. A small problem arises in this case because POKE statements require the address to be in the [-32768, +32767] range; when it exceeds the upper bound the address must be converted to its two's complement form by subtracting 65536 from it. In this manner, 32768 will be converted to -32768 and 65535 to -1.

Simple down scrolling is achieved by transferring the video memory block corresponding to the first 15 lines to a block starting 64 positions higher. A powerful feature in the Z-80 processor allows transfer of blocks of characters with a single instruction: you need only indicate the source block address, the destination block address and the number of characters to be transferred. In our case, however, the area occupied by the car must not be touched. Transfer must therefore be done in several steps.

Note that when the source and destination blocks overlap, as is the case here, transfer must be done from end to start. Referring to diagram A, the order of transfer is as follows:

1) Move block	A-A' to block	a-a'
2)	B-B'	b-b'
3)	C-C'	c-B'
4)	D-D'	C-d'
5)	E-E'	e-D'
6)	F-F'	E-f'
7)	G-G'	g-g'

continued on page 78

CONCENTRATION

CONCENTRATION is an Integer BASIC program for the Apple and requires at least 16K.

by Harris Kirk

Test your powers of concentration!! You are confronted by 36 closed doors, behind which are 18 pair of randomly- distributed pictures.

Just as in the television game, the object is to find the matching pairs by remembering their location in previous turns.

The game is very unique with respect to its detailed graphics. It is an excellent party game for 2 or more players, although it can also be played as 'solitaire'. It is challenging and just plain fun!

INSTRUCTIONS

There are 18 pairs of pictures randomly hidden behind 36 doors.

The object of the game is to find more matching pairs than your opponent.

Only two doors may be opened at a time.

If you have chosen a matched pair, the doors remain open, and you are awarded five points and another turn.

If you fail to find a match, the doors close and you lose your turn!!

Use game controls to choose a door:

1) Choose the column,

2) Hold the button down firmly,

3) Choose the row,

4) Release the button.

The game ends when all doors are open.

OPTIONAL

Player variations:

1) Game can be played alone using paddle 0 by pressing return for the second player.

2) Several people can play as two teams (enter team as one player).

MAKING YOUR OWN PICTURE MODIFICATIONS:

All pictures are plotted by use of subroutines, each one starting at even multiples of 10. i.e., 20, 40, 60, etc.

1) Find a picture that you would like to replace by listing lines 20 to 379.

2) Delete the appropriate line numbers: e.g. DEL 80,98 (note that line 99 is needed to return to the main program loop and line 100 is the start of the next picture).

3) Give your picture a name and enter the command A$="NAME": GOSUB 820 at the starting point of the subroutine (line 80 in the previous example).

4) The top left corner of the picture frame is X,Y. The maximum horizontal direction is X+4 and the maximum vertical direction is Y+4. As an example: to place orange dots at the four corners of the picture frame, enter a line number (between 81 and 88 in the example) followed by COLOR=9: PLOT X,Y: PLOT X,Y+4: PLOT X+4,Y: PLOT X+4,Y+4. Any PLOT, HLIN, or VLIN will appear correctly, provided they are within the limits previously mentioned.

5) You can use any variable starting with I, J, K, or L and not affect the main program.

6) You may also add sound to your picture by inserting a line number followed by POKE 0,P: POKE1,D: CALL 2. P is a number between 0 and 255 (low number is a high pitch) D is a number between 0 and 255 (low number is a short tone) The command CALL 2 starts a fast machine language routine that clicks the Apple speaker. It is poked into memory in lines 2030 and 2040. Try the following: FOR I=100 to 10 step -1:POKE 0,I: POKE 1,10: CALL 2 NEXT I (on the next line number).

If you wish to view your picture before actually playing the game, type the following: (each followed by a 'return')

DIM A$(20)
GR
GOTO XX (where XX is the first line number of your picture, which must be an even multiple of 10)
The bad return error will not appear when the program is run.

8) Be sure to: check that the last line in your routine is RETURN. SAVE your new version.

An example of a picture subroutine

Picture: Lighthouse

Line #	Function
40	Put 'lighthouse' into A$; print A$ using subroutine at 820
41	Plot lighthouse and rocks
42	Do the following lines (up to line 55) three times
43	Plot the lighthouse beam, using black
44	Plot the lighthouse window,using grey
45	Wait
46	Plot the beam, using white; plot the window, using yellow
54	Wait
55	End of for-next loop
57	Plot beam, using black; plot window, using grey
59	Return to the main program loop

VARIABLES

Arrays:

SC	Player's game score
SUM	Player's total score
CH	Status of door (open or closed)
G	Contains 2 each of picture numbers (0-17)
PIX	Contains 2 each of picture numbers (0-17) randomly distributed.
W	Picture numbers of the 2 doors chosen per turn
C	Colors of the door rows
K	Colors of the books in the 'bookshelf' subroutine

Strings:

A$	Picture name and temporary storage for printing a string with sound
Y$	Name of first player
Z$	Name of second player
H$	Holds response to program questions

Single Variables:

X	Horizontal (X) coordinate of graphics grid
Y	Vertical (Y) coordinate of graphics grid
Z	Counter in a pause loop
I,- J,K,L	Temporary counters
II	Holds random position of candle flame
X- N,YN	New x and y coordinates of goldfish position
X- O,YO	Old x and y coordinates of goldfish position
ND	Number of door choice (1 or 2)
M- TCH	Number of matched pairs
TRY	Number of turns (when playing as 'solitaire')
P	Player number (0 or 1)
N	Set to 0 if played as 'solitaire', otherwise set to 1
COL	Column number (0-5)
C1	Column number of first door choice
ROW	Row number (0-5)
R1	Row number of first door choice
D	Door number (0-35)
D1	Door number of first door choice
X1,Y1	X and y coordinates of first door choice
PP	Pitch of tone poke during door closings

```
 0 REM   CONCENTRATION
 1 REM   BY HARRIS KIRK
 2 REM   MAY,1980
 3 REM
 4 DIM SC(2),CH(36),K(3),PIX(40
   ),W(5),G(40),C(10)
 6 DIM A$(25),Y$(40),Z$(40),H$
   (5)
 8 DIM SUM(2):SUM(0)=0:SUM(1)=
   0
16 GOSUB 2800
17 GOTO 1000
18 REM   PICTURE SUBROUTINES
20 A$="CANDLE": GOSUB 820
21 COLOR=8: HLIN X,X+3 AT Y+4:
    PLOT X+2,Y+3: PLOT X+4,Y+3

22 COLOR=15: PLOT X+2,Y+2: COLOR=
   13: PLOT X+2,Y+1
23 II=X+2
24 FOR I=1 TO 14
26 COLOR=0: PLOT II,Y
28 II= RND (3)+1+X
30 COLOR=13: PLOT II,Y: FOR Z=
   1 TO 60: NEXT Z
32 NEXT I
39 RETURN
40 A$="LIGHTHOUSE": GOSUB 820
41 COLOR=1: VLIN Y,Y+3 AT X+2:
    COLOR=5: HLIN X+1,X+3 AT Y+
   4
42 FOR J=1 TO 3
43 COLOR=0: HLIN X,X+4 AT Y+1
44 COLOR=5: PLOT X+2,Y+1:
45 FOR Z=1 TO 175: NEXT Z
46 COLOR=15: HLIN X,X+4 AT Y+1
   : COLOR=13: PLOT X+2,Y+1
54 FOR Z=1 TO 175: NEXT Z
55 NEXT J
57 COLOR=0: HLIN X,X+4 AT Y+1:
    COLOR=5: PLOT X+2,Y+1
59 RETURN
60 A$="PONY": GOSUB 820
62 COLOR=15
64 HLIN X+1,X+3 AT Y+2: VLIN Y,
   Y+3 AT X+3
66 PLOT X,Y+1: PLOT X+1,Y+3: PLOT
   X+3,Y+3: PLOT X+4,Y
67 COLOR=8: PLOT X+2,Y+2
68 FOR J=1 TO 6
70 COLOR=15: PLOT X,Y+4: PLOT
   X+4,Y+4
71 FOR Z=1 TO 65: NEXT Z
72 COLOR=0: PLOT X,Y+4: PLOT X+
   4,Y+4
73 COLOR=15: PLOT X+2,Y+4
74 FOR I=1 TO 3: POKE 0,191: POKE
   1,14: CALL 2: FOR Z=1 TO 10
   : NEXT Z,I
75 COLOR=0: PLOT X+2,Y+4
76 NEXT J
77 COLOR=15: PLOT X,Y+4: PLOT
   X+4,Y+4
79 RETURN
80 A$="R R CROSSING": GOSUB 820
82 COLOR=15
84 PLOT X,Y: PLOT X+1,Y+1: PLOT
   X+2,Y+2: PLOT X+3,Y+3: PLOT
   X+4,Y+4
86 PLOT X,Y+4: PLOT X+1,Y+3: PLOT
   X+2,Y+2: PLOT X+3,Y+1: PLOT
   X+4,Y
88 FOR J=1 TO 4
90 FOR I=1 TO 3 STEP 2
92 COLOR=1: PLOT X+I,Y+2: POKE
   0,114: POKE 1,133: CALL 2
94 COLOR=0: PLOT X+I,Y+2
96 NEXT I,J
99 RETURN
100 A$="BOOKSHELF": GOSUB 820
102 COLOR=8: HLIN X,X+4 AT Y+3:
     PLOT X+1,Y+4: PLOT X+3,Y+4

105 COLOR=5: PLOT X,Y+2: PLOT X+
   4,Y+2
110 K(1)=7:K(2)=1:K(3)=15
111 FOR I=1 TO 3: COLOR=K(I): VLIN
   Y,Y+2 AT X+I: NEXT I
119 RETURN
120 A$="QUILT": GOSUB 820
130 FOR I=X TO X+4
132 FOR J=Y TO Y+4
134 COLOR= RND (15)+1
135 PLOT I,J
136 NEXT J,I
139 RETURN
140 A$="GOLDFISH TANK": GOSUB 820

142 COLOR=15: VLIN Y,Y+4 AT X: VLIN
   Y,Y+4 AT X+4: HLIN X,X+4 AT
   Y+4
144 COLOR=7: FOR I=0 TO 3: HLIN
   X+1,X+3 AT Y+I: NEXT I
148 FOR J=1 TO 15
150 COLOR=9:XN=XO+ RND (3)-1: IF
   XN>3 OR XN<1 THEN 150
152 YN=YO+ RND (3)-1: IF YN>3 OR
   YN<0 THEN 152
154 PLOT XN+X,YN+Y
156 FOR Z=1 TO 60: NEXT Z: IF J=
   15 THEN 158: COLOR=7: PLOT
   XN+X,YN+Y
157 XO=XN:YO=YN
158 NEXT J
159 RETURN
160 A$="HEART": GOSUB 820
163 COLOR=1
```

continued on page 74

STRATOBLASTER OUTPOST

STRATOBLASTER OUTPOST requires an Atari with at least 8K, and a joystick.

by David Bohlke

ALERT! ALERT! ALERT! STRATOBLASTER OUTPOST No. 6: A Slipton invasion force has been sighted moving into your sector. We don't want you to panic – but you are Earth's first defense!

From past experience you know that a direct hit is needed to vaporize the Sliptons. Yet, you have just 75 shots in your mighty STRATOBLASTER cannon to put the Slipton force in disarray. Aiming the STRATOBLASTER is easy – you need only push the control stick (paddle slot No. 1) to the left or right, then press the Fire button. Speed is crucial, since you must fire before the Sliptons move out of your sector. Luckily, they don't all come at the same time!

Your marksmanship is well charted. During play, a red bar at the top of the screen will display the number of hits, and a blue bar will show the number of misses. A green bar will indicate the number of invaders that have entered your sector. When the green or blue bar reaches the right edge of the screen, the invasion will have ended.

When the game begins, you have a choice of three possible play modes. In a Vertical Invasion, the Sliptons will move from the top of the screen to the bottom. During a Horizontal Invasion, the saucers will progress either left or right on the display. And for a Random Invasion, the targets will move in varied directions. At the completion of the game, your hits, misses, and a percentage will be displayed – so several players can compete for most hits and/or the highest percentage.

```
1 GOTO 1000
2 REM  STRATOBLASTER  OUTPOST
3 REM by David Bohlke, Cosson  IA
```

5-15

subroutine calls for STICK (0) command directions. (Atari BASIC allows a variable to call a subroutine)

5,6,7 move cannon to right

```
5 GOTO 7
6 GOTO 7
7 D=2:GOSUB 50:RETURN
```

9-11 move cannon to left

```
9 GOTO 11
10 GOTO 11
11 D=-2:GOSUB 50:RETURN
```

13-15 no action

```
13 RETURN
14 RETURN
15 RETURN
```

50-56 plots cannon

```
50 COLOR 4:PLOT 39,47:DRAWTO C,R:DRAWTO
41,47
52 C=C+D:IF C<30 THEN C=30
54 IF C>50 THEN C=50
56 R=36+ABS(40-C):COLOR 1:PLOT 39,47:DRA
WTO C,R:DRAWTO 41,47:RETURN
```

100-120 initialize

```
100 GRAPHICS 5+16:C=40:R=36:TM=1:H=0:MT=
0:GOSUB 50
110 SETCOLOR 4,0,6:DIM Q(5),R(5):FOR I=1
 TO 5:Q(I)=0:R(I)=0:NEXT I
120 SETCOLOR 0,4,10:SETCOLOR 2,7,10
```

200-350 main game loop

200

get STICK direction from joystick controller, branch to 300 if no FIRE

```
200 S=STICK(0):GOSUB S:IF STRIG(0)=1 THE
N 300
```

210 blue background, sound for FIRE

```
210 SETCOLOR 4,7,9:FOR I=33 TO 43:SOUND
0,I,10,14:NEXT I
```

220-240 direction for laser fire

```
220 Y=4:X=(C-36)*8+8
230 IF C<36 THEN X=1:Y=(35-C)*8
240 IF C>44 THEN X=78:Y=(C-45)*8
```

250 plot laser beam

```
250 COLOR 2:PLOT C,R:DRAWTO X,Y
```

260 check 5 targets for hit

```
260 U=0:FOR I=1 TO 5:FOR J=Q(I)+1 TO Q(I
)+3
```

262 branch to 280 if hit

```
262 LOCATE J,R(I),U:IF U>0 THEN T=I:GOTO
 280
```

264 else add 1 to miss counter, check 75 for end of game

```
264 NEXT J:NEXT I:MT=MT+1:IF MT>75 THEN
600
```

266 plot miss bar, GOTO 290

```
266 COLOR 3:PLOT MT,2:GOTO 290
```

280 get new target (after hit)

```
280 COLOR 4:GOSUB 920:Q(T)=0
```

282 red background, hit blast, add to hit count, zero target

```
282 SETCOLOR 4,4,11:FOR I=33 TO 65:SOUND
 0,I,2,14:NEXT I:H=H+1:Q(T)=0
```

284 plot hit bar

```
284 COLOR 1:PLOT H,0
```

290 black background, cancel out laser beam

```
290 SETCOLOR 4,0,6:COLOR 4:PLOT C,R:DRAW
TO X,Y
```

targets move only 70% of the time

```
295 IF RND(0)<0.3 THEN 200
```

300,305 increment time, plot time bar, check for end of game

```
300 T=T+1:IF T>5 THEN T=1
305 COLOR 2:PLOT TM,1:IF TM>75 THEN 600
```

get new target if one leaves screen

```
310 IF Q(T)=0 THEN GOSUB 900:TM=TM+1:GOT
O 200
```

315 saucer sound, black out saucer

```
315 SOUND 0,Q(T)+R(T),6,4:COLOR 4:GOSUB
920
```

320-335 new saucer location

```
320 U=1:IF T/2=INT(T/2) THEN U=-1
323 IF TD=1 THEN 335
330 R(T)=R(T)+INT(RND(0)*3)+1
333 IF TD=2 THEN 340
335 Q(T)=Q(T)+INT(RND(0)*5+1)*U
```

340 Test for new location off screen

```
340 IF Q(T)<2 OR Q(T)>74 OR R(T)>44 THEN
 Q(T)=0:GOTO 200
```

350 plot saucer in new location, GOTO 200 to continue

```
350 GOSUB 910:GOTO 200
```

600-612 end of game statistics

```
600 GRAPHICS 5+32:PRINT "HITS ";H,"MISSE
S ";MT,INT(H/(H+MT)*100);"%"
605 PRINT :PRINT " HIT   FIRE for next g
ame!";
607 FOR I=1 TO 1000:NEXT I
610 IF STRIG(0)=0 THEN RUN
612 SOUND 0,RND(0)*250,10,6:GOTO 610
```

900 new target location

```
900 QX(T)=INT(RND(0)*60)+3:R(T)=INT(RND(0
)*20)+4
```

910 new target color

```
910 COLOR INT(RND(0)*3)+1
```

920 plot new target

```
920 A=QX(T):B=R(T):PLOT A,B:PLOT A+4,B:PL
OT A+1,B-1:DRAWTO A+3,B-1:PLOT A+1,B+1:D
RAWTO A+3,B+1:RETURN
```

1000- enter invasion options

```
1000 GRAPHICS 0:PRINT " STRATOBLASTER
OUTPOST ":PRINT :PRINT :POKE 752,1
1001 FOR I=1 TO 333:NEXT I
1002 PRINT "PRESS  STICK  FOR OPTIONS,":
PRINT :PRINT "PRESS  FIRE  TO ENGAGE ?"
1003 IF STRIG(0)=0 THEN GOTO 100
1004 POSITION 4,14:IF STICK(0)=15 THEN G
OTO 1003
1005 TD=TD+1:IF TD>3 THEN TD=1
1006 GOSUB TD+1006:GOTO 1003
1007 PRINT #6;"HORIZONTAL INVASION.":RET
URN
1008 PRINT #6;"VERTICAL INVASION.  ":RET
URN
1009 PRINT #6;"RANDOM INVASION.    ":RET
URN
```

KEYSTROKE KARNIVAL

KEYSTROKE KARNIVAL is an Atari Program for 8K and up.

by James Garon

As an alternative to OPEN, GET etc. it is possible to check for keyboard input during program execution by PEEKing at location 764. If no key has been pressed, PEEK(764) will yield 255. If a key has been pressed, the following table shows what PEEKing will reveal at 764:

KEY	PEEK(764)	KEY	PEEK(764)	KEY	PEEK(764)
A	63	O	8	3	26
B	21	P	10	4	24
C	18	Q	47	5	29
D	58	R	40	6	27
E	42	S	62	7	51
F	56	T	45	8	53
G	61	U	11	9	58
H	57	V	16	0	50
I	13	W	46	+ (LEFT)	6
J	1	X	22	*(RIGHT)	7
K	5	Y	43	– (UP)	14
L	0	Z	23	=(DOWN)	15
M	37	1	31	RETURN	12
N	35	2	30	ESC	28

To use this information in your own programs, you will want to POKE 764 with 255 AFTER your program discovers that a key has been pressed. This will reset location 764 until the next key is pressed.

EXAMPLE:

```
200 P=PEEK(764)
210 IF P=255 THEN 200: REM KEEP LOOPING UNTIL A
    KEY IS PRESSED
220 POKE 764,255:REM RESET LOCATION 764 FOR NEXT
    KEY
230 IF P=6 THEN? "LEFT"
240 IF P=7 THEN? "RIGHT"
250 IF P=14 THEN? "UP"
260 IF P=15 THEN? "DOWN"
270 GOTO 200
```

We have here the beginnings of "The Poorman's Joystick Routine". If you replace the PRINT statements in lines 230-260 with GOSUBS to the appropriate subroutines, you can use the arrow keys (no need to press CTRL) to control the direction of a moving object on the screen, to change the SOUND from your TV speaker and whatever else your programming heart desires.

ATARI DOUBLES PRODUCT LINE
continued from page 19

232C ports, one of which can also be used for 20 ma current loop, and an 8-bit parallel printer port. It features full duplex operation, programming baud rate from 75 to 9600 baud, choice of stop bits and parity, and even automatic telephone answering with appropriate modem. Different pictures of the interface have been released. The unit on top of the disk drives in the second picture, and not the black box in the first picture, is apparently the production model. The Atari 830 Modem is apparently the popular Cat ™ Modem.

Among the software packages announced are a series of accounting packages for small business; a series of investment analysis programs; language lessons in French, German, and Spanish; and Space Invaders (™ of Taito Corp.), all on cassette.

ATARI EDITOR/ASSEMBLER
continued from page 19

space. The TAB key is useful for field separation.

Op Codes and Operands

Standard 6502 Operation code mnemonics are used, and are listed in the manual. Among the operands, $ indicates a hex number, and # an immediate operand. Absolute operands are evaluated as 16 bit numbers. Absolute indexed operands use a comma and X or Y after a 16 bit number or variable. Indexed indirect operands are written with an 8 bit number or variable, X is always used and the whole operand is enclosed in parentheses. Indexed page zero operands are written with an 8 bit number or variable, a comma, and X or Y. String operands are enclosed in quotes and use specified directives such as .byte.

Commands

The command SIZE tells you the starting address of the current line buffer, the starting address of the edit text buffer, and the highest address in memory. LOMEM allows you to change the location of the buffers. Other commands include LIST, PRINT, FIND, DEL (Delete), NUM (Automatic line numbering), REN (renumber), REP (Replace), SAVE, LOAD, ENTER, and ASM (Assemble). Programs can be assembled to the edit text buffer, screen printer, or disk as well as the indicated memory location.

Pseudo Ops

A number of directives or pseudo operations are available. The required directive * = allows you to set the memory location at which your program will be assembled. The other required directive is .END, which identifies the end of your program to the assembler. Most of the directives are preceded by a period, and include:

.OPT (Option: List, no list, object, no object, errors, no error messages, page spacing, or no page spacing)

.PAGE (print page numbers)

.TITLE (print title on each page of listing)

.TAB (set field spacing)

.BYTE (reserve space in memory of an operand)

.DBYTE (reserve 2 bytes of memory for each expression in the operand, stored most significant byte first)

.WORD (same as DBYTE, except that low order byte is stored first)

= (assign a value to a label)

Demonstration

Here is a sample program to show the format of an assembled program. All it does is add 2 + 2 and store the result.

```
10        *=$3002 ;START OF PROGRAM
20 SUM=$3000      ;MEMORY FOR SUM
30 START  LDA #2  ;FIRST NUMBER
40        ADC #2  ;SECOND NUMBER
50        STA SUM ;STORE RESULT
60        .END

ASM
0000          10          *=   $3002
 ;START OF PROGRAM
3000          20 SUM      =    $3000
 ;MEMORY FOR SUM
3002 A902     30 START    LDA  #2
 ;FIRST NUMBER
3004 6902     40          ADC  #2
 ;SECOND NUMBER
3006 800030   50          STA  SUM
 ;STORE RESULT
3009          60          .END
```

My main disappointment with the assembly process is that a symbol table is not generated at the end of the program.

The Monitor

The DEBUG program is called from the EDIT program by the command BUG. It allows the following commands:

DR (display registers)

CR (change registers)

Dxxxx (Display memory at xxxx in hex)

Cxxxx (Change memory at xxxx)

Mxxx yyyy,zzzz (Move memory between yyyy and zzzz to new location starting at xxxx)

Vxxxx yyyy,zzzz (Compare memory between yyyy and zzzz with memory starting at xxxx)

Lxxxx (Disassemble memory starting at xxxx)

Txxxx (Trace execution of memory with disassembly and register contents starting at xxxx)

Sxxxx (Single step through memory execution starting at xxxx)

Gxxxx (Execute program, beginning at xxxx)

X (Return to edit mode)

A (Transfer to mini assembler)

The mini assembler allows you to assemble one instruction at a time until you press return at the beginning of an instruction or type .END to return to DEBUG.

Other features

Eighteen error messages are given to assist in finding difficulties in your programs, in addition to the normal DOS error messages that function during disk I/O operations.

Only a minimal selection of operating system memory locations was provided in the preliminary manual for the editor assembler. However, more are listed in the Basic reference manual, and there is an operating system reference manual in preparation with extensive documentation. I have already obtained more than ten pounds of printed material, and have been very impressed by the speed with which Atari is assembling complete documentation for their computer system.

In conclusion, Atari has provided an excellent tool for assembly language development. It should be well worth the $59.95 price, especially considering the convenience of a ROM cartridge.

HOW MUCH COMPUTER?
continued from page 15

with accessories for special applications. If you are serious about making money from your computer, you will want to add a printer. Inexpensive printers like the Eaton LRC or Atari 820 are good for making line listings to improve your productivity as a programmer. A slightly better printer like the Centronics 737 (Atari 825 or Radio Shack Line Printer IV) is good for printing reports, minor word processing, mailing labels, and similar applications. If you want true word processing, then you need a letter quality printer such as the NEC Spinwriter. Other accessories might include an interface and modem for time sharing or electronic mail, a plotter, a control unit for connecting the computer to an appliance or lab instrument, or any of several dozen other items, depending on the application.

You will also need software for the application you desire. SoftSide will provide you with plenty of good games, and also teach you the skills necessary to write your own application programs. In addition, we sell programs ready for many applications.

How much should you budget for software? That depends on your application, but the general rule of the last ten or fifteen years has been that software costs as much as the rest of the system. If you are doing a lot of your own programming, the same rule still holds, when you consider your time at a fair value. That is the shocker for most people, especially those such as my correspondent with the new Apple.

In summary, here is a chart listing the items needed for a complete general purpose computer system based on our three favorite computers. by George Blank

Radio Shack TRS-80	Apple II	Atari 800
	General Purpose cassette based system	
TRS-80 computer	Apple II	Atari 800
Level II BASIC	Cassette Player	Cassette Player
16K memory	Video Modulator	16K memory
Color TV	Color TV	
	General purpose disk based system	
TRS-80 Computer	Apple II	Atari 800
Level II BASIC	48K memory	Color TV
Expansion Interface	Video Modulator	32-40K memory
32K memory	Color TV	Disk Drive
Disk Drive	Disk Drive	

Along with this, you will also need software and any special devices such as a printer.

630 Commence first sailplane launch by printing cockpit using S/R.

```
630 CLS:PRINT"YOUR GO ";B$(B)
640 FORN=1TO1500:NEXT
650 CLS
660 GOSUB3690
670 '
680 '
```

690 Calculate any change in height due to change in velocity of sailplane. (So called total energy effect - or converting potential energy into kinetic energy or the other way around).

```
690 Z2=(W(B)-S(B))*10:H(B)=H(B)+Z2
700 T3=S(B)
```

710-720 Calculate height loss assuming a glide thru a static air mass.

```
710 IFS(B)<50THENT3=60
720 T4=T3^2/50:T3=0:H(B)=H(B)-T4
730 IFD(B)=0THEND(B)=360
```

740-800 Calculate, from information on compass course, the new position of sailplane using polar - rectangular conversion.

```
740 T5=(D(B)/360)*6.28318
750 '
760 D(B)=INT(D(B)):IFD(B)>90THEN780:IFD(B)>180THEN790:IFD(B)>270
THEN800
770 T1=SIN(T5)*(S(B)/12):X(B)=X(B)+T1:T2=T1/(TAN(T5)+.01):Y(B)=Y
(B)+T2:GOTO810
780 T5=T5-1.5708:T1=SIN(T5)*(S(B)/12):Y(B)=Y(B)-T1:T2=T1/TAN(T5)
:X(B)=X(B)+T2:GOTO810
790 T5=T5-3.142:T1=SIN(T5)*(S(B)/12):X(B)=X(B)-T1:T2=T1/TAN(T5):
Y(B)=Y(B)-T2:GOTO810
800 T5=T5-4.7120:T1=SIN(T5)*(S(B)/12):Y(B)=Y(B)+T1:T2=T1/TAN(T5)
:X(B)=X(B)-T2
```

810 Apply wind drift.

```
810 IFA3=<2THENX(B)=X(B)+.41ELSEIFA3>2ANDA3<5THENX(B)=X(B)+.94EL
SEIFA3>4THENX(B)=X(B)+2.0
```

820-850 Sort through thermals and locate nearest one to current position of sailplane.

```
820 FORN=1TO30:IFA3=<2THENTH(N,1)=TH(N,1)+.36ELSEIFA3>2ANDA3<5TH
ENTH(N,1)=TH(N,1)+1ELSEIFA3>4THENTH(N,1)=TH(N,1)+1.9
830 NEXTN
840 E4=31
850 FORN=1TO30:IFABS(X(B)-TH(N,1))>30GOTO870ELSEE6=ABS(X(B)-TH(N
,1)):E7=ABS(Y(B)-TH(N,2)):E5=SQR((E6^2)+(E7^2))
860 IFE4>E5THENE4=E5:E8=N
870 NEXTN
```

880 Set cloudbase flag if sailplane is currently above cloudbase.

```
880 IFH(B)>(A1*A5*2500)THENF1=1ELSEF1=0
890 IFF1=1THENF3=1
```

900-1020 Calculate change in height due to sailplane flying in sinking or rising air.

```
900 IFE4>30THENH(B)=H(B)-(H(B)/400)-RND(10):GOTO1060
910 IFTH(E8,3)<0THEN3310
920 IFE4>15THENH(B)=H(B)-(H(B)/(35+RND(15))):GOTO1060
930 IFE4=<6GOTO940ELSEZ3=TH(E8,3)*60/(E4-5):GOTO950
940 Z3=TH(E8,3)*60:GOTO950
950 IFF1=1THEN960ELSEZ3=Z3*((H(B)+1000)/(A1*A5*2500))+.2
960 T8=TH(E8,4)
970 IFT8<H(B)THENT8=T8/(H(B)+100)
980 IFT8=>H(B)THENT8=H(B)/(T8+100)
990 Z3=Z3*T8
1000 IFH(B)>(A1*A5*2500)+4500THENZ3=0
1010 Z3=Z3*A4
1020 H(B)=H(B)+Z3
1030 IFF1=1THEN1060
1040 TH(E8,4)=TH(E8,4)+(A1*20)
1050 IFTH(E8,4)>(A1*A5*2500)THENTH(E8,4)=A1*A5*2500
1060 H(B)=H(B)-ABS(CZ/1.5)
1070 '
```

1080-1250 Print instrument readings.

```
1080 PRINT@769,"       ";:PRINT@897,"       ";:PRINT@672,"   ";:
PRINT@789,"   ";:PRINT@915,"     ";:PRINT@925,"   .00 ";:PRINT@8
58,"           ";:PRINT@807,"       ";
1090 PRINT@671,D(B);:PRINT@787,S(B);:PRINT@914,CINT(H(B));:V1=((
H(B)-Z2)-K(B))*.02:V1=CINT(V1*100):IFV1=0THENV1=.1
1100 V1=V1/100:PRINT@926,V1;
1110 IFCZ=0THENT1=0:GOTO1180
1120 IFCZ=>40THENT1=3:GOTO1180
1130 IFCZ=<-40THENT1=-3:GOTO1180
1140 IFCZ=>10THENT1=2:GOTO1180
1150 IFCZ=<-10THENT1=-2:GOTO1180
1160 IFCZ<10ANDCZ>0THENT1=1:GOTO1180
1170 IFCZ>-10ANDCZ<0THENT1=-1
1180 PRINT@(810+T1),"^";
1190 IFV1>10THENV1=10
1200 IFV1<-10THENV1=-10
1210 V1=FIX(V1/2)
1220 PRINT@(863+V1),"^";
1230 PRINT@769,"X=";:PRINT@897,"Y=";
1240 PRINT@771,"   ";:PRINT@899,"   ";
1250 PRINT@771,CINT(X(B))/10;:PRINT@899,CINT(Y(B))/10;
```

1260 Test to see if glider has landed (for negative altitude)

```
1260 IFH(B)=<0THENL(B)=1:GOTO3030
```

1270 Test for climb up through cloudbase.

```
1270 IFF1=1ANDF2=0ANDE4=<28THEN3320
```

1280 If glider is already in cloud jump over cloud printing routine.

```
1280 IFF1=1ANDF2=1ANDE4=<28THEN1770
```

1290 Test if glider has emerged from cloud.

```
1290 IFF2=1ANDF1=0THEN3370
1300 IFF1=1ANDF2=1ANDE4>28THEN3370
1310 '
```

1320-1330 Wipe off screen.

```
1320 T1=ABS((W(B)/10)-13)
1330 FORN=0TOT1-1:PRINT@N*64,"
                              ";:NEXTN
1340 IFF3=1THEN1360
1350 IFS(B)=W(B)THEN1380
```

1360-1370 Clear off old horizon and set new one.

```
1360 T1=ABS((W(B)/10)-13):PRINT@(T1*64),"
                                        ";
1370 T1=ABS((S(B)/10)-13):PRINT@(T1*64),"----------------------
------------------------------------";
```

1380-1750 Sorts through list of thermals. Detect those that are in the sector of sky covered in the forward view from the cockpit and directs control to the cloud printing S/R which prints each sorted cloud at the appropriate location on the screen.

```
1380 IFD(B)>90ANDD(B)=<180THEN1500
1390 IFD(B)>180ANDD(B)=<270THEN1590
1400 IFD(B)>270THEN1680
1410 FORN=1TO30
1420 IFTH(N,1)<X(B)ORTH(N,1)>(X(B)+200)THEN1480
1430 IFTH(N,2)<Y(B)ORTH(N,2)>(Y(B)+200)THEN1480
1440 G1=TH(N,1)-X(B):G2=TH(N,2)-Y(B):G3=SQR((G1^2)+(G2^2)):J3=G3
:G4=G1/G3
1450 G5=ATN(G4/SQR(-G4*G4+1.001))
1460 G5=(G5/6.284)*360:G6=G5-D(B):IFABS(G6)>20THEN1480
1470 J1=INT((21+G6)*1.47):GOSUB3420
1480 NEXTN
1490 GOTO1770
1500 FORN=1TO30
1510 IFTH(N,1)<X(B)ORTH(N,1)>(X(B)+200)THEN1570
1520 IFTH(N,2)>Y(B)ORTH(N,2)<(Y(B)-200)THEN1570
1530 G2=TH(N,1)-X(B):G1=Y(B)-TH(N,2):G3=SQR((G1^2)+(G2^2)):J3=G3
:G4=G1/G3
1540 G5=ATN(G4/SQR(-G4*G4+1.001))
1550 G5=(((G5/6.284)*360)+90):G6=G5-D(B):IFABS(G6)>20THEN1570
1560 J1=INT((21+G6)*1.47):GOSUB3420
1570 NEXTN
1580 GOTO1770
1590 FORN=1TO30
1600 IFTH(N,1)>X(B)ORTH(N,1)<(X(B)-200)THEN1660
1610 IFTH(N,2)>Y(B)ORTH(N,2)<(Y(B)-200)THEN1660
1620 G1=X(B)-TH(N,1):G2=Y(B)-TH(N,2):G3=SQR((G1^2)+(G2^2)):J3=G3
:G4=G1/G3
1630 G5=ATN(G4/SQR(-G4*G4+1.001))
1640 G5=(((G5/6.284)*360)+180):G6=G5-D(B):IFABS(G6)>20THEN1660
1650 J1=INT((21+G6)*1.47):GOSUB3420
1660 NEXTN
1670 GOTO1770
1680 FORN=1TO30
1690 IFTH(N,1)>X(B)ORTH(N,1)<(X(B)-200)THEN1750
1700 IFTH(N,2)<Y(B)ORTH(N,2)>(Y(B)+200)THEN1750
1710 G2=X(B)-TH(N,1):G1=TH(N,2)-Y(B):G3=SQR((G1^2)+(G2^2)):J3=G3
:G4=G1/G3
1720 G5=ATN(G4/SQR(-G4*G4+1.001))
1730 G5=(((G5/6.284)*360)+270):G6=G5-D(B):IFABS(G6)>20THEN1750
1740 J1=INT((21+G6)*1.47):GOSUB3420
1750 NEXTN
1760 IFF1=0THENF3=0
1770 M1=M1+1
```

1780-1820 Print stopwatch in cockpit

```
1780 R(B)=R(B)+1/120
1790 T9=R(B)-CINT(R(B)):T9=CINT(T9*60)/100
1800 PRINT@820,"   .00 ";
1810 PRINT@820,CINT(R(B));
1820 IFT9=0THEN1840ELSEPRINT@822,STR$(T9);
1830
1840 W(B)=S(B):K(B)=H(B)
```

1850-2080 Control input routine using INKEY for single commands and a keyboard PEEK for the steering command which needs to have a quasi analog function.

```
1850 C$=INKEY$
1852 IFC$="K"ANDQ5=0THENQ5=1:GOTO2080
1854 IFC$="K"ANDQ5=1THENQ5=0
1856 IFQ5=1THEN2080
1860 PRINT@677," xxx ";
1870 PRINT@935,"      ";
1880 FORN=1TO42
1890 T7=PEEK(14352)
1900 IFT7=32THENCZ=0
1910 IFT7=16THENCZ=CZ-3
1920 IFT7=64THENCZ=CZ+3
1930 IFCZ<-90THENCZ=-90
1940 IFCZ>90THENCZ=90
1950 IFCZ<0THENPRINT@935,ABS(CZ);
1960 IFCZ>0THENPRINT@938,CZ;
1970 IFCZ=0THENPRINT@935,"0    0";
1980 A$=INKEY$
1990 IFA$=""THEN2060
2000 IFA$="8"THENS(B)=S(B)+10
2010 IFA$="2"THENS(B)=S(B)-10
2020 IFA$="S"THENQ1=1
2030 IFA$="T"THENQ2=1
2040 IFA$="F"THENQ3=1
2050 IFA$="R"THENQ4=1
2055 IFA$="K"THENQ5=1
2060 NEXTN
2070 '
2080 PRINT@677,"     ";
2090 IFS(B)<40THENS(B)=40
2100 IFS(B)>120THENS(B)=120
```

2110-2140 Direct control to appropriate S/R for R,S,T,F functions.

```
2110 IFQ1=1THENGOSUB2800
2120 IFQ2=1THENGOSUB2860
2130 IFQ3=1THENGOSUB2950
2140 IFQ4=1THENGOSUB3930
2150 Q1=0:Q2=0:Q3=0
```

2160-2190 Calculate new compass heading.

```
2160 D(B)=D(B)+CZ
2170 IFD(B)>360THEND(B)=D(B)-360
2180 IFD(B)<0THEND(B)=360+D(B)
2190 D(B)=CINT(D(B))
```

2200 If less than one hour of flight has elapsed loop back to start of active program for next program loop.

```
2200 IFM1=<120THEN690
2210 IFB=P1THENZ390
```

2220-2340 If players hour of flight is over print map via S/R and print position report on sailplane.

```
2220 GOSUB3060
2230 IFL(B)=1THENPRINT@769,"LANDED";
2240 PRINT@513,B$(B);":-";:PRINT@577,"POSITION";:PRINT@641,"REPO
RT";
2250 T6=INT(X(B)/10)+30:IFT6<30THENT6=30
2260 IFT6>125THENT6=125
2270 T7=INT(Y(B)/20):T7=45-T7:IFT7<1THENT7=1
2280 IFT7>40THENT7=40
2290 FORO=1TO7
2300 FORN=1TO50
2310 SET(T6,T7):NEXTN
2320 FORN=1TO50
2330 RESET(T6,T7):NEXTN
2340 NEXTO
2350 M1=0
2360 IFL(B)=1THENPRINT@769,"      ";
2370 B=B+1:IFL(B)=0THEN100
2380 B=B+1:IFL(B)=0THEN100
```

2390 At the end of an hour's flight for all the competitors a map and position report is printed for all competitors in order.

```
2390 B=1:M1=0
2400 GOSUB3060
2410 PRINT@513,B$(B);":-";:PRINT@577,"POSITION";:PRINT@641,"REPO
```

```
RT";
2420 IFL(B)=1THENPRINT@769,"LANDED";
2430 T6=INT(X(B)/10)+30:IFT6<30THENT6=30
2440 IFT6>125THENT6=125
2450 T7=INT(Y(B)/20):T7=45-T7:IFT7<1THENT7=1
2460 IFT7>40THENT7=40
2470 FORD=1TO7
2480 FORN=1TO50
2490 SET(T6,T7):NEXTN
2500 FORN=1TO50
2510 RESET(T6,T7):NEXTN
2520 NEXTD
2530 IFL(B)=1THENPRINT@769,"      ";
2540 B=B+1
2550 IFB<P1+1THEN2410
2560 IFL(1)=1ANDL(2)=1ANDL(3)=1THEN2660
```

2570-2640 Adjust cloudbase heights and thermal strengths depending on the time of day.

```
2570 IFA5<1THENA5=A5+.1
2580 IFF8=0THENA4=A4+.1
2590 IFF8=1THENA4=A4-.2
2600 IFA4=>1THENF8=1
2610 IFA4<=.6THENA4=.1
2620 IFL(1)=0THENB=1
2630 IFL(1)=1THENB=2
2640 IFL(1)=1ANDL(2)=1THENB=3
```

2650 Test to see if randomiser wants to insert an unpredictable change in weather conditions - if so update weather definition variables.

```
2650 IFRND(8)<=7THEN100ELSEA1=RND(3):GOTO80
2660 B=1
```

2670 Calculate finish times.

```
2670 T1=VV(B)-CINT(VV(B)):T1=CINT(T1*60)/100:VV(B)=CINT(VV(B))+T
1:B=B+1
2680 IFB<4THENZ670ELSEB=1
```

2690 Print results

```
2690 CLS:PRINT@76,CHR$(23);"R E S U L T S":PRINT
2700 IFVV(B)=0THENZ780ELSEPRINTB$(B);" TOOK";CINT(VV(B));"/";(VV
(B)-CINT(VV(B)))*100;"(HRS/MIN)":B=B+1
2710 IFB<4THENZ700
```

2730-2770 Is another game required?

```
2730 PRINT:PRINT"ANOTHER GAME? (Y OR N)"
2740 I$=INKEY$
2750 IFI$="Y"THENRUN
2760 IFI$=""THEN2740
2770 END
2780 IFB>P1THENB=B+1:GOTO2710
2790 PRINTB$(B);" DID NOT FINISH TASK":B=B+1:GOTO2710
```

2800-2850 Start line S/R.

```
2800 IFX(B)>=200THEN2840
2810 IFY(B)>=200THEN2840
2820 IFH(B)>3281THEN2850
2830 R(B)=0:PRINT@20,"'**** - G O O D   S T A R T - ****';:RET
URN
2840 PRINT@20,"'NEGATIVE START - OUT OF POSITION'";:RETURN
2850 PRINT@20,"'NEGATIVE START  -  TOO HIGH'";:RETURN
```

2860-2940 Turnpoint S/R.

```
2860 IFW(B)=1THEN2910:'TURN S/R
2870 IFY(B)<U2*10THEN2900
2880 IFX(B)<((U1*10)-20)ORX(B)>((U1*10)+20)THEN2900
2890 WZ(B)=1:PRINT@20,"'GOOD TURN AT FIRST TURNPOINT'";:RETURN
2900 PRINT@6,"'NEGATIVE CONTROL AT FIRST TURN - OUT OF POSITION'
";:RETURN
2910 IFY(B)>U4*10THEN2940
2920 IFX(B)<((U3*10)-20)ORX(B)>((U3*10)+20)THEN2940
2930 WZ(B)=2:PRINT@20,"'GOOD TURN AT SECOND TURNPOINT'";:RETURN
2940 PRINT@6,"'NEGATIVE CONTROL AT SECOND TURN - OUT OF POSITION
'";:RETURN
```

2950-3020 Finish line S/R.

```
2950 IFX(B)>=200THEN3000
2960 IFY(B)>=200THEN3000
2970 IFH(B)>3281THEN3010
2980 IFW(B)<2THEN3020
2990 VV(B)=R(B):L(B)=1:Q3=0:CLS:PRINTCHR$(23):PRINT@448,"'";B$(B
);", GOOD FINISH
        WELL DONE'";:FORN=1TO2600:NEXT:GOTO2210
3000 PRINT@20,"'NEGATIVE FINISH - OUT OF POSITION'";:RETURN
3010 PRINT@20,"'NEGATIVE FINISH  -  TOO HIGH'";:RETURN
3020 PRINT@0,"'NEGATIVE FINISH - TURNPOINTS HAVE NOT BEEN ROUNDE
D CORRECTLY'";:RETURN
3030 CLS:PRINTCHR$(23):PRINT@448,B$(B);" YOU HAVE LANDED SAFELY.
";
3040 FORN=1TO2600:NEXT
3050 GOTO2210
```

3060-3300 Map printing S/R.

```
3060 CLS:FORN=28TO127:SET(N,0):SET(N,41):NEXT
3070 FORN=0TO41:SET(28,N):SET(29,N):SET(126,N):SET(127,N):NEXT
3080 FORN=5TO40 STEP 5:SET(26,N):SET(27,N):NEXT
3090 FORN=40TO120 STEP 10:SET(N,42):SET(N+1,42):NEXT
3100 POKE15562,55:POKE15563,48:POKE15882,52:POKE15883,48:POKE162
02,49:POKE16203,48
3110 FORN=16340TO16380 STEP 5:POKEN,48:NEXT
3120 N=16339:POKEN,49:POKEN+5,50:POKEN+10,51:POKEN+15,52:POKEN+2
0,53:POKEN+25,54:POKEN+30,55:POKEN+35,56:POKEN+40,57
3130 PRINT@969,"X-AXIS";:POKE16335,94:PRINT@130,"Y-AXIS";:POKE15
497,92:PRINT@376,"360";:PRINT@500,"270     90";:PRINT@696,"180";
:POKE15800,160:POKE15801,190:POKE15802,180:FORN=21TO28:SET(115,N
):NEXT:FORN=111TO119:SET(N,22):NEXT
3140 PRINT@2,"POSITION";:PRINT@66,"DISPLAY";:FORN=30TO95 STEP 5:
SET(N,10):NEXT:FORN=10TO40 STEP 3:SET(95,N):NEXT:PRINT@84,"DOTTE
D LINE INDICATES SUGGESTED TASK AREA";:PRINT@160,"WIND DIRECTION
 -> ->";
3150 FORN=16079TO16088:POKEN,160:NEXT
3160 FORN=16143TO16152:POKEN,153:NEXT
3170 FORN=16207TO16216:POKEN,182:NEXT
3180 T6=INT(U1)+27:IFT6<30THENT6=30
3190 IFT6>125THENT6=125
3200 T7=INT(U2/2):T7=45-T7:IFT7<1THENT7=1
3210 IFT7>40THENT7=40
3220 FORN=1TO6:SET(T6+N,T7):NEXT
3230 T7=T7-2:T6=T6+3:FORN=1TO3:SET(T6,T7+N):SET(T6+1,T7+N):NEXT
3240 T6=INT(U3)+27:IFT6<30THENT6=30
3250 IFT6>125THENT6=125
3260 T7=INT(U4/2):T7=45-T7:IFT7<1THENT7=1
3270 IFT7>40THENT7=40
3280 FORN=1TO6:SET(T6+N,T7):NEXT
3290 T7=T7-2:T6=T6+3:FORN=1TO3:SET(T6,T7+N):SET(T6+1,T7+N):NEXT
3300 CZ=0:RETURN
3310 Z3=TH(E8,3)*25:Z3=Z3/E4:Z3=Z3*H(B)/(A1*2500):H(B)=H(B)+Z3:G
OTO1060
```

3320 Screen white out.

```
3320 Z=0
3330 PRINT@Z,STRING$(64,CHR$(191));
3340 IFZ>512THEN3360
3350 Z=Z+64:GOTO3330
3360 F2=1:GOTO1770
3370 Z=576
```

3380 Clear screen white out.

```
3380 PRINT@Z,STRING$(64,CHR$(32));
3390 IFZ<64THEN3410
3400 Z=Z-64:GOTO3380
3410 F2=0:GOTO1370
```

3420-3680 Cloud printing S/R.

```
3420 IFJ3>200THENJ3=200
3430 J2=ABS((S(B)/10)-13)
3440 IFJ3>100THEN3610
3450 IFJ3>40ANDJ3=<100THEN3540
3460 IFJ3<10THEN3680
3470 J4=-12:J5=12:J4=J1+J4:J5=J1+J5:IFJ4<0THENJ4=0
3480 IFJ5>63THENJ5=63
3490 IFF1=1THEN3510
3500 J2=J2-5
3510 IFJ2<0THEN3680ELSEJ2=(J2*64)
3520 PRINT@J2+J4,CHR$(184);:PRINT@J2+J4+1,STRING$(J5-J4-1,CHR$(1
91));
3530 PRINT@J2+J5,CHR$(180);:GOTO3680
3540 J4=-8:J5=8:J4=J1+J4:J5=J1+J5:IFJ4<0THENJ4=0
3550 IFJ5>63THENJ5=63
3560 IFF1=1THEN3580
3570 J2=J2-3
3580 IFJ2<0THEN3680ELSEJ2=(J2*64)
3590 PRINT@J2+J4,CHR$(184);:PRINT@J2+J4+1,STRING$(J5-J4-1,CHR$(1
91));
3600 PRINT@J2+J5,CHR$(180);:GOTO3680
3610 J4=-4:J5=4:J4=J1+J4:J5=J1+J5:IFJ4<0THENJ4=0
3620 IFJ5>63THENJ5=63
3630 IFF1=1THEN3650
3640 J2=J2-2
3650 IFJ2<0THEN3680ELSEJ2=(J2*64)
3660 PRINT@J2+J4,CHR$(184);:PRINT@J2+J4+1,STRING$(J5-J4-1,CHR$(1
91));
3670 PRINT@J2+J5,CHR$(180);
3680 RETURN
```

3690-3920 Cockpit printing S/R.

```
3690 CLS:Z=16000
3700 FORN=0TO12:POKEZ+N,188:NEXT
3710 FORN=13TO26:POKEZ+N,140:NEXT
3720 POKEZ+27,191:POKEZ+36,191:POKEZ+28,67:POKEZ+29,79:POKEZ+30,
77:POKEZ+31,80:POKEZ+32,58:FORN=37TO47:POKEZ+N,140:NEXT
3730 FORN=48TO63:POKEZ+N,188:NEXT:Z=Z+64
3740 FORN=0TO12:POKEZ+N,191:NEXT
3750 FORN=14TO38:POKEZ+N,191:NEXT
3760 POKEZ+39,131:POKEZ+40,147:POKEZ+41,147:POKEZ+42,179:POKEZ+4
3,163:POKEZ+44,163:POKEZ+45,131:POKEZ+46,191:POKEZ+48,143:POKEZ+
49,143
3770 FORN=50TO63:POKEZ+N,191:NEXT
3780 Z=Z+64:FORN=0TO12:POKEZ+N,191:NEXT
3790 POKEZ+14,191:POKEZ+24,191:POKEZ+16,65:POKEZ+17,83:POKEZ+18,
73:POKEZ+19,58:POKEZ+25,45:FORN=26TO30:POKEZ+N,160:NEXT
3800 POKEZ+31,188:FORN=32TO36:POKEZ+N,144:NEXT
3810 POKEZ+37,43:POKEZ+38,191:POKEZ+46,191:POKEZ+47,191:POKEZ+48
,191:FORN=50TO63:POKEZ+N,191:NEXT
3820 Z=Z+64:FORN=0TO12:POKEZ+N,191:NEXT
3830 FORN=14TO24:POKEZ+N,191:NEXT
3840 POKEZ+40,93:POKEZ+44,94:POKEZ+25,86:POKEZ+37,86:POKEZ+38,19
1:POKEZ+41,84:POKEZ+42,38:POKEZ+43,83:POKEZ+46,191:POKEZ+47,191:
POKEZ+48,191:FORN=50TO63:POKEZ+N,191:NEXT
3850 Z=Z+64:FORN=0TO12:POKEZ+N,191:NEXT
3860 POKEZ+14,191:POKEZ+15,65:POKEZ+16,76:POKEZ+17,84:FORN=24TO2
8:POKEZ+N,191:NEXT
3870 POKEZ+38,191:POKEZ+46,191:POKEZ+48,188:POKEZ+49,188:FORN=50
TO63:POKEZ+N,191:NEXT
3880 Z=Z+64:FORN=0TO12:POKEZ+N,191:NEXT
3890 POKEZ+13,176:POKEZ+14,179:POKEZ+15,179:POKEZ+16,131:FORN=17
TO46:POKEZ+N,191:NEXT
3900 FORN=48TO63:POKEZ+N,191:NEXT
3910 PRINT@((ABS((W(B)/10)-13))*64),"------------------------------
--------------------------------";
3920 RETURN
```

3930-3960 Relaunch S/R.

```
3930 IFXY(B)=1THENPRINT@10,"'YOU HAVE ALREADY TAKEN YOUR ONE PER
MITTED RELAUNCH'":Q4=0:RETURN
3940 IFR(B)=>1THENPRINT@10,"'RELAUNCH NOT NOW PERMITTED'":Q4=0:R
ETURN
3950 PRINT@20,"'RELAUNCH EXECUTING'";:PRINT@((ABS((W(B)/10)-13))
*64),"
     ";:W(B)=3000:S(B)=60:W(B)=60:D(B)=45:K(B)=3000:X(B)=200:Y(
B)=200:R(B)=0:XY(B)=1:Q4=0:CZ=0
3960 PRINT@((ABS((W(B)/10)-13))*64),"------------------------------
--------------------------------";:RETURN
```

LIST OF VARIABLES FOR SAILPLANE

ARRAYS:

D(1-3) =	Compass heading
H(1-3) =	Height
K(1-3) =	Previous height
L(1-3) =	Landings flags
R(1-3) =	Time taken
S(1-3) =	Current speed
TH(1-30,1-4) =	Thermal array . . . . .

Line 1 = X-coordinate
Line 2 = Y-coordinate
Line 3 = Strength (pos. or neg.)
Line 4 = Height of vertical centroid of thermal above ground level.

VV(1-3) =	Finish times
W(1-3) =	Previous speed
X(1-3) =	X-coordinate of sailplanes
XY(1-3 =	Relaunch flags
Y(1-3) =	Y-coordinate of sailplanes

String Array

B$(1-3)=	Name of competitors

SINGLE VARIABLES

A1	=	Thermal strength multiplier
A2	=	Weather type
A3	=	Wind factor
A4	=	Multiplier for strength of thermal at particular time of day.
A5	=	Cloud base factor
B	=	Player counter
CZ	=	Control store for INKEY commands
E1	=	X-coordinate of closest thermal
E2	=	Y-coordinate of closest thermal
E3	=	Strength
E4	=	Diagonal distance to closest thermal
E5	=	Temporary trigonometric store
E6	=	Temporary trigonometric store
E7	=	Temporary trigonometric store
E8	=	Temporary store to hold identity of closest thermal

F1 = Cloudbase flag
F2 = Is screen white out operative (1=yes, 0=no)
F3 = Flag to indicate requirement to print horizon if descending out of cloud.
F8 = Flag to indicate middle of convective period
F9 = Flag to inhibit printing of new Met. Forecast if randomiser changes weather during middle of game.
G1 = Temporary X-coordinate
G2 = Temporary Y-coordinate
G3 = Diagonal distance to thermal
G4 = Sine of angle from datum
G5 = Angle of thermal from datum
G6 = Angle of thermal from course line of sailplane
(ALL USED IN PRINTING CLOUDS ON SCREEN)
J1 = Position left and right on screen (Values 0-60)
J2 = Position of horizon on screen (0-9)
J3 = Distance from cloud
J4 = Left limit of screen
J5 = Right limit of screen
(USED IN CLOUD PRINT ROUTINE)
M1 = Time taken in current go (up to one hour)
P1 = Number of players (1 to 3)
Q1 = Start flag
Q2 = Turn flag
Q3 = Finish flag
Q4 = Relaunch flag
T1-T5= Temporary stores
T6 = Launch time
U1 = X-coordinate of first turnpoint
U2 = Y-coordiante of first turnpoint
U3 = X-coordinate of second turnpoint
U4 = Y-coordinate of second turnpoint
V1 = Variometer
Z = Cloud print location store
Z2 = Total energy height change value
Z3 = Value for thermal lift before allowance is made for height above ground and height above or below vertical centroid of thermal.

SAVE YOUR (CRASHED) SCRIPSIT

32K or 48K TRS-80 Disk Systems.

Several times I have lost work because my TRS-80 crashed back to DOS from Scripsit. Scripsit does not allow reentry without zeroing the current file. The following procedure will allow you to recover a file that is still in memory, but lost due to such a problem.

You will need a copy of RSM/2D from Small Systems Software (still this editor's favorite program!) and the ability to convert Hex numbers to decimal.

Once the program has crashed to DOS READY, remove the diskette from drive 0 and insert the RSM/2D diskette. This has a very short boot routine that will not interfere with the resident program.

Press RESET to boot the RSM/2D loader. Select the version that will load in the top of memory in your computer, and load the program.

Use the A command to search for your text file. I use the command A 7000. When you find the program, write down the starting and ending points. Convert these numbers to POKE addresses. A POKE address is a decimal number up to 32767. After that, it is the decimal equivalent of the Hex number minus 65536. Thus, 7FFF Hex equals 32767, and 8003 Hex equals 32771-65536 or -32765.

If you have trouble converting a number from Hex to Decimal, use this process. Assume that you have a four-digit Hex number, such as 72CF.

Multiply the left digit by 4096: 7 x 4096 equals 28672.

Multiply the next digit by 256: 2 x 256 equals 512.

Multiply the next digit by 16: 12 x 16 equals 192.

Write down the decimal equivalent of the remaining digit: 15.

Add these four numbers: 29391. This is your decimal number.

Remember that in the Hexadecimal system, A equals 10, B equals 11, **C equals 12, D equals 13, E equals 14, and F equals 15.**

Now, if your number is greater than 32767, subtract 65536 to get the POKE address. For example, if you had 82FE, the decimal equivalent is 33534. Subtracting 33534 minus 65536, we get a POKE address of -32002. We now know that our text file is located in memory starting at 29391 and ending at -32002. All we need to **do now is go to disk BASIC without disturbing the text area, PEEK our text our of memory and write it to a disk file, then reload Scripsit, and load our file back in.**

If our program starts at 29391, we can proceed by inserting a SYSTEM diskette (it is best to keep fresh disks on hand with DOS and BASIC already on them, but any disc that has enough free space for the file and no AUTO message will do) into drive 0 and rebooting DOS.

Now we enter BASIC, protecting our text area by reserving memory right below our text, with the command: BASIC 29390 (NewDOS), or BASIC and answering Memory Size with 29390 (TRSDOS).

Now we need to write a program to save the program on disk. This program can be used by simply changing the starting and ending points from 29391 and -32002 to the values you find.

```
10 OPEN, "0", 1, "RECOVER/TXT:0"
20 FOR M = 29391 TO 32767
30 PRINT # 1, CHR$(PEEK(M));
40 NEXT M
50 FOR M = -32768 TO -32002
60 PRINT # 1, CHR$(PEEK(M));
70 NEXT M
80 PRINT # 1,CHR$(0)
90 CLOSE
```

Now type RUN and your program will store the contents of memory in a file called RECOVER/TXT. Line 80 terminates the file with a zero to tell Scripsit where the text ends. All that is left is to load Scripsit and then the file RECOVER/TXT.

Of course, this same method can be used to recover almost anything in memory and store it as a disk file. The Scripsit example is just to start you thinking. If memory is tight, go back to BASIC specifying only one file buffer to save nearly 600 bytes.

by George Blank

You can have SOUND on YOUR computer!!!

Sound for the Apple, TRS–80, and Atari!

Anyone who owns a TRS-80, Atari 800, or Apple computer has the capability for sound effects. Three programs for using these sounds (one for each computer) are presented here. Although all three programs do the same thing, they are very different. The main difference is that the TRS-80 does not have built-in sound capabilities, the Apple does not have built-in sound routines, and the Atari has both. Therefore, sound on the TRS-80 and Apple must be added by a machine language program. The TRS-80 program sends signals out the cassette port, which can be picked up through an audio amplifier, such as Soundware, available with three demonstration programs from TSE for $24.95. One simply removes the plug from the AUX jack in the tape recorder and plugs it into the amplifier for sound. The Apple program uses the Apple's built-in speaker for sound. On the Atari, you can control the pitch, distortion and loudness of each of four separate voices independently and from BASIC, giving it a slight advantage.

The TRS-80 program uses a string variable, S$, to hold the notes to be played. Each note consists of two characters, the note duration (1-255) and the note pitch (also 1-255 with 1 being the highest on the scale). The program then jumps to a machine language program which plays the notes. On the Apple program, the ampersand (&) is defined as a note playing routine. (See April's SoftSide: Apple for details on the ampersand.) A statement like '& T50,100' would play one note with a pitch of 50 and a duration of 100. The limits for the duration and pitch are the same as with the TRS-80 program, 1-255. On the Atari, the notes are played using the SOUND command. An example of a BASIC sound command is 'SOUND 1,100,10,7'. That would produce a sound from voice 1 with a pitch of 100, a distortion of 10, and a loudness of 7. The voice must be in the range of 0-3, the pitch 1-255, the distortion an even number from 0-14 and the loudness 1-15. Any note will be played until an END statement is encountered or until another note is played using the same voice. To shut off a voice, play a note to that voice with a loudness of zero. In this program, the variables V, T and L may be set to different values for the voice number, tone and loudness to be used.

Sound can add to any program, making it more interesting to the user, and, with a little imagination, the possibilities of sound are almost endless.

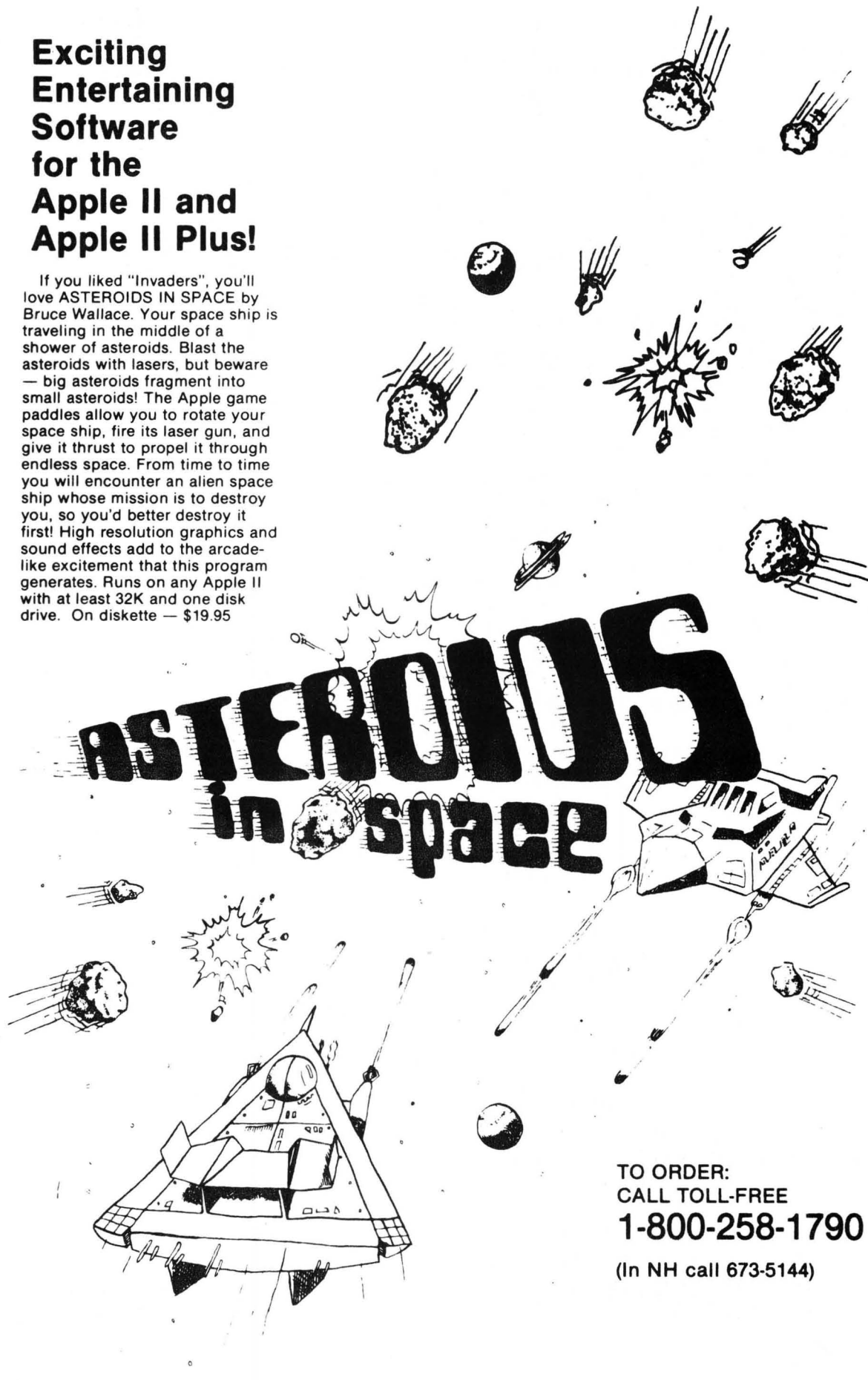
Listing for the TRS-80 :

```
100 CLEAR1000
110 FORX=32512TO32578:READA:POKEX,A:NEXTX
120 DEFUSR=&H7F00'    FOR LEVEL II - POKE 16526,0:POKE16527,127
130 DATA 205,127,10,126,50,66,127,35,94,35,86,235,43,58,64,56,20
3,87,192,58
140 DATA 66,127,183,200,61,200,61,50,66,127,35,86,30,0,35,70,62,
1,211,255
150 DATA 16,254,70,60,211,255,16,254,70,122,183,32,7,123,183,40,
212,61,40
160 DATA 209,27,27,16,241,24,225,0
170 CLS:PRINT"PHASER SOUNDS":FORP=1TO100STEP2:S$=S$+CHR$(2)+CHR$
(P):NEXTP
180 FORI=1TO5:U=USR(VARPTR(S$)):NEXTI
190 PRINT:PRINT"OLD MACDONALD"
200 FORT=1TO12:READA,B:S$=CHR$(B)+CHR$(A):U=USR(VARPTR(S$)):NEXT
T
210 DATA 106,88,106,88,106,88,142,88,126,88,126,88,142,170,84,88
220 DATA 84,88,94,88,94,88,106,255
230 PRINT:PRINT"CLEMENTINE"
240 FORT=1TO30:READA,B:S$=CHR$(B)+CHR$(A):U=USR(VARPTR(S$)):NEXT
T
250 DATA 138,44,138,44,138,108,180,108,110,44,110,44,110,108,138
,108,110,44
260 DATA 110,44,86,108,86,108,102,44,110,44,128,255,128,44,110,4
4,102,108
270 DATA 102,108,110,44,128,44,110,108,138,108,138,44,110,44,128
,108,180
280 DATA 108,150,44,128,44,138,255
290 PRINT:PRINT"LOW NOTES - PAUSES BETWEEN PLAYING TONES"
300 FORJ=1TO9:S$=CHR$(1)+CHR$(255):FORI=1TO20:K=USR(VARPTR(S$)):
FORP=1TOJ
310 NEXTP,I,J:PRINT:PRINT"MISCELLANEOUS"
320 FORI=2TO127STEP2:S$=S$+CHR$(1)+CHR$(128-I)+CHR$(1)+CHR$(128+
I):NEXTI
330 FORT=1TO7:U=USR(VARPTR(S$)):NEXTT
340 S$="":PRINT
350 FORP=100TO1STEP-3:S$=S$+CHR$(2)+CHR$(P):NEXTP
360 FORP=1TO88STEP4:S$=S$+CHR$(1)+CHR$(P):NEXTP
370 FORI=1TO5:U=USR(VARPTR(S$)):PRINTCHR$(29);"RED ALERT";
380 U=USR(VARPTR(S$)):PRINTCHR$(29);STRING$(9,143);:NEXTI
```


continued on page 68

Exciting Entertaining Software for the Apple II and Apple II Plus!

If you liked "Invaders", you'll love ASTEROIDS IN SPACE by Bruce Wallace. Your space ship is traveling in the middle of a shower of asteroids. Blast the asteroids with lasers, but beware — big asteroids fragment into small asteroids! The Apple game paddles allow you to rotate your space ship, fire its laser gun, and give it thrust to propel it through endless space. From time to time you will encounter an alien space ship whose mission is to destroy you, so you'd better destroy it first! High resolution graphics and sound effects add to the arcade-like excitement that this program generates. Runs on any Apple II with at least 32K and one disk drive. On diskette — $19.95



TO ORDER:
CALL TOLL-FREE
1-800-258-1790
(In NH call 673-5144)

Listing for the Apple in Applesoft :

```
100  REM  AMPERSAND TONES
110  REM        BY
120  REM    MARK CROSS
130  REM
140  REM
150  GOSUB 480
160  TEXT : HOME
170  PRINT "PHASER SOUNDS"
180  FOR I = 1 TO 5: FOR P = 1 TO
     100 STEP 2: & TP,2: NEXT P: NEXT
     I
190  FOR PAUSE = 1 TO 1000: NEXT
     PAUSE
200  PRINT : PRINT "OLD MACDONALD
     "
210  & T106,128: & T106,128: & T1
     06,128: & T142,128
220  & T126,128: & T126,128: & T1
     42,255
230  & T84,128: & T84,128: & T94,
     128: & T94,128
240  POKE 805,3: & T106,128: REM
      805 PLAYS TRIPLE LENGTH NOT
     E
250  POKE 805,1: REM  RETURN TO N
     ORMAL LENGTH
260  FOR PAUSE = 1 TO 1000: NEXT
     PAUSE
270  PRINT : PRINT "CLEMENTINE"
280  & T188,64: & T188,64
290  & T188,128: & T255,128: & T1
     50,64: & T150,64
300  & T150,128: & T188,128: & T1
     50,64: & T150,64
310  & T126,128: & T126,128: & T1
     42,64: & T150,64
320  & T168,255: & T168,64: & T15
     0,64
330  & T142,128: & T142,128: & T1
     50,64: & T168,64
340  & T150,128: & T188,128: & T1
     88,64: & T150,64
350  & T168,128: & T255,128: & T2
     00,64: & T168,64
360  & T188,255
370  FOR PAUSE = 1 TO 1000: NEXT
     PAUSE
380  PRINT : PRINT "LOW NOTES - P
     AUSES BETWEEN PLAYING TONES"
390  FOR J = 1 TO 9: FOR I = 1 TO
     100: & T255,1: FOR PAUSE = 1
      TO J: NEXT PAUSE,I,J
400  FOR PAUSE = 1 TO 1000: NEXT
     PAUSE
410  PRINT : PRINT "MISCELLANEOUS
     "
420  FOR I = 1 TO 127: POKE 805,4
     : & T128 - I,1: POKE 805,1: &
     T128 + I,1: NEXT I
430  FOR PAUSE = 1 TO 1000: NEXT
     PAUSE
440  PRINT : PRINT "RED ALERT"
450  FOR I = 1 TO 5: FOR P = 100 TO
     1 STEP  - 2: & TP,3: NEXT P
460  FOR P = 1 TO 88 STEP 2: & TP
     ,2: NEXT P,I
470  END
480  RESTORE : REM  THIS SUBROUTI
     NE POKES THE TONE
490  REM  PLAYING PROGRAM INTO ME
     MORY
500  DATA 201,84,208,15,32,177,0,
     32,248,230,138,72,32,183,0,2
     01,44,240,3,76,201,222,32,17
     7,0,32,248,230
510  FOR I = 768 TO 833: READ P: POKE
     I,P: NEXT I
520  DATA 104,134,3,134,1,133,0,1
     70,160,1,132,2,173,48,192,13
     6,208,4,198
530  DATA   1,240,7,202,208,246,1
     66,0,208,239,165,3,133,1,198
     ,2,208,241,96
540  REM  CALL TONE SUB WITH AMPE
     RSAND T PITCH COMMA DURATION
550  REM  PITCH AND DURATION MUST
      BE 255 OR LESS
560  REM  POKE 805, NUMBER OF TIM
     ES TO REPEAT THE NOTE
570  REM  DEFAULT LOCATION 805=1
580  POKE 1013,76: POKE 1014,0: POKE
     1015,3
590  RETURN
```

Listing for the Atari:

```
50 L=10:T=100:V=0
100 GRAPHICS 0:PRINT "PHASER SOUNDS"
110 FOR X=1 TO 5:FOR P=1 TO 100 STEP 3
120 SOUND V,P,T,L:NEXT P:NEXT X
130 PRINT :PRINT "OLD MACDONALD"
140 FOR X=1 TO 12:READ A,B:SOUND V,A,T,L

150 FOR I=1 TO B:NEXT I:SOUND V,0,0,0:NE
XT X
160 DATA 106,108,106,108,106,108,142,108
,126,108,126,108,142,200,84,108,84,108,9
4,108,94,108,106,255
165 PRINT :PRINT "CLEMENTINE"
170 FOR X=1 TO 30:READ A,B:SOUND V,A,T,L

180 FOR I=1 TO B:NEXT I:SOUND V,0,0,0:NE
XT X
190 DATA 188,64,188,64,188,128,255,128,1
50,64,150,64,150,128,188,128,150,64,150,
64,126,128,126,128,142,64,150,64
200 DATA 168,255,168,64,150,64,142,128,1
42,128,150,64,168,64,150,128,188,128,188
,64,150,64,168,128,255,128
219 DATA 200,64,168,64,188,255
220 PRINT :PRINT "LOW NOTES - PAUSES BET
WEEN NOTES"
230 FOR J=1 TO 9:FOR I=1 TO 20:SOUND V,2
55,T,L:SOUND V,0,0,0
240 FOR P=1 TO J:NEXT P:NEXT I:NEXT J
250 PRINT :PRINT "MISCELLANEOUS"
260 FOR X=1 TO 3:FOR I=0 TO 127:SOUND 0,
I*2,10,10
262 SOUND 1,255-I*2,10,10:SOUND 2,I,10,1
0:SOUND 3,255-I,10,10:NEXT I:NEXT X
265 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,
0,0,0:SOUND 3,0,0,0
270 PRINT :PRINT "RED ALERT"
280 FOR X=1 TO 5:FOR P=100 TO 1 STEP -3
290 SOUND V,P,T,L:FOR I=1 TO 2:NEXT I:NE
XT P
300 FOR P=1 TO 88 STEP 4:SOUND V,P,T,L:N
EXT P:NEXT X
```

For those interested, an assembly listing of the TRS-80 machine language tone maker follows:

continued on page 69

```
7F00            00100           ORG     7F00H
7F00 CD7F0A     00110 START     CALL    0A7FH       ;GET USR ARGUMENT
7F03 7E         00120           LD      A,(HL)      ;GET LENGTH OF STRING
7F04 32427F     00130           LD      (COUNT),A   ;PUT IN COUNT
7F07 23         00140           INC     HL          ;ADVANCE VARPTR POINTER
7F08 5E         00150           LD      E,(HL)      ;PUT STRING ADDRESS
7F09 23         00160           INC     HL          ;POINTER IN
7F0A 56         00170           LD      D,(HL)      ;THE DE REGISTER PAIR,
7F0B EB         00180           EX      DE,HL       ;THEN TRANSFER TO HL.
7F0C 2B         00185           DEC     HL          ;START ONE BEFORE
7F0D 3A4038     00190 PLAY      LD      A,(14400)   ;SCAN KBD
7F10 CB57       00200           BIT     2,A         ;CHECK FOR BREAK KEY
7F12 C0         00210           RET     NZ          ;IF BREAK IS HIT, RETURN
7F13 3A427F     00220           LD      A,(COUNT)   ;GET CHARACTER COUNT
7F16 B7         00230           OR      A           ;IS IT ZERO?
7F17 C8         00240           RET     Z           ;YES, RETURN
7F18 3D         00250           DEC     A           ;IS IT ONE?
7F19 C8         00260           RET     Z           ;INCOMPLETE PAIR, RETURN
7F1A 3D         00270           DEC     A           ;DECREMENT AGAIN
7F1B 32427F     00280           LD      (COUNT),A   ;PUT BACK IN CHAR. COUNT
7F1E 23         00285           INC     HL          ;ADVANCE TO DURATION
7F1F 56         00290           LD      D,(HL)      ;PUT DURATION * 256
7F20 1E00       00300           LD      E,0         ;INTO DE PAIR.
7F22 23         00310           INC     HL          ;ADVANCE TO PITCH
7F23 46         00330 SOUND     LD      B,(HL)      ;GET PITCH
7F24 3E01       00340           LD      A,1
7F26 D3FF       00360           OUT     (255),A     ;CYCLE ONE
7F28 10FE       00370 LOOP1     DJNZ    LOOP1       ;DELAY FOR PITCH
7F2A 46         00380           LD      B,(HL)      ;GET PITCH BACK
7F2B 3C         00400           INC     A
7F2C D3FF       00410           OUT     (255),A     ;CYCLE TWO
7F2E 10FE       00420 LOOP2     DJNZ    LOOP2       ;DELAY FOR PITCH
7F30 46         00430           LD      B,(HL)      ;GET PITCH AGAIN
7F31 7A         00440 LOOP3     LD      A,D         ;CHECK MSB OF DE
7F32 B7         00444           OR      A           ;FOR ZERO.
7F33 2007       00448           JR      NZ,JUMP     ;NO, CONT
7F35 7B         00452           LD      A,E         ;YES, CHECK LSB
7F36 B7         00456           OR      A           ;FOR ZERO
7F37 28D4       00460           JR      Z,PLAY      ;IF YES, NEW NOTE
7F39 3D         00464           DEC     A           ;WAS DE A ONE
7F3A 28D1       00468           JR      Z,PLAY      ;IF YES, NEW NOTE
7F3C 1B         00472 JUMP      DEC     DE          ;DECREMENT DURATION
7F3D 1B         00476           DEC     DE          ;TWICE
7F3E 10F1       00480           DJNZ    LOOP3       ;LOOP FOR DURATION
7F40 18E1       00488           JR      SOUND       ;CONTINUE NOTE UNTIL
                00489                               ;DURATION IS OVER.
0001            00510 COUNT     DEFS    1           ;STORAGE AREA
0001            00520           END                 ;THAT'S ALL FOLKS!
00000 TOTAL ERRORS

COUNT  7F42 00510   00130 00220 00280
JUMP   7F3C 00472   00448
LOOP1  7F28 00370   00370
LOOP2  7F2E 00420   00420
LOOP3  7F31 00440   00480
PLAY   7F0D 00190   00460 00468
SOUND  7F23 00330   00488
START  7F00 00110
```

continued from page 24

whether there is a next line. If there is, jump back to line 1 and keep looking; otherwise, end the program.

Variables:

A, A+1, A+2, A+3: correspond with X, Y, A, B in format example,respectively.

A+4: beginning of the program information in the line.

B: loop to skip Modprog lines.

CT: character to change.

LN: current line number.

NP: position of two locations preceding next line number.

TC: character to change.

X: loop to search line.

```
0 DEFINTA-Z:A=PEEK(16548)+PEEK(16549)*256:FORE=0TO2:A=PEEK(A)+PE
EK(A+1)*256:NEXT:INPUT"STATEMENT/CHARACTER TO CHANGE (NUMBER)";T
C:INPUT"STATEMENT/CHATACTER TO CHANGE TO (NUMBER)";CT
1 NP=PEEK(A)+PEEK(A+1)*256:LN=PEEK(A+2)+PEEK(A+3)*256:FORX=A+4TO
NP-1:IFPEEK(X)=TCTHENPOKEX,CT:PRINT"JUST CHANGED LINE";LN
2 NEXTX:A=NP:IFPEEK(A)+PEEK(A+1)=0THENPRINT"ALL DONE!":ENDELSE1
```

TRS-80 Codes Chart

ABS	217	DEFINT	153	INP	219	NAME	169	RSET	172
AND	210	DEFSNG	154	INPUT	137	NEW	187	RUN	142
ASC	246	DEFSTR	152	INSTR	197	NEXT	135	SAVE	173
ATN	228	DELETE	182	INT	216	NOT	203	SET	131
AUTO	183	DIM	138	KILL	170	ON	161	SGN	215
CDBL	241	EDIT	157	LEFT$	248	OPEN	162	SIN	226
CHR$	247	ELSE	149	LEN	243	OR	211	SQR	221
CINT	239	END	128	LET	140	OUT	160	STEP	204
CLEAR	184	EOF	233	LINE	156	PEEK	229	STOP	148
CLOAD	185	ERL	194	LIST	180	POINT	198	STR$	244
CLOSE	166	ERR	195	LLIST	181	POKE	177	STRING$	196
CLS	132	ERROR	158	LOAD	167	POS	220	SYSTEM	174
CMD	133	EXP	224	LOC	234	PRINT	178	TAB(	188
CONT	179	FIELD	163	LOF	235	PUT	165	TAN	227
COS	225	FIX	242	LOG	223	RANDOM	134	THEN	202
CSAVE	186	FN	190	LPRINT	175	READ	139	TIME$	199
CSNG	240	FOR	129	LSET	171	REM	147	TO	189
CVD	232	FRE	218	MEM	200	RESET	130	TROFF	151
CVI	230	GET	164	MERGE	168	RESTORE	144	TRON	150
CVS	231	GOSUB	145	MID$	250	RESUME	159	USING	191
DATA	136	GOTO	141	MKD$	238	RETURN	146	USR	193
DEF	176	IF	143	MKI$	236	RIGHT$	249	VAL	245
DEFDBL	155	INKEY$	201	MKS$	237	RND	222	VARPTR	192

```
]LIST

0 A =  PEEK (103) +  PEEK (104) *
     256: FOR B = 1 TO 4:A =  PEEK
     (A) +  PEEK (A + 1) * 256: NEXT
     : INPUT "STATEMENT/CHARACTER
      TO CHANGE (NUMBER)";TC: INPUT
     "STATEMENT/CHAR CHANGE TO (N
     UMBER)";CT
1 NP =  PEEK (A) +  PEEK (A + 1) *
     256:LN =  PEEK (A + 2) +  PEEK
     (A + 3) * 256: FOR X = A + 4
      TO NP - 1: IF  PEEK (X) = T
     C THEN  POKE X,CT: PRINT "JU
     ST CHANGED LINE ";LN
2  NEXT X:A = NP: IF  PEEK (A) =
     0 AND  PEEK (A + 1) = 0 THEN
      PRINT "ALL DONE!!!": END
3  GOTO 1
```

Mod-Prog in Applesoft

Below is a listing of the Applesoft version of Mod-Prog, which allows you to change any character or statement token in a program to any other character or token. The lines below must appear at 0-3 in your program. The variables used are the same as those in the TRS-80 version. Applesoft tokens appear on page 121 in the Applesoft manual, and the ASCII codes are on pages 138-9 of the same book.

EDU-WARE

UNIQUE SOFTWARE FOR THE UNIQUE MIND.

Windfall

The Oil Crisis Game

With the help of the Apple, this game puts you on the other side of an Oil Crisis . . . as the Chief Executive of Engulf Oil. As a helpless public lines up at your stations, you attempt to turn crisis into cash, making your "Windfall Profits". Huge gains (or losses) accrue from your manipulation of prices, wages, foreign suppliers, stock sales and collusion with the competition.

For Apple II and II-plus computers
Requires 32K (48K with disk and RAM Applesoft)
Cassette $14.95
Diskette $19.95

Compu-read

A series of four programs designed to enhance the user's reading ability. A sequence of increasingly complex skills, from simple character recognition to identification of rapidly displayed words is presented in the first two programs. In the third program, vocabulary practice is provided using synonyms and antonyms as a tool. Finally, user response to high speed sentence presentation builds skills in speed and comprehension.

For Apple II and II-plus computers
Requires 32K RAM and Disk Drive
Disk $24.95

TO ORDER TOLL-FREE
1-800-258-1790 (In NH call 673-5144)

The Software Exchange

Market Basket

SOFTWARE FOR THE TRS-80

TO ORDER TOLL FREE

1-800-258-1790

(In NH call 673-5144)

HOURS:

MONDAY - FRIDAY
9 AM to 9 PM
SATURDAY
11 AM to 3 PM

*EASTERN TIME

✶ GAMES

TRS-80 Opera Theatre
Magnificent sound! by Richard Taylor
Level II, 16K $9.95

Challenge
Word game with sound by Richard Taylor
Level II, 16K $9.95

Opera Theatre Plus Challenge
on disk, 32K $19.95

Bridge Challenger
by Personal Software
Level II, 16K $14.95

Invasion
by Chris Freund
Level II, 16K $9.95
Disk $14.95

X-Wing Fighter II
by Chris Freund
Level II, 16K $9.95

Taipan
by Art Canfil
Level II, 16K $9.95

Sargon Chess
by Dan and Kathe Spracklen
Level II, 16K $19.95

Sargon II
by Dan and Kathe Spracklen
Level II, 16K $29.95
on disk $34.95

Amazin' Mazes
by Robert Wallace
Level II, 16K $9.95

Snake Eggs
With sound by Leo Christopherson
Level II, 16K $14.95

Life Two
With sound by Leo Christopherson
Level II, 16K $14.95

Android Nim
With sound by Leo Christopherson
Level II, 16K $14.95

Bee Wary
With sound by Leo Christopherson
Level II, 16K $14.95

Mastermind II
by Lance Micklus
Level II, 16K $7.95

Tycoon
by David Bohlke
Level II, 16K $7.95

9 Games for Preschool Children
by George Blank
Level II, 16K $9.95

End Zone II
by Roger W. Robitaille
Level II, 16K $9.95

Space Battles
by Level IV
Level II, 16K tape $14.95
32K disk $19.95

Star Trek
by Lance Micklus
Level II, 16K (May 1980 issue) $14.95

Adventures on Tape
by Scott Adams
Level II, 16K $14.95
Choose one:
- Adventureland: Magical beings, perils and puzzles!
- Pirate's Cove: Clue in a bloodsoaked book.
- Mission Impossible: Foil the saboteur, save the reactor!
- The Count: Protect your neck! Who do you think lives here?
- Voodoo Castle: Remove the curse from Count Cristo.
- Strange Odyssey: The ruins of an ancient alien culture?
- Fun House: Takes all your brains to get past the gate!
- Pyramid of Doom: Watch out for the Nomad!
- Ghost Town

Adventures on Disk
by Scott Adams
$39.95 for three-game combinations:
1) Mission Impossible, Pirate's Cove, Adventureland
2) Voodoo Castle, The Count, Strange Odyssey
$24.95 for two-game combinations:
3) Fun House, Pyramid of Doom

Adventure Sampler
by Scott Adams
Mini-version of Adventureland, serves as introduction
Level II, 16K $5.95

Pork Barrel
by George Blank
Level II, 16K $9.95

Barricade
by Small Systems Software
Machine Language $9.95

Pentominoes
by James Garon
Level II, 16K $7.95

Mean Checkers Machine
by Lance Micklus
Level II, 16K tape $19.95
Disk $24.95

Kriegspiel II $14.95

Time Trek
by Joshua Lavinsky
Level II, 16K $14.95

Kamikaze
by Russell Starkey
Level II, 16K $7.95

Wordo
from Microfantastic Programming
Level II, 16K $9.95

Fastgammon
from Quality Software
Level II, 16K $19.95

Cribbage
by Roger W. Robitaille Sr.
Level II, 16K $7.95

Galactic Empire
by Doug Carlston
Level II, 16K $14.95

Galactic Trader (a sequel)
by Doug Carlston
Level II, 16K $14.95

Galactic Revolution
by Doug Carlston
Level II, 16K $14.95

Galactic Trader and Empire
on disk 32K $29.95

All 3 on disk $39.95

Airmail Pilot $7.95

Air Traffic Controller $7.95

Air Raid
by Small Systems Software
Level I or II, 4K $9.95

Lost Dutchman's Gold
by Teri Li
Level II, 16K $9.95

Original Adventure
by Microsoft as played on PDP-10
32K disk $29.95

Pigskin (Football)
by Acorn Software
Level II, 16K $9.95

PR Dogfight
by David Bohlke
Level II, 16K $7.95

Ting Tong
from Acorn Software
Machine Language, with sound
Level II, 16K $9.95

Journey to the Center of the Earth
by Greg Hassett
Level II, 16K $7.95

See SoftSide on cassettes for additional games

Datestones of Ryn
cassette $14.95
disk $19.95

Starfleet Orion
cassette $19.95
disk $24.95

Invasion of Orion
cassette $19.95
disk $24.95

Morloc's Tower
cassette $14.95
disk $19.95

Rescue at Rigel
cassette $19.95
disk $24.95

Temple of Apshai
cassette $24.95
disk $29.95

Interlude
cassette $14.95
disk $17.95

B-1 Bomber $15.00

Nukewar $15.00

Midway Campaign $15.00

North Atlantic Convoy Raider $15.00

Planet Miners $15.00

Pinball
cassette $14.95
disk $20.95

Amateur Astronomy Handbook
cassette $14.95

Computer Bismarck
cassette $49.95

✶ BUSINESS

Inventory System II.3
by M. Kelleher
Improved version $79.95

Inventory 'S'
by Roger W. Robitaille, Sr.
Level II, 16K tape $24.95
32K disk/without invoicing $39.95
with invoicing $59.95

Payroll
by Stephen Hebbler
Disk 32K $39.95

Appointment Log
by Michael Kelleher
Level II, 16K $9.95

Mail List II
by SBSG
32K disk $49.95

Small Business Bookkeeping
by R.W. Robitaille, Sr.
Level II, 16K with journal
disk $36.95
tape $31.95
without journal
disk $29.95
tape $24.95

On Line Invoicing
by Stephen Hebbler
1 disk, manual; 32K, 1 drive, line printer $39.95

Accounts Receivable(with invoicing)
by Stephen Hebbler
3 disks, 2 manuals; 32K, 2 drives, line printer $69.95

CCA Data Management System
from Personal Software
Ready for transfer to disk, with manual
Tape ..... $74.95 + $2

✶ SPECIAL PURPOSE

Histograph/Scattergram
by Gary Breschini
Level II, 16K $9.95

Tiny Comp A BASIC Compiler in BASIC
by David Bohlke
Tape $19.95
Disk $24.95

RPN Calculator
by Russell Starkey
Level II, 16K $9.95

Ham Radio
by Michael Kelleher
Level II, 16K $9.95

Ham Radio
Advanced version for 32K
disk $24.95

Electronic Assistant
by John Adamson
Level II, 16K $9.95

Basic Statistics
by Steve Reisser
Level II, 16K $9.95

Magic Paper Calculator
by Russell Starkey
Level II, 16K $14.95

Keyboard 80 (Improved)
by John Adamson
Level II, 16K $9.95

Text-80
by Frank B. Rowlette, Jr.
32K disk systems $59.95

Scripsit ..... $95 + $2

Dynamic Data Base
by Ken Knecht
Manual and 3 programs on disk for 32K or larger systems with one or more drives $39.95

✶ PERSONAL

Typing Tutor
by 80 US
Level II, 16K $19.95

I Ching/Biorhythms
together $9.95

Secrets of the Tarot
Cards of Fortune
together $9.95

Numerology
by John T. Phillipp
32K disk $14.95

Personal Finance
by Lance Micklus
Level II, 16K $9.95

Advanced Personal Finance
by Lance Micklus
32K disk $24.95

Home Financial Management
by Michael Kelleher
Level II, 16K $9.95

Periodical Cross Reference
by Dave Stambaugh
Level II, 16K $14.95
32K disk $19.95

Roots
by Bill Sholar
32K disk $19.95

✶ UTILITIES

T-Short
by Web Associates
Level II, 16K $9.95

Z80 Zap Command
disk $29.95

IRV
by Robin Mork
Level II, 16,32, and 48K
cassette $24.95
disk $29.95

APL-80 3.0
by Phelps Gates
Deluxe version—32K disk $39.95
Level II, 16K tape—reduced features, no course or book $14.95
Book (separately) ..... $16.95 + $3

File Manager 80
by Nepenthe
32K disk $49.95

Automated Disk Directory
by George Blank
32K disk, requires NEWDOS $14.95

STAD
Trace and Debug Monitor for tape and disk systems 16, 32 and 48K on one tape $24.95

NEWDOS by Apparat $49.95

NEWDOS+ by Apparat $99.95

NEWDOS 80 by Apparat $149.00

RSM2 by Small Systems Software
Level II, 16K $26.95

RSM2D by Small Systems Software
Disk for 16-48K on one tape $29.95

KVP by Lance Micklus
Tape + 232 ..... $14.95
Disk $19.95

ST80 Smart Terminal
by Lance Micklus
Level II, 16K $49.95

ST80D Smarter Terminal
by Lance Micklus
For Disk $79.95

ST80-UC dedicated to THE SOURCE
by Lance Micklus
Level II, 16K $24.95

ST80 III
by Lance Micklus $150.00

Micro Text Editor
by Don Coons
Level II, 4K or 16K $9.95

Level III BASIC
by Microsoft $49.95

Level I in Level II
by Apparat
Level II, 16K $15.00

FORTRAN
by Microsoft
32K, 2 disks ..... $80.00 + $2.50

Assembler
by Microsoft
32K ..... $80.00 + $2.50

FORTRAN Plus Assembler ..... $150.00 + $5

Spool
Will print an ASCII file to a parallel line printer at the same time you are using your computer for another program. For 32K disk systems only. Will NOT work with NEWDOS (2.3 or VTOS 3.0 OK).
Disk $24.95

Cross Reference Utilities
by Lance Micklus
Level II cassette versions
XREF $19.95
RENUMX (XREF with RENUMBER and rescue after reset) $24.95
RENUMBER separately $7.95

RX—Cross Reference for Disk
by Lance Micklus
32K, 1 disk $24.95

RENUMBER
by Lance Micklus
The full-featured RENUMBER program. One version now works in 16K, 32K, or 48K $7.95

System Copy
by Kalman Bergen
Level II, 16K $9.95

Floppy Disk Diagnostic
by Dave Stambaugh
Disk with manual $24.95

✶ BOOKS

Sargon Handbook
by Dan and Kathe Spracklen ..... $15.95 + $1

The BASIC Handbook
by Dr. David A. Lien ..... $14.95 + $1

Z-80 Instruction Handbook
by Scelbi Publications ..... $5.95 + $1

The Little Book of BASIC Style
by John Nevison ..... $5.95 + $1

Pathways Through the ROM
..... $19.95 + $1

TRS-80 Assembly Language Programming
by William Barden, Jr. ..... $3.95 + $1

APL — An Interactive Approach
by Gilman and Rose ..... $16.95 + $3

Learning Level II
by Dr. David A. Lien ..... $15.95 + $1

Z-80 Software Gourmet Guide and Cookbook
from Scelbi Publications ..... $15.95 + $1

TRS-80 Disk and Other Mysteries
by Harvard Pennington ..... $22.95 + $1

TRS-80 Interfacing
by Jonathan Titus ..... $8.95 + $1

Secret Guide to Computers
by Russ Walters ..... $5.95 + $1

Information Management System
P.I.M.S. ..... $11.95 + $1

Calculating with BASIC ..... $8.95 + $1

Understanding Micro Computers
..... $9.95 + $1

Take my Computer Please
..... $5.95 + $1

Learn Micro Computers
..... $14.95 + $1

Micro Computer PotPourri
..... $3.95 + $1

✶ SUPPLIES

Cassettes: Boxes of 10 each
C-10 ..... $6.95 + $1
C-20 ..... $7.95 + $1

Diskettes:
Dysan (premium quality)
Box of 5 ..... $29.95 + $1
BASF
Box of 5 ..... $19.95 + $1.50
Box of 10 ..... $34.95 + $2
Case of 100 ..... $299.00 + $3
Scotch
Box of 10 ..... $39.95 + $2

Diskette Storage Box ..... $5.00 + $1

Floppy Armor™
Protective envelopes for shipping floppy disks.
5 pack ..... $4.95 + $1

SoftSide Vinyl Binders ..... $4.95 + $1

Tape Recorder Alignment Kit ..... $9.95 + $1

Floppy Disk Savers $14.95

Floppy Disk Refills $7.95

- **No sales tax.**
- **All C.O.D.'s or special delivery orders are a minimum of $5.00 for special handling.**

For more details descriptions of our software and supplies send for the TSE Catalog - it's FREE! Write or call today for your copy:

NOTICE
All prices are subject to change without notice. The Software Exchange is not responsible for typographical errors, including prices.

* TRS-80 is a trademark of Radio Shack division of Tandy Corporation - makers of the TRS-80, and Video Genie.

? ? MOVING ? ?

TO CORRECT OR CHANGE YOUR ADDRESS ATTACH LABEL FROM YOUR LATEST COPY HERE AND PRINT NEW ADDRESS BELOW:

Name _____

Address _____

City _____ State _____ Zip _____

MAIL TO: The Software Exchange, PO Box 68, Milford, NH 03055

CONCENTRATION continued from page

```
166 HLIN X,X+4 AT Y+1: HLIN X,X+
    4 AT Y+2: HLIN X+1,X+3 AT Y+
    3: PLOT X+2,Y+4: PLOT X+1,Y:
     PLOT X+3,Y
179 RETURN
180 A$="PICNIC TABLE": GOSUB 820
 182 COLOR=8: HLIN X,X+4 AT Y
 184 PLOT X+1,Y+1: PLOT X+3,Y+1:
      PLOT X+2,Y+2: PLOT X+1,Y+3
     : PLOT X+3,Y+3: PLOT X+4,Y+
     4: PLOT X,Y+4
 199 RETURN
 200 A$="BEER MUG": GOSUB 820
 203 COLOR=15: VLIN Y,Y+4 AT X+2
     : VLIN Y,Y+4 AT X
 205 HLIN X,X+2 AT Y+4: VLIN Y+1
     ,Y+3 AT X+4: PLOT X+3,Y+3: PLOT
     X+3,Y+1
 210 COLOR=13: FOR Z=1 TO 150: NEXT
     Z
 212 FOR I=Y+3 TO Y STEP -1
 214 PLOT X+1,I: FOR Z=1 TO 150:
      NEXT Z
 216 NEXT I
 219 RETURN
 220 A$="SAILBOAT": GOSUB 820
 224 COLOR=2: HLIN X,X+4 AT Y+4:
      COLOR=1: VLIN Y,Y+3 AT X+1

 226 COLOR=15: VLIN Y,Y+2 AT X+2
     : VLIN Y+1,Y+2 AT X+3: PLOT
     X+4,Y+2
 239 RETURN
 240 A$="BED": GOSUB 820
 242 COLOR=8: VLIN Y,Y+4 AT X+4:
      VLIN Y+1,Y+4 AT X: HLIN X,
     X+4 AT Y+3
 246 COLOR=15: HLIN X+1,X+3 AT Y+
     2: PLOT X+3,Y+1
 259 RETURN
 260 A$="TOASTER": GOSUB 820
 264 COLOR=5: FOR I=Y+2 TO Y+4: HLIN
     X,X+4 AT I: NEXT I
 265 COLOR=15: PLOT X,Y+2: PLOT
     X+4,Y+2: PLOT X,Y+4: PLOT X+
     4,Y+4: VLIN Y,Y+1 AT X+1: VLIN
     Y,Y+1 AT X+3
 267 COLOR=0: PLOT X+2,Y+2: FOR
     Z=1 TO 200: NEXT Z
 268 VLIN Y,Y+1 AT X+1: VLIN Y,Y+
     1 AT X+3
 270 COLOR=5: PLOT X+2,Y+2: COLOR=
     0: PLOT X+2,Y+3: FOR Z=1 TO
     550: NEXT Z
 272 COLOR=5: PLOT X+2,Y+3: COLOR=
     0: PLOT X+2,Y+2: COLOR=13: VLIN
     Y,Y+1 AT X+1: VLIN Y,Y+1 AT
     X+3
 274 POKE 0,90: POKE 1,58: CALL
     2
 279 RETURN
 280 A$="BOILING POT": GOSUB 820
 282 COLOR=13: HLIN X,X+2 AT Y+2
     : HLIN X,X+2 AT Y+3
 284 COLOR=8: HLIN X+3,X+4 AT Y+
     2
 286 COLOR=5: GOSUB 296: COLOR=11
     : GOSUB 296: COLOR=9: GOSUB
     296
 287 FOR I=1 TO 125: COLOR= RND
     (2)*15: PLOT RND (3)+X, RND
     (2)+Y: NEXT I
 288 COLOR=0
 289 HLIN X,X+3 AT Y: HLIN X,X+3
      AT Y+1
 290 COLOR=5: HLIN X,X+2 AT Y+4
 293 GOTO 299
 296 HLIN X,X+2 AT Y+4: FOR Z=1 TO
     250: NEXT Z
 299 RETURN
 300 A$="DOOR CHIMES": GOSUB 820

 302 COLOR=8: HLIN X,X+4 AT Y
 305 COLOR=13: VLIN Y+1,Y+2 AT X:
      VLIN Y+1,Y+3 AT X+2: VLIN
     Y+1,Y+4 AT X+4
 310 POKE 0,113: POKE 1,200: CALL
     2
 312 POKE 0,125: POKE 1,200: CALL
     2
 314 POKE 0,118: POKE 1,200: CALL
     2
 316 POKE 0,174: POKE 1,200: CALL
     2
 319 RETURN
 320 A$="CIGARETTE": GOSUB 820
 322 COLOR=15: HLIN X,X+4 AT Y+4
     : COLOR=13: PLOT X+4,Y+4: FOR
     Z=1 TO 200: NEXT Z
 323 COLOR=1: PLOT X,Y+4
 324 FOR K=1 TO 9
 326 COLOR=0: VLIN Y,Y+3 AT X: VLIN
     Y,Y+3 AT X+1: COLOR=5: PLOT
     X,Y+3
 327 FOR I=Y+2 TO Y STEP -1
 330 L=X+ RND (2)
 332 PLOT L,I: FOR Z=1 TO 26: NEXT
     Z
 335 NEXT I
 337 NEXT K
 339 RETURN
 340 A$="SWISS FLAG": GOSUB 820
 344 COLOR=1
 345 FOR I=X TO X+4: VLIN Y,Y+4 AT
     I: NEXT I
 350 COLOR=15: HLIN X+1,X+3 AT Y+
     2: VLIN Y+1,Y+3 AT X+2
 359 RETURN
 360 A$="PACKAGE": GOSUB 820
 363 COLOR=15: FOR I=X TO X+4: VLIN
     Y+2,Y+4 AT I: NEXT I
 365 COLOR=1: VLIN Y+1,Y+4 AT X+
     2: HLIN X,X+4 AT Y+3: PLOT
     X+3,Y: PLOT X+1,Y
 379 RETURN
 820 REM  PRINTS PICTURE NAME
 821 IF ND=2 THEN PRINT
 822 TAB 24: PRINT A$;: RETURN
1000 GOSUB 1990
1010 REM  INITIAL CONDITIONS
1020 FOR I=0 TO 35:CH(I)=1: NEXT
     I: REM  'CLOSE' ALL DOORS
1030 SC(0)=0:SC(1)=0:MTCH=0:TRY=
     0
1040 GOSUB 1600
1050 REM
1060 REM  BEGIN MAIN LOOP
1070 REM
1080 FOR P=0 TO N: REM  P=PLAYER
1090 GOSUB 1270
1100 ND=0: REM   ND=DOOR NO#
1110 GOSUB 1410
1120 W(ND)=PIX(D): REM SAVE EACH DOOR

1130 GOSUB 1790
1140 GOSUB PIX(D)*20+20: REM   PLOT T
     HE PICTURE BEHIND THE DOOR
1150 IF ND=1 THEN 1110
1160 TRY=TRY+1
1170 IF W(1)#W(2) THEN 1200: REM NO M
     ATCH
1180 GOTO 1310
1190 GOTO 1090
1200 FOR Z=1 TO 500: NEXT Z
1210 GOSUB 1870
1220 NEXT P
1230 GOTO 1080
1240 REM
1250 REM  END OF MAIN LOOP
1260 REM
1270 REM  PRINTS PLAYER NAME
1280 CALL -936: VTAB 23: GOTO 1290
     +10*P
1290 PRINT Y$;: RETURN
1300 PRINT Z$;: RETURN
1310 REM  ORANGE FLASH ON MATCH
1320 REM   AND INCREASE SCORE
1330 FOR K=1 TO 3
1340 FOR J=9 TO 0 STEP -9
1350 COLOR=J
1360 FOR I=0 TO 18 STEP 6: VLIN
     0,36 AT 18+I: VLIN 0,36 AT
     18-I: HLIN 0,36 AT 18+I: HLIN
     0,36 AT 18-I
```

```
1370 POKE 0,(I+4)*2: POKE 1,10: CALL
     2
1380 NEXT I,J,K
1390 SC(P)=SC(P)+5:MTCH=MTCH+1: IF
     MTCH=18 THEN 2590
1400 GOTO 1190
1410 REM  GET DOOR THRU PDL'S
1420 REM  GET COLUMN
1430 COL= PDL (P)*5/255:X=COL*6+
     1
1440 COLOR=15: HLIN X,X+4 AT 38
1450 IF PEEK (-16287+P)>127 THEN
     1480
1460 IF PDL (P)*5/255=COL THEN 1450

1470 COLOR=0: HLIN 0,39 AT 38: GOTO
     1430
1480 REM  GET ROW
1490 ROW= PDL (P)*5/255:Y=ROW*6+
     1
1500 COLOR=C(ROW): VLIN Y,Y+4 AT
     38:
1510 IF PEEK (-16287+P)<=127 THEN
1520 IF PDL (P)*5/255=ROW THEN 1510

1530 COLOR=0: VLIN 0,39 AT 38: GOTO
     1490
1540 COLOR=0: VLIN 0,39 AT 38: HLIN
     0,39 AT 38:D=ROW*6+COL: IF
     CH(D)=1 THEN 1570
1550 VTAB 22: TAB 1: PRINT "ALREADY O
     PEN  !!";: POKE 0,255: POKE
     1,255: CALL 2
1560 TAB 1: CALL -868: GOTO 1430

1570 ND=ND+1: IF ND=2 THEN 1590
1580 R1=ROW:X1=X:Y1=Y:D1=D: REM  SAVE
      THE FIRST DOOR
1590 RETURN
1600 REM  PLOT DOORS
1610 REM  ROW COLORS SET BY C( )
1620 C(0)=11:C(1)=4:C(2)=2:C(3)=
     8:C(4)=13:C(5)=3
1630 GR
1640 FOR K=0 TO 35: COLOR=C(K/6)
     : HLIN 0,36 AT K: NEXT K
1650 COLOR=0: FOR I=0 TO 36 STEP
     6: HLIN 0,36 AT I: VLIN 0,36
      AT I: NEXT I
1660 VTAB 21: FOR I=4 TO 34 STEP
     6: TAB I: PRINT I/6+1;: NEXT
     I
1670 POKE 34,21: RETURN
1680 REM  ARRAY G IS FILLED WITH
1690 REM  0-18 (2 OF EACH)
1700 FOR I=0 TO 35:G(I)=I/2: NEXT
     I
1710 REM  A RAND # IS CHOSEN
1720 REM  THEN ARRAY IS SHRUNK
1730 REM  TO DROP # CHOSEN
1740 REM  RESULT IS 18 PAIRS OF
1750 REM  NUMBERS THAT ARE PICTURE
1760 REM   SUBROUTINES IN PIX()
1770 FOR L=36 TO 1 STEP -1:J= RND
     (L):PIX(L-1)=G(J): FOR I=J TO
     L:G(I)=G(I+1): NEXT I,L
1780 RETURN
1790 REM  OPEN DOOR W/SOUND
1800 COLOR=0
1810 FOR I=X+4 TO X STEP -1
1820 VLIN Y,Y+4 AT I
1830 POKE 0,9: POKE 1,8: CALL 2
1840 NEXT I
1850 CH(D)=0: REM  DOOR IS OPEN
1860 RETURN
1870 REM  CLOSE DOORS W/SOUND
1880 PP=6: COLOR=C(R1)
1890 FOR I=X1 TO X1+4
1900 VLIN Y1,Y1+4 AT I
1910 PP=PP+1: POKE 0,PP: POKE 1,
     5: CALL 2
1920 NEXT I:CH(D)=1
1930 PP=6: COLOR=C(ROW)
1940 FOR I=X TO X+4
1950 VLIN Y,Y+4 AT I
1960 PP=PP+1: POKE 0,PP: POKE 1,
     5: CALL 2
1970 NEXT I:CH(D1)=1: REM  DOOR IS CL
     OSED
1980 RETURN
1990 REM  INTRODUCTION
2000 TEXT : CALL -936
2010 PRINT "CONCENTRATION, BY HARRIS
     KIRK"
2060 GOSUB 1680
2460 CALL -936
2470 TAB 13: PRINT ">> PLAYERS <<"

2480 PRINT : PRINT
2490 A$="FIRST PLAYER ": GOSUB 2760
     : INPUT Y$: PRINT
2500 IF LEN(Y$)<19 THEN 2520
2510 GOSUB 2570: GOTO 2490
2520 A$="SECOND PLAYER ": GOSUB
     2760: INPUT Z$
2530 IF LEN(Z$)<19 THEN 2550
2540 GOSUB 2570: GOTO 2520
2550 N=1: IF Z$="" THEN N=0
2560 GOTO 2580
2570 PRINT "NAME TOO LONG !": PRINT
     : REM  CTRL-G
2580 RETURN
2590 POKE 34,20: CALL -936
2600 A$="PRESS RETURN FOR SCORES"
     : GOSUB 2760: INPUT H$
2610 COLOR=0
2620 FOR I=0 TO 4: FOR J=1 TO 31
      STEP 6
2630 HLIN 0,39 AT J+I: POKE 0,250
     : POKE 1,25: CALL 2
2640 NEXT J,I: FOR Z=1 TO 300: NEXT
     Z
2650 TEXT : CALL -936: IF N=0 THEN
     2720
2660 FOR I=0 TO 1:SUM(I)=SUM(I)+
     SC(I): NEXT I
2670 TAB 5: PRINT "PLAYER";: TAB
     19: PRINT "GAME";: TAB 32: PRINT
     "SERIES"
2680 TAB 5: PRINT "------";: TAB
     19: PRINT "----";: TAB 32: PRINT
     "------"
2690 PRINT Y$;: TAB 20: PRINT SC(
     0);: TAB 33: PRINT SUM(0)
2700 PRINT Z$;: TAB 20: PRINT SC(
     1);: TAB 33: PRINT SUM(1)
2710 GOTO 2730
2720 A$="GAME COMPLETED IN ": GOSUB
     2760: PRINT TRY;:A$=" TRIES"
     : GOSUB 2760
2730 GOSUB 1680
2740 VTAB 20: TAB 1:A$="ANOTHER GAME?
      (Y/N)": GOSUB 2760: INPUT
     H$
2750 IF H$="" THEN 1010: IF H$(1
     ,1)#"N" THEN 1010: GOTO 2780

2760 REM  PRINT A$ WITH SOUND
2770 FOR J=1 TO LEN(A$): PRINT A$
     (J,J);: POKE 0,30: POKE 1,23
     : CALL 2: NEXT J: RETURN
2780 CALL -936: PRINT "THANKS FOR PLA
     YING...HAVE A NICE DAY!"
2790 END
2800 REM  POKES FOR A SIMPLE TONE
2810 REM  ROUTINE. SEE THE
2820 REM  APPLE RED BOOK
2830 POKE 2,173: POKE 3,48: POKE
     4,192: POKE 5,136: POKE 6,208
     : POKE 7,4: POKE 8,198: POKE
     9,1: POKE 10,240
2840 POKE 11,8: POKE 12,202: POKE
     13,208: POKE 14,246: POKE 15
     ,166: POKE 16,0: POKE 17,76
     : POKE 18,2: POKE 19,0: POKE
     20,96: RETURN
```

Watch for the S-80 version of CONCENTRATION next month . . . .

GRAND PRIX continued from page 50

```
10800 IFPEEK(LC)=BLANDPEEK(LC-1)=BL AND PEEK(RC)=BLANDPEEK(RC+1)
=BL GOTO11000
10810 T=T+20: IF PEEK(LC)<>BLORPEEK(LC-1)<>BLTHEN LEFT=0 ELSE LE
FT=-1
10812 L1=PEEK(CAR)+640: IF LEFT THEN L1=L1-17
10815 PRINT@L1," xCRASHx";
10820 IF NOT(LEFT) THEN GOSUB20000:GOSUB20010:GOSUB20000:GOSUB20
000 ELSE GOSUB20010:GOSUB20000:GOSUB20010:GOSUB20010
10830 FORL=1TO50:NEXTL
10850 PRINT@L1,CHR$(201);
11000 NEXTI
11010 T=T+DX: IFLP<TKTHEN11200
11020 T1=T/SC:T2=T-T1xSC:PRINT@965,TB$;:PRINT@1010,;:PRINTUSING"
##:##.#";T1,T2/DX;
11030 IFLP=TLTHENPRINT@RPS-15360,STRING$(5,153)"FINISH"STRING$(5
,166);
11200 NEXT LP
11205 FORI=1TO10:PRINT@0,CHR$(255);:Z=USR(0):NEXTI
11210 IFT<TB THEN ME$="YOU BROKE THE LAP RECORD !":M=470:GOSUB21
000:GOTO11280
11220 IFT-TB>200 PRINT@464,"YOUR DRIVER'S LICENCE HAS BEEN CANCE
LLED!":GOTO11280
11230 IFT-TB>100 PRINT@464,"PERHAPS YOU NEED A WIDER TRACK":GOTO
11280
11240 PRINT@464,"YOU'VE QUALIFIED IN ";
11250 PLACE=INT((T-TB)/5)+1:IFPL=1THENPRINT"1ST";ELSEIFPL=2THENP
RINT"2ND";ELSEIFPL=3THENPRINT"3RD";ELSEPRINTPLACE"TH";
11260 PRINT" PLACE";
11280 T=0
11290 FORI=1TO1000:NEXTI
11300 PRINT@528,CHR$(31);"HIT ENTER TO TRY AGAIN OR"
11305 PRINT@592,"X  TO MOVE TO ANOTHER CIRCUIT";
11310 Y$=INKEY$:IFY$=CHR$(13)THEN10570ELSEIFY$="X"THEN10500ELSE1
1310
12000 IF B=32 THEN L1=LC-15360-6:LEFT=-1 ELSE L1=RC-15360+1:LEFT
=0
12005 L1=570+PEEK(CAR)
12010 PRINT@L1,"OOOPS!";: FORL=1TO50:NEXTL: PRINT@L1,CHR$(198);
12020 IF LEFT THEN12500
12030 IFPEEK(LC)<>BLORPEEK(LC-1)<>BLTHEN12600
12040 GOSUB20010:GOTO12030
12500 IFPEEK(RC)<>BLORPEEK(RC+1)<>BLTHEN12600
12510 GOSUB20000:GOTO12500
12600 T=T+10
12620 GOTO10812
15000 PART=PART+1:IFINT(PART/2)x2<>PART RETURN
15050 ONPART/2GOTO16000,16025,16030,16040,16050,16070
15060 RETURN
16000 PRINT@25,"xx GRAND PRIX xx"
16010 PRINT@192,"YOU ARE ABOUT TO TAKE PART IN THE QUALIFYING SE
SSION"
16020 PRINT"OF AN INTERNATIONAL GRAND PRIX RACE."
16022 RETURN
16025 PRINT"YOUR 'FORMULA ONE' CAR IS CONTROLLED BY THE ARROW KE
YS "CHR$(93)" AND "CHR$(94)
16027 RETURN
16030 PRINT"YOU WILL TRY TO TURN IN THE FASTEST LAP KEEPING IN M
IND THAT :"
16035 RETURN
16040 PRINT"     - EVERY TIME YOU STEER YOU LOOSE 2/10 OF A SECO
ND"
16042 PRINT"       SO YOU SHOULD DRIVE CLOSE TO EDGE OF THE TRAC
K."
16045 RETURN
16050 PRINT"     - IF YOU LEAVE THE TRACK YOU'LL BOUNCE BACK IN
AND"
16060 PRINT"       LOOSE 2 SECONDS"
16065 RETURN
16070 PRINT"     - IF YOU CHANGE STEERING DIRECTION TOO SUDDENLY
 YOU"
16080 PRINT"       WILL SKID, LEAVE THE TRACK AND LOOSE 1 SECOND
."
17000 RETURN
20000 Z=USR(2):POKE CAR,PEEK(CAR)-TWO:LC=LC-TWO:RC=RC-TWO
20008 RETURN
20010 Z=USR(1):POKE CAR,PEEK(CAR)+TWO:LC=LC+TWO:RC=RC+TWO
20020 RETURN
21000 FORK=1TO10:PRINT@M,CHR$(192+LEN(ME$));:FORI=1TO50:NEXTI:PR
INT@M,ME$;:FORI=1TO25:NEXTI,K:RETURN
```

MASTER'S GOLF continued from page 39

colors

```
900 GRAPHICS 5:COLOR 1:SETCOLOR 0,12,6:S
ETCOLOR 4,14,8:SETCOLOR 2,0,12:POKE 752,
1:SETCOLOR 1,12,10
```

T1 is location of tee

```
910 T1=INT(RND(1)*32)+4
```

G1,G2 is location of green; R1 is radius of green

```
912 G1=INT(RND(1)*55)+20:G2=INT(RND(1)*3
0)+5:R1=INT(RND(1)*2)+3
```

conditions for tee, green

```
913 IF (G1<45) AND (ABS(T1-G2)>10) THEN
910
914 IF RND(1)<0.1 THEN 920
916 IF ABS(T1-G2)<20 THEN 910
```

subroutine 950-983 plots the fairway

```
920 GOSUB 950
```

plots green, YA is yardage to green

```
922 FOR X=0 TO R1
923 Y=SQR(R1*R1-X*X):COLOR 2
930 PLOT G1-X,Y+G2:DRAWTO G1-X,G2-Y
932 PLOT G1+X,G2+Y:DRAWTO G1+X,G2-Y
934 NEXT X:YA=G1*5+ABS(G2-T1)*4
```

plots tee, cup

```
936 FOR X=T1-3 TO T1+3:PLOT 0,X:DRAWTO 5
,X:NEXT X
938 COLOR 1:PLOT G1,G2
```

par for hole

```
939 P(HL)=3:IF YA>210 THEN P(HL)=4
940 IF YA>480 THEN P(HL)=5
```

plots sandtrap, returns

```
942 B=INT(G2-T1)/1.5+T1:A=G1-10-RND(0)*7
:IF G1<29 THEN A=G1-10
944 COLOR 3:PLOT A,B:DRAWTO A,B+2:PLOT A
+1,B-1:DRAWTO A+1,B+3:PLOT A+2,B-2:DRAWT
O A+2,B+4
946 PLOT A+3,B-1:DRAWTO A+3,B+3:PLOT A+4
,B:DRAWTO A+4,B+2
949 RETURN
950 IF G1<35 THEN B1=RND(1)+3
952 IF G1>34 THEN B1=RND(1)+1.8
954 B1=INT(G1/B1)
960 F1=T1-4:F2=T1+4
962 FOR I=0 TO B1:PLOT I,F1:DRAWTO I,F2
964 IF RND(1)<0.1 THEN F1=F1-1:IF F1<0 T
HEN F1=0
966 IF RND(1)<0.1 THEN F2=F2+1:IF F2>39
THEN F2=39
968 SOUND 0,I*2,10,2:NEXT I
970 PLOT B1,F1:DRAWTO B1,F2:SOUND 0,F1+F
2,10,2
971 B1=B1+1:IF B1>79 THEN 983
972 IF B1>G1+5+RND(1)*8 THEN 983
975 IF F1>G2-8 THEN F1=F1-1
976 IF F1<G2-8 THEN F1=F1+0.5
977 IF F1<0 THEN F1=0
978 IF F2>G2+8 THEN F2=F2-0.5
979 IF F2<G2+8 THEN F2=F2+1
980 IF F2>79 THEN F2=79
981 GOTO 970
983 RETURN
```

GRAND PRIX TAKE APART continued from page 51

LINES 10440-10548 - CIRCUIT SET UP

The DATA list in statement 10440 is read into array C; each element of the array contains the number of positions the upper part of the track has to be shifted, on each update, to draw a curve. As an example, a curve to the right is formed by printing the top part of the track at position X + C (0) followed by one line down scroll, then at position X + C (1) . . . and so on. TM is the index of the last element of the array LAP (50) which, as explained earlier, contains the sequence of curves and straight lines. In other words, TM represents the circuit length. The time to beat, TB, is calculated according to the minimum number of shifts the car must make (+ a tolerance factor of .1) to reach the finish line. Here the order in the sequence of curves is very important, for two successive curves in the same direction require more shifts than, let's say, a right curve followed by a left.

LINES 10550-11200
MAIN PROGRAM CYCLE
Here are some of the variables used:

CH$
: clears the top line of the screen.

CAR
: absolute address of the byte containing the position of the car on the screen.

RCRASH, LCRASH
: right and left "crash spots"

RPS
: position of track on top line; modified by array C when drawing curves.

RO
: Character forming current line of track. The effect of the track receding in a straight line is created by alternating character 132 (1 dot) with character 145 (2 dots). For curves, character 132 alone is used.

RD
: indicates whether character in RO is to be modified.

TK
: First of the last 5 elements in array LAP. When counter LP reaches TK, the current time T and the time to beat TB are displayed at the bottom of the screen.

Although the role of some of the variables in this program may seem a bit confusing at first, a closer study of the listing should clarify all your doubts and perhaps reveal some programming techniques that you can apply in your own programs.

I hope you have as much fun playing with this program as I had in writing it.

by Max Chauvet

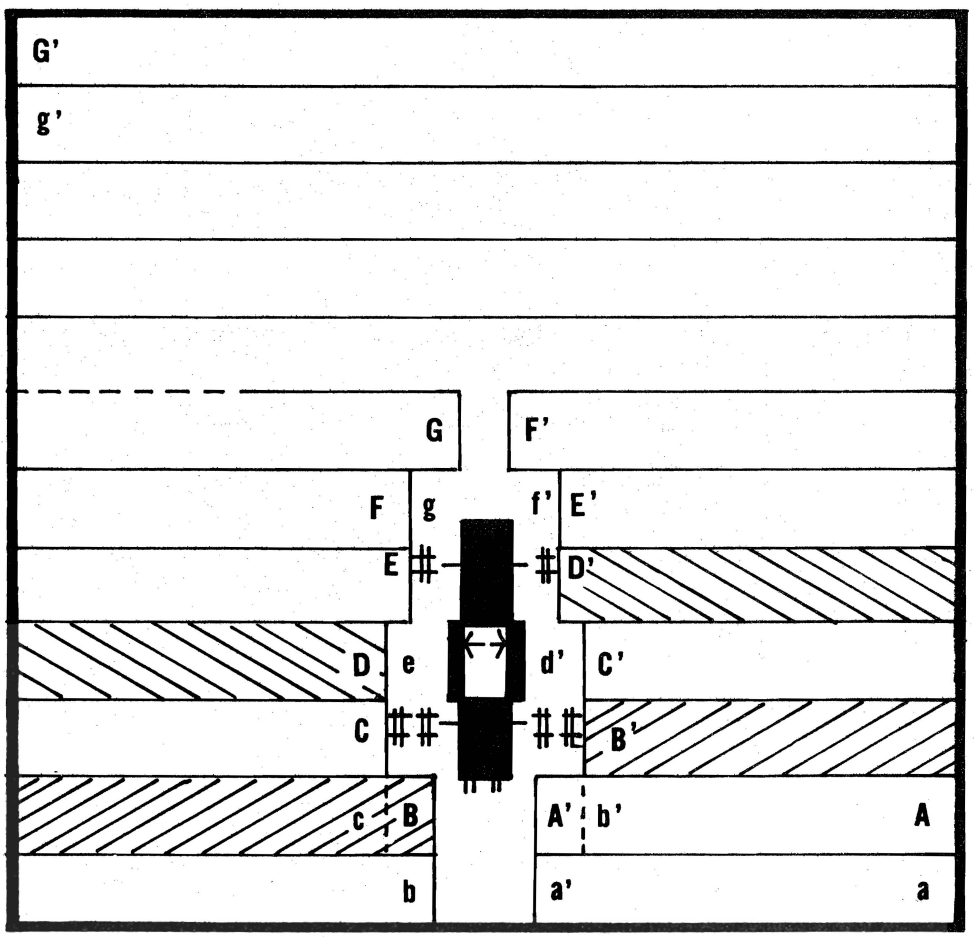


IMAGE
COMPUTER PRODUCTS

STRATEGY PACK I 6404

Roman Checkers. This ancient game has been a favorite for hundreds of years. It couldn't be easier to play, yet playing the game well takes skill, cunning, and strategy as you try to out-think your opponent.
Frame up. Try to out-manuever your opponent or play against the computer in this game of wits and calculated strategy. You will alternate selecting numbers and controlling your opponents choices. Joysticks are optional.
2 programs on cassette..... $29.95

STRATEGY PACK II 6405

Target Chase. Choose a partner or challenge the computer to this fast action, catch-or-be caught adventure.
Tunnels. In this electronics hide-and-seek, only the computer knows where you will appear next.
Survival. Intercept your opponent before you crash or are devoured.
Snake Hunt. Fast maneuvering and quick response are essential to capture - or escape from - your opponent. Joysticks are required.
4 programs on cassette..... $29.95

SKILL BUILDER I 6406

Bingo Duel. This fast-action skill game for one or two players provides an exciting challenge, because young children and adults can compete equally. The computer adjusts to match your skill and problems are specifically selected to help you gain speed.
Number Hunt. Matching numbers is easy enough for young children, yet this computer game quickly advances in difficulty to challenge the experts. Joysticks are required.
2 programs on cassette..... $29.95

WALL STREET CHALLENGE 6402

This computer simulation of the stock exchange is easy to play and always challenging. Invest in several corporations ranging from Municipal Power and Light, a blue chip stock that usually provides steady growth, to Offshore Industries Limited, a high-flying speculative stock that is certain to change often. Stock charts, and the Dow Jones show you the trends. Both 8K and 16K memory versions are included.
Cassette $19.95

ALL STAR BASEBALL 6401

This computerized version of America's favorite sport is full of fast action fun. Each player takes a turn at bat while the other player is both pitcher and the outfield. Innings, score, batting averages, and even errors are calculated by the computer and displayed on the official scoreboard. Balls, strikes, and all plays are called with absolute precision. Joysticks are optional for both players. Both 8K and 16K memory versions included.
Cassette $19.95

MIND MASTER 6403

This classic strategy game takes on a new dimension as the computer designs the hidden problems and reports the results of each guess. Multiple players may compete against the computer and each player may select the level of difficulty that matches their skill, ability, and patience. This program also contains a formula for solving logic problems. Create the answer and watch the computer use deductive logic to discover the secret code. Both 8K and 16K versions.
Cassette $19.95

TO ORDER TOLL FREE
1-800-258-1790
(In NH call 673-5144)

IMAGE
COMPUTER PRODUCTS

Mr. Roger Robitaille, Publisher:

We may be the first to complain, but I'll bet we won't be the only ones to take exception to the art work on page 9 of the May issue and page 25 of the June 1980 issue of the S-8ighty magazine. If you want to bite one of the hands that feed you, then you have a lot to learn about the feelings of anyone who owns a computer, as we all think our system is the greatest, whatever make it may be.

We take exception to the insult of the garbage can with the frowny face inside, next to "SoftSide: S-80". We will not resubscribe to SoftSide, the cassette program, Prog 80, or buy any more software from the Software Exchange, and SoftSide, until an apology is published and this "emblem" is no longer used.

John R. and John D. Lindemann
Cincinnati, Ohio

Thanks for your concern. Radio Shack is very sensitive about their trademarks, and we have been searching for a readily recognizable symbol for products that can be used with the Model I. We are open to suggestions. P.S. It is a trash can, not a garbage can.

Dear Mr. Blank,

First, I would like to thank you for the best computer magazine ever.

I was very excited to see Lance Micklus' 3.4 version STARTREK program in your May issue of SoftSide. To save a lot of typing, I loaded my 3.3 version of STARTREK. Then after entering the new lines, I just renumbered the program to match the 3.4 version. This made it easy to check the GOTO's, and add any data needed to a line. Then the program was up and running, but it was missing something. Edit line 3180 to this and you will have a 50% better program (that's 150% total!): 3180 For K equals 0 to 4: print at 512, B$: for J equals 0 to 100: OUT 255,1: OUT 255,2: next J. Hook the "aux" wire to an amplifier to add a buzzing when you enter yellow, or red alert. (You can also add this to give you sound when firing phazers, etc.)

Just a little sound to a super program. Keep up the great work.

Roy K. Brookins
Beaufort, S.C.

Subject: Apple Computer - Complaint

Folks, I won't waste much of your time. Just thought you might want to know about this and perhaps pass the word.

Since buying my Apple at the end of last year, I've written Apple nine times regarding warranty registration, the "Contact" updates and the "Apple Orchard." In essence, I would like them to acknowledge that I am on record there somewhere.

No response is forthcoming. I've called them and even talked to them at the NCC and got a promise of action, but no results to- date. Some of the letters I sent had a S.A.S.E. enclosed, and the last letter was certified and they had to sign for it. Inside I warned that if they didn't respond, I would write every club, magazine, etc. I could find and announce it on the Source. Still no response.

So far, the computer has behaved perfectly. I hope it never needs service....Thank you for your time.

Sincerely,
Tom Wade
Instant Screen Service
San Diego, CA

Follow-up letter

You folks reported that Radio Shack's Model II was having problems. That took guts - I haven't heard anyone else even HINT such a thing.

See if you find (as I have) that this communication problem with Apple, Inc. is common. People should know these things before they buy.

Thanks,
Tom

By POKEing the final character (line 2050), we avoid the scrolling.

The concept of "fooling" a string into being something different is quite useful. It is used in Mr. Jensen's PINBALL (July S-80 SoftSide) to cause lines of graphics to advance toward the paddles. MONSTER MAZE and ESCAPE, by Joel Mick (to be published in an upcoming issue) both use this approach to save a picture of a randomly generated maze, for later display. It looks like VARPTR-fiddling is a technique whose time has come.

for your Apple

CCA DATA MANAGEMENT SYSTEM

PERSONAL SOFTWARE

- Complete, manual in a three ring binder
- Handles 1 or 2 disk drives
- Written in BASIC, with instructions for writing your own programs to use DMS files, including I/O routines.
- Create, add, update, or delete files easily.
- Search any field for any data.
- Sort by any field.
- Print reports, selecting any fields in any order, and maintain totals and subtotals on any numberic fields.
- Print mailing labels.
- Print records selectively, for example, only customers with New York addresses.
- Complete index in manual
- Scan function to report statistically on file. You might ask how many accounts are overdue and receive a report that started 10 records found in 200 records scanned (5% of file).
- Automatically tests line printer ready state to avoid system lockup.
- Printer alignment pattern for labels.
- Warning issued if you try to compact data without creating a backup.
- On Disk (32K) Applesoft ROM.
- Works in conjuction with Visicalc. $99.95 plus $2 shipping.

VISICALC

Now you can dispense with pencil, paper and calculator — instead you can have an "electronic sheet" to do your calculations, projections, and planning!

Calculate sales projections, income taxes, personal budget, cost estimates, engineering change — even balance your checkbook. Visicalc is partitioned into a matrix of rows and columns. At each position in this matrix you can define a title, formula, or number. By writing on your "electronic sheet" you can set up individualized charts, tables, and records. For example, you can design your own sales forecasting format to assist in making the important "What if?" business decisions.

What would happen if sales increased by 10%?

What if delivery time were decreased by 2 weeks?

What would be the result if I produced 500 widgets this month instead of 600? What if I produced 700?

Let **Visicalc** assist you in performing these calculations and save you countless hours.

Visicalc requires 32K, Applesoft and disk drive. $150.00

TRS-80 and Atari versions available soon!

TO ORDER TOLL FREE

1-800-258-1790

(In NH call 673-5144)

The routine at line 10 uses SET and single precision variables. Lines 100-150 also use SET, but with integer variables. Lines 1000-1050 use single precision POKEs, and finally, the routine at line 10000 uses integer variables combined with POKEs.

The four methods were timed before and after compiling. The times (in seconds) are given below:

METHOD	BEFORE	AFTER
1	52	43
2	42	3
3	9	3
4	8	0

Of course, integer POKEs didn't really take zero seconds, but the screen was whited-out so fast that the only thing certain about the time is that it was MUCH less than half a second; if you blink at the wrong time you will miss it.

ADDITIONAL FEATURES

In addition to compiling most Level II or Disk BASIC programs, the Microsoft compiler adds many more features. Notice lines 1000 and 1010 in the above program. Level II would treat the variables VIDEO and FINISH as if they were VI and FI. The compiler lets you use long variable names and does NOT ignore all characters after the second one.

You can now make liberal use of REMark statements, since they will not be compiled. Thus your source program can be extensively documented without paying the price in either memory or time in the final object program.

All of the advanced math functions, including LOG, EXP, SIN, COS, TAN and the up-arrow can now produce answers which have double precision accuracy.

Assuming you have two disk drives, here's how you would compile a program. (The procedure for compiling on a one-drive system is similar - it merely involves some disk-swapping.) First you write the source program. Let's refer to it as TEST/BAS. Since you are writing in Disk BASIC, you do not need to learn a new language. For editing purposes you use the familiar EDIT command. Once TEST/BAS is written and RUNning to your satisfaction, you must SAVE it on disk in ASCII format.

Type:

SAVE"TEST/BAS",A

You now return to DOS by typing CMD"S" or pressing the reset button. There are two diskettes supplied with the compiler. Place diskette 1, which contains the compiler and something called a "linking loader" in drive 1. To test for errors which might prevent a successful compilation, type:

BASCOM = TEST

BASCOM is the name of the compiler. It will search for the disks for a BASIC source program called TEST/BAS (no need for you to tell it "/BAS"; it knows) and test it for errors. Hopefully you will soon be told that there are no "fatal" errors. If not, BASCOM will show you where in your program the difficulty is. You must correct the problem in your source program and try again. Once BASCOM signifies that the source program is error-free, it is time to begin the actual compiling.

This time, type

BASCOM TEST,TEST=TEST

This will produce two new files on disk: TEST/REL and TEST/LST. The first one, TEST/REL, is the first stage of the compiled program, but it is not quite finished. The second file, TEST/LST, is a listing of your source program lines along with the partially translated portion of your machine language object code. Now type:

L80 TEST

This runs the linking loader, which pinpoints the places in TEST/REL which BASCOM couldn't handle. After producing a list of these places, L80 stops. Now it is time to use the second diskette provided with the compiler. This diskette contains an enormous (38 grans!) file called BASLIB which is a library of all routines that might be needed to complete BASCOM's job. Type:

BASLIB-S

The "-S" is called a "switch". It gives additional information to the compiler. The various switches are explained in the manual. Now it is time to stand up and stretch while the huge BASLIB library is searched. This usually takes a few minutes. When BASLIB is through, type:

TEST-N-E

(-N and -E are two more switches.) In a few moments, another file will be written to disk; this one is called TEST/CMD. TEST/CMD is the long-awaited compiled version of TEST/BAS. TEST/CMD is self-contained - that is, it does not need any of the compiler programs to run properly.

To use TEST/CMD at any time, simply type:

TEST

from DOS.

There are some important things to be aware of. For one thing, the compiled version of a program will be much larger than the original. A program which took one gran of disk storage in its BASIC form required nine grans in its compiled form. Just out of curiosity, I compiled a one-line program:

10'

When compiled, it needed 3 grans of disk space! (A gran is 1280 bytes.)

Another thing to be aware of is that Microsoft expects you to pay them a royalty if you sell compiled versions of your programs. You have your choice of a flat yearly rate of $195.00 or 9 percent of your total sales. Of course, if you only use the compiler to speed up your personal programs, and do not sell them, you do not need to pay Microsoft. If you do choose to sell compiled programs, the fee is certainly worth it if you have a reasonable sales volume. Your programs will run faster and will be more difficult to copy. Any special BASIC code you have developed will not be exposed to the public eye.

All in all, the Microsoft Compiler package (2 diskettes and manual) is well worth the price.

TSE has the Microsoft Compiler in stock for 48K Disk TRS-80 systems. Price is $195.00 plus $3.00 shipping and handling.

A Review of Program Line Editor from Synergistic Software (Apple)

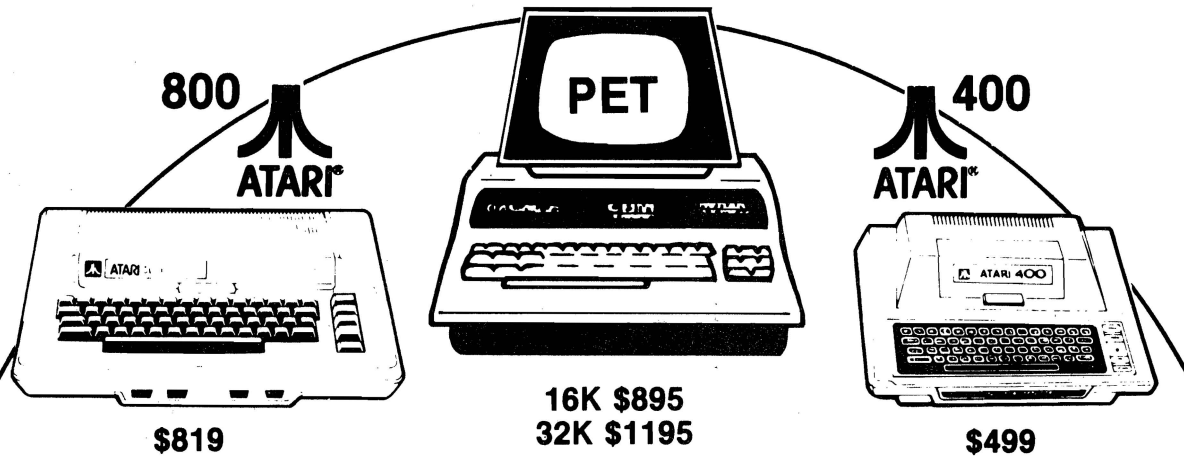
It's going to be difficult to keep this review from sounding like an advertisement. The Program Line Editor is the handiest program I've yet to find for the Apple. Written by Neil Konzen, Program Line Editor is a machine language program that works with any Apple. The package contains a HELLO program that will put the Editor in place as soon as you

boot your disk system. Once there, it remains invisible to most normal operations unless you need it.

Its first function, and the one you might expect, is in editing a program. Just press Control-E and the line number you wish to edit and you can insert, replace, or delete characters with ease. Delete a string of characters? Move the cursor to a certain character immediately? Type upper and lower case for an adapter? All of the above. But the power that isn't apparent from the title lies in a little key in the upper left corner of the keyboard that will soon wear through to the contact points when you get this program. The Program Line Editor also allows you to create 'Escape' functions. These are the things that allow you to type 'ESC 1' and magically get a catalog from drive 1. You can set 'ESC H' to compute and print HIMEM, 'ESC L' for LOMEM. Define functions to correspond to any key you desire. The functions can be words, commands, or sets of commands; anything that you type often can be reduced to a matter of two keystrokes. The package, which includes excellent documentation, costs $40 and works with any Apple that has a disk.

Mini-Review-
Monty from Ritam Corporation (S–80 and Apple)

Monty, from Ritam Corporation, is a program that makes a Monopoly player out of your computer. Available for both the Apple and TRS-80, not only does it play a good game of Monopoly, it also is quite entertaining. You play with a regular Monopoly set, and Monty will play along with whomever else happens to be playing. The graphic displays are very well done (yes, Monty even has an almost life–size head behind that screen), and he also has a little tune that goes with every space on the board. Part of the fun is just trying to figure out the connections between the songs and the spaces; for example, 'Hail to the Chief' goes with — –Pennsylvania Avenue!



$819

16K $895
32K $1195

$499

APPLE*

MICROSOFT ADVENTURE
The original of ADVENTURE written for the DEC-10 systems is now available for the APPLE. Explore Colossal Cave for treasures while avoiding the dangers hidden within its many passages. 130 different rooms. 15 treasures, and characters ranging from helpful to deadly await you within the cave. Be careful where you step, and also who's behind you!
32K disk machine language $29.95

ANDROID NIM
by Leo Christopherson
The game that made Leo Christopherson famous is now available for the APPLE! The improved graphics and color of the APPLE make the game even better. Try to be the last one to shoot the androids on the screen. If you do, you win! Also includes realistic sound effects.
24K cassette machine language.. $14.95

MAGIC PAINT BRUSH
Hi-Res graphics package plus! Draw Hi-Res pictures using all APPLE's colors. Connect any points on screen, fill areas, plot, rotate, and scale shape, or 'paint' with a set of 9 brushes. Also comes with Shape Table Designer and 2 demo programs. Slot Machine and Applesoft Invaders.
32K disk Applesoft-ROM $29.95

THREE-D
You don't have to be an engineer or scientist to have high resolution graphics for your computer! This program permits rotation, scaling, shift, distortion, and combination of three dimensional graphics on the screen. MP Software.
48K disk Applesoft-ROM $29.95

ATARI*

STAR RAIDERS
The best! A ROM cartridge holds the game. A fast-paced full-color, space battle in which you must defeat the enemy Zylon ships while protecting your home bases. Real-time action and effects make this game the best space game available. Sixty levels of rating from Garbage Scow 4th Class to Commander make for continuously exciting play.
ROM cartridge $59.95

3D GRAPHICS
by Tim Hayes
High quality graphics program for the ATARI computer allows you to rotate, distort, shrink, and combine three dimension graphic projections on the screen. With the high resolution abilities of the ATARI, one of the finest graphics packages available anywhere!
16K cassette $29.95

WALL STREET CHALLENGE 6402
A computer simulation of the Stock Exchange is easy to play and always challenging. Invest in stocks, and try to make it big!
8K and 16K version on one cassette
..... $19.95

ALL STAR BASEBALL 6401
Two players face each other, one at batand the other pitcher and outfield. Innings, balls, strikes, and a variety of plays make for an exciting game. Joysticks are optional.
8K and 16K versions on one cassette.
..... $19.95

PET*

STARFLEET ORION
Command a starfleet! 2 player game system includes rule book, battle manual, control sheets. Two programs. 22 space ship types and 12 play tested scenarios.
8K cassette $19.95

RESCUE AT RIGEL
Search the moon base and rescue Delilah Rookh from the High Tollah. Automatic Simulations.
24K cassette $19.95

MORLOC'S TOWER
Match wits with the evil wizard and try to defeat him! Automated Simulations.
32K cassette $14.95

TIME TREK
by Brad Templeton from Personal Software. Real time action. Star Trek type game with sound effects. There are no 'turns'. The action continues whether you move or not. You and the Klingons can move, steer, and fire at the same time.
8K cassette $14.95

This is only a very small sample of our product line. For a complete selection, send $1 for our catalog of hardware, software, and publications and receive a $2 credit toward your first order.

APPLE, ATARI, and PET are trademarks of Apple Computer Co., Warner Communications, and Commodore, respectively

The Software Exchange 6 SOUTH ST., MILFORD, NH 03055

To order: Call Toll-Free 1-800-258-1790 (In NH call 673-5144)

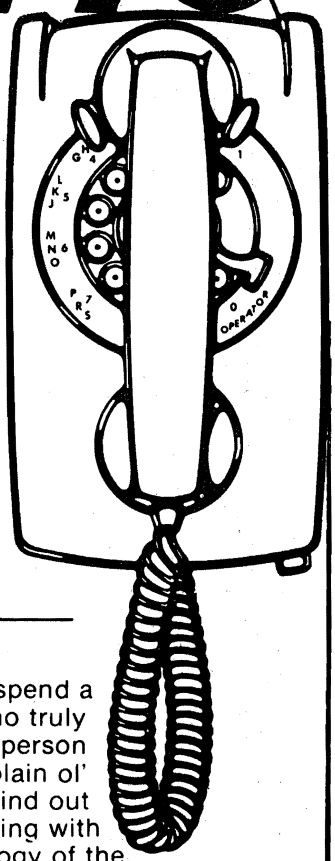
The Software Exchange & Hardside (Div. of Robitaille & Sons, Enterprises, Inc.), SoftSide Publications

master charge VISA

Where can you dial one
TOLL FREE number and be
able to select the finest software
from the world's foremost suppliers?

1-800-258-1790

The Software Exchange

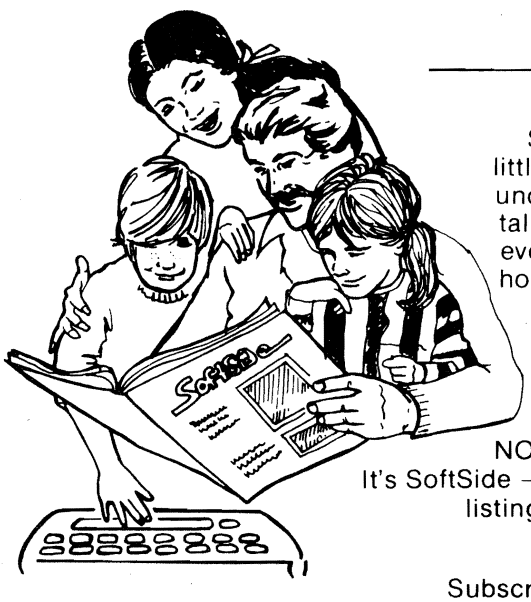


Our stockrooms are overflowing with the finest software available in today's marketplace. We carry software from all these companies:

- **Racet**
- **Adventure International**
- **Personal Software**
- **Web Associates**
- **Small Systems Software**
- **Sensational Software**
- **Synergistic Software**
- **Strategic Simulations**
- **Lance Micklus**
- **Softape**
- **Quality Software**
- **Image Products**
- **Acorn Software**
- **Hayden**
- **Muse**
- **Microsoft**
- **Apparat**

We could not possibly list all of the software on this page, however, if you send us $1.00, you'll receive our catalog plus a $2.00 credit toward your next purchase!

TSE



Suppose you and your family could spend a little time each month with someone who truly understands microcomputers. And this person talked to you about your computer in plain ol' everyday English. Suppose you could find out how these people are successfully dealing with the ever-increasing technology of the microcomputer. Suppose too, that each article was geared to the person with a beginning or intermediate interest in personal computing.

NOW you can get all this, and more, at a price you can afford. It's SoftSide — the magazine that provides its readers with tried and true listings, along with structured "walk-thrus" of some of the most stimulating software pieces, month after enjoyable month.

Subscribe today! Until November 1, 1980, the price is only $15.00. After November 1, the price will be $18.00, so why not save $3.00 and start your family on the road to microcomputing?

A SoftSide Publication

The Software Exchange 6 SOUTH ST., MILFORD, NH 03055

To order: Call Toll-Free 1-800-258-1790 *(in NH call (603) 673-5144)*

VISA

master charge

The Software Exchange & HardSide (Div. of Robitaille & Sons Enterprises, Inc.), SoftSide Publications

S-80 PROGRAMMING TIPS
continued from page 13

The up-arrow method of raising a number to a power is sometimes subject to rounding error.

On my computer (16k Level II) it says that 16 raised to the third equals 4095.998046875 when obviously it equals 4096. These types of errors can be dangerous in a program that uses large numbers. An alternative approach is to start your program with a subroutine that fills a matrix with the values. For example:

```
10 DEFDBL D:DIM
D(20):D(0)=1:FORX = 1 to 20:
D(X) = D(X-1)*16:NEXTX
```

This loads the D matrix with the values of 16 raised to the numbers 0-20. To calculate different powers simply change X or the 16.

Robert W. Lord
Rajeev Jayavant

In the 32 character mode, only messages that start at even-numbered PRINT positions will be displayed; those PRINTed at odd-numbered locations will not appear on the screen.

Example: ?@513,"THIS IS AN EXAMPLE" will not appear on the screen if the video display is in the 32 character mode.

How many times have you tried to BACKUP a diskette only to receive the message "BACKUP REJECTED, DIFFERENT PACK IDS?" The PACK ID is the name of the diskette, such as TRSDOS. To get by this problem, get the name of the original diskette from the DIRectory, then FORMAT the new diskette with the same name. Then BACKUP the new diskette. You should no longer have any problems.

For those of you who write programs for SoftSide and also have Level III BASIC from Microsoft, I suggest the following. Since SoftSide now only accepts programs for publication which are compatible with both Level II and Disk BASIC, if your program makes use of a USR routine, you can use Level III to debug your DEFUSR initialization of machine code. This works because Level III and Disk BASIC use the same technique to initialize USR calls.

Recently, I have noticed the use of a new kind of sound effect in some programs. The sound is created by rapidly toggling the cassette relay. I have talked with the repair people at HardSide and I've been informed that failure or fusing of the cassette relay is one of the most common problems with the TRS-80. It is not a good practice to use any program which toggles this relay for sound as the relay gets overheated very easily and, given a little time, WILL fuse solid. This will prevent the computer from being able to stop the cassette recorder upon completion of a CSAVE, CLOAD or any other tape operation. Put very simply, if you want sound, use an amplifier: it won't damage your computer.

Phil Case
Springfield, MO

ST80* has a smart terminal package for you!

TO ORDER TOLL FREE
1-800-258-1790
(In NH call 673-5144)

All four programs include the ability to use an unmodified TRS-80 keyboard to produce RUB, ESC, and other control characters for time sharing, software control of the RS-232-C board, repeat key, bell, software support for the three most common upper/lower case hardware conversions, and line printer output.

ST80*UC

Preset parity, word length, and baud rate (regardless of switch settings on the RS-232-C board) for THE SOURCE, MICRONET, and FORUM 80, automatic testing of the RS-232-C board; and even spooling of prepared messages on tape directly into FORUM 80 using a basic program supplied as a line listing. 4K Level II cassette, $24.95

ST80*

Reprogram your RS-232-C board from the keyboard, and run at different baud rates. Note: does not have auto testing of the RS-232-C or tape spooling. 4K Level II cassette $49.95

ST80*D

Connection time clock, option of user-created translation tables for keyboard, gathering and pre-formatting data to be sent directly from disk to host computer, spooling of received files to disk or printer, editing of received files, and auto logon. Use it with VTOS 3.1, and get device-driven I/O, job logging,and chaining. 32K disk program, $79.95

ST80*III

ST-80 D with extra utility programs. 32K disk program, $150.00

**ST80 is a trademark of Lance Micklus Inc.*

Unlock the power of the Z-80

STAD

SYMBOLIC TRACE AND DEBUG

A powerful monitor for the TRS-80 with special Debugging. Single Step through the machine language programs or set up to three breakpoints, and look at this display format!

```
AF   BC   DE   HL   IX   IY   AF'  BC'  DE'  HL'  SP   PC
0044 0000 C000 B77C 6433 FFFF 0102 0000 4000 3FC0 41FC 4400
4400    LD     A,93
```

A FIRST(0) LAST(FFFF)	ASCII dump
A FIRST 0	formatted ASCII dump
B	start of branch table
B VALA	display in decimal
B VALA VALB(0)	hex arithmetic
C	check system tape
D FIRST(0) LAST(FFFF)	dump hex
E FIRST(0)	edit memory
F FIRST LAST VALUE	find byte
G BRKPTS (3 max.)	set breakpoints, continue
H FIRST LAST VALUE	find word
I PORT	read port
K	keyboard echo
L	load system tape
L SECTOR MEMORY COUNT(1)	load from disk
M FIRST LAST BLOCK	move memory
N	display symbol table
N 0	symbol table to tape
N VALUE	define value for symbol table
N FIRST 0	define start symbol table
O PORT VALUE	write to port
P	initialize memory blocks
P ENTRY	write memory blocks and start
P FIRST LAST	define a memory block
Q FIRST LAST	calculate checksum
R	display / modify registers
S FIRST LAST OPTION(0)	disassembler
T COUNT OPTION(6)	trace instructions
U FIRST COUNT OPTION(0)	unformatted tape I/O
V FIRST LAST BLOCK	verify memory
W SECTOR MEMORY COUNT(1)	write to disk
X FIRST LAST BLOCK	exchange memory
Z FIRST LAST VALUE(0)	zero memory

TO ORDER TOLL-FREE: 1-800-258-1790

(In NH call 673-5144)

CONVERTING GRAPHICS
continued from page 23

It is important to distinguish between the color number and the color register on the Atari computers. The color number selects the color in the SETCOLOR command and puts it in a COLOR register. For example, the command SETCOLOR 2, 12, 4 sets one of the color registers to a medium green. (It sets register 3 in modes 3, 5, and 7). This can be confusing as it does not set register number two. Here is a chart of the registers affected by 'C' in SETCOLOR C, x, x:

	Register affected Graphics mode		
'C'	3,5 & 7	4,6	8
0	1	1	none
1	2	none	1
2	3	none	0
3	none	none	none
4	0	0	none

Text modes 0,1, and 2 work differently

The COLOR command selects a color register, not a color. Therefore, if we wanted to use our green in register 3, you would specify COLOR 3 before plotting your points. After that, you could change the color of the points already plotted by changing register 3 to another color.

It is not necessary to use a SETCOLOR command, as there is a default color for each of the five registers, as follows:

Register 0	Orange
1	Green
2	Dark Blue
3	Pink
4	Black

The GRAPHICS n command resets the registers to the default values.

The primary graphics commands for the Atari are as follows:

PLOT x, y (PL.) Displays one block at horizontal position x and vertical position y.

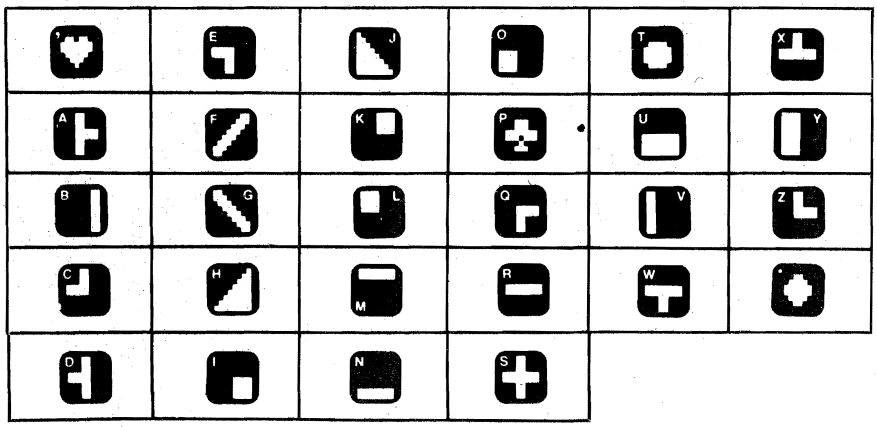
DRAWTO x, y (DR.) Draws a line from the previously plotted position to the position specified.

POSITION x, y (POS.) Places the cursor at the specified position.

LOCATE a, b, X (LOC.) Places the cursor at a, b and stores the current color register number of that point in variable X.

PUT # 6, c Places an ASCII character at the current location of the graphics cursor. The 6 is the device number of the graphics window.

PRINT # 6 It is also possible to use graphics characters and print them directly on the screen. The following chart shows most of the graphics characters. To get these characters directly from the keyboard, hold down the CONTROL key and press the letter in the corner of the character illustration on the chart.



Here is our little box for the Atari:

```
10 GR.3: PLOT 15,8
20 DR. 25,8 : DR. 25,12 : DR.
15,12 : DR. 15,8
```

You could also use the control graphics characters, with CONTROL Q, E, Z, and C for the four corners, CONTROL R for the horizontal lines, and control V and B for the vertical lines. These would have to be put in a string variable or enclosed in quotes in a PRINT statement.

Converting low resolution plotted graphics from one computer to another can be quite simple. Just determine the scale from the screen display descriptions in this article, and convert the horizontal and vertical coordinates over. Most of these calculations can be done mentally. For example, if you start with the Apple II in low resolution graphics with a 40 by 40 display, you can go to the TRS-80's 128 by 48 display by multiplying the horizontal dimension by three and leaving the vertical dimension untouched. You might have to double or triple the width of the vertical lines to get it to look right. The same conversion can go to Atari Graphics mode 4 by doubling the horizontal coordinate or to mode 3 by halving the vertical coordinate.

SET on the TRS-80, PLOT and HPLOT on the Apple, and PLOT on the Atari differ only in the scale of the drawing. The HLIN statements on the Apple are very similar to DRAWTO on the Atari, but you need a FOR NEXT loop on the TRS-80.

High resolution graphics can be converted directly between the Apple and the Atari by adjusting for scale, but it takes creative imagination to convert these programs to the TRS-80. You may have to plot out the graphics on a piece of graph paper, then use your ingenuity on your own computer.

"our price vs. list price" compare!

TRS-80

	List Price	Our Price
Level-II, 4K	$649.00	$599.00
Level-II 16K no keypad	$768.00	$669.00
Level-II 16K with keypad	$849.00	$749.00
COMM-80 Interface		$179.95
CHATTER BOX Interface		$259.95
Disk-80 Interface		$299.95
Expansion Interface, no RAM	$299.00	$279.00
Expansion Interface 16K RAM	$418.00	$369.00
Expansion Interface 32K RAM	$537.00	$459.00
RS-232-C Board	$99.00	$89.00
TRS-232 Printer Interface	$59.95	$49.95
16K Memory Kit, TRS-Keyboard	$119.00	$89.00
16K Memory Kit, TRS-Exp. Int	$119.00	$85.00
Percom, TFD-100, 40-track	$399.95	$389.00
Percom, Dual TFD-100 Drives	$795.00	$775.00
Percom, TFD-200, 77-track	$675.00	$599.00
Percom Data Separator		$29.95
Percom Extender Card	$15.95	$15.00
2-Drive Cable	$29.95	$29.00
4-Drive Cable	$39.95	$39.00
Data Dubber	$49.95	$44.95
Percom Electric Crayon w/cbl		$279.95
Busy Box, TRS-80	$109.95	$99.95
BSR X-10 Starter Kit	$124.95	$99.95
Busy Box 4-yr Control Program		$19.95

ATARI

ATARI 400 Comp. System, 8K	$629.95	$499.00
ATARI 800 Comp. System 16K	$1079.95	$819.00
ATARI 410 Program Recorder	$89.95	$59.00
ATARI 810 Disk Drive	$699.95	$569.00
ATARI 815 Dual Density Disks	$1495.95	$1219.00
ATARI 820 Printer (40 col.)	$599.95	$469.00
ATARI 822 Thermal Printer	$449.95	$369.00
ATARI 825 Printer	$999.95	$819.00
ATARI 830 Acoustic Modem	$199.95	$169.00
ATARI 850 Interface Module	$219.95	$179.00
ATARI Light Pen	$74.95	$59.00
8K RAM Memory Module	$124.95	$99.00
16K RAM Memory Module	$199.95	$159.00
ATARI Joystick Controllers	$19.95	$18.95
ATARI Paddle Controllers	$19.95	$18.95

CARRYING CASES

TRS-80 Computer Case	$109.00	$99.00
TRS-80 Monitor Case	$84.00	$79.00
APPLE System Case w/1 drive	$109.00	$99.00
APPLE System Case w/2 drive	$119.00	$109.00
APPLE System Case w/mon.	$129.00	$119.00

APPLE

	List Price	Our Price
Mtn. Hard. Apple Clock	$280.00	$262.50
Mtn. Hard. ROMPLUS+ w/Fil	$200.00	$187.50
16K Memory Kit	$119.00	$85.00
Busy Box, Apple	$114.95	$99.95
BSR X-10 Starter Kit	$124.95	$99.95

PET/CBM

PET 2001 large keyboard, 16K	$995.00	$895.00
PET 2001 large keyboard, 32K	$1295.00	$1195.00
CBM 8032 Business Comp.	$1795.00	$1619.00
CBM Cassette Recorder	$95.00	$84.95
CBM 2040 Dual Disk Drives	$1295.00	$1195.00
CBM 8050 Dual Disk Drives	$1695.00	$1495.00
CBM Modem	$395.00	$349.00
CBM Voice Synthesizer	$395.00	$349.00
CBM to IEEE Cable	$40.00	$37.50
IEEE to IEEE Cable	$40.00	$37.50

PRINTERS

Centronics 730	$795.00	$699.00
Centronics 737	$995.00	$875.00
Centronics 779	$1559.00	$1095.00
Centronics 779 w/lower case & motor control	$1779.00	$1295.00
NEC 5510 SpinWriter	$3195.00	$2595.00
NEC 5520 SpinWriter	$3395.00	$2895.00
NEC 5530 SpinWriter	$3195.00	$2495.00
NEC Tractor-Feed Option	$225.00	$195.00
LRC 7000+ (64-col.)	$405.00	$349.00
LRC 7000+ (40-col.)	$389.00	$339.00
Okidata Microline-80	$800.00	$709.00
Tractor-Feed Option	$140.00	$129.00
RS-232-C (2K) Option	$299.00	$279.00
LRC to TRS-80 or APPLE		$20.00
LRC to PET, IEEE		$59.00
LRC to RS232C, male/female		$65.00
730 or 737 to TRS-80		$29.00
NEC or 779 to TRS-80		$35.00
RS-232-C/RS-232-C male/male		$24.95

DUST COVERS

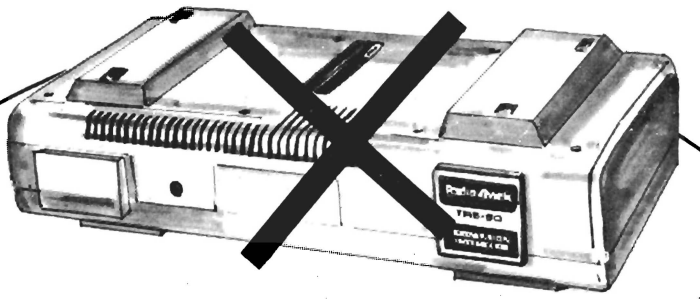
TRS-80 (3pc set)	$7.95
APPLE	$7.95
ATARI 800	$7.95
ATARI 400	$7.95
PET/CBM 2001	$11.95

*TRS-80, ATARI, APPLE, and PET/CBM are trademarks of Tandy Corp. Warner Communications, Apple Computer Co., and Commodore respectively.

HARDSIDE 6 SOUTH ST., MILFORD, NH 03055

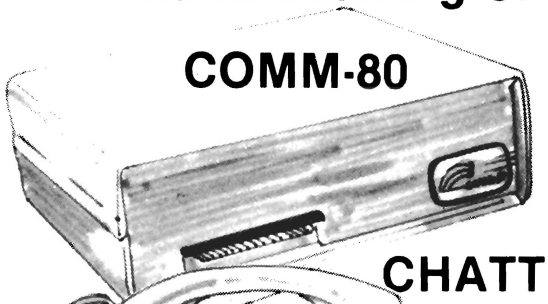
To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

The Software Exchange & HardSide (Div. of Robitaille & Sons Enterprises Inc.) SoftSide Publications



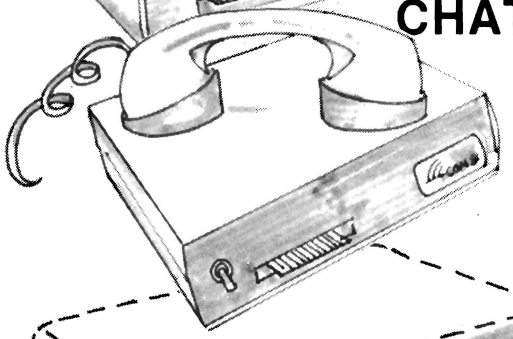
TRS-80* interfacing alternatives...

Featuring the Quality Engineering and Manufacturing of Steve Ciarcia and Micromint



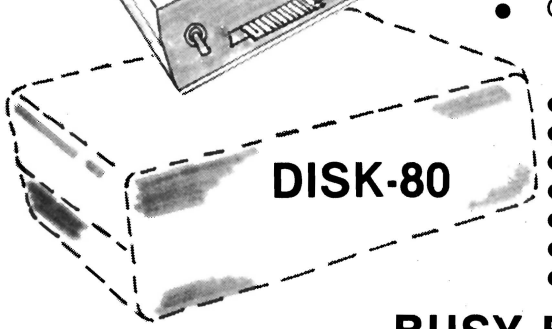
COMM-80

- FULL 8-BIT PARALLEL PORT
- RS-232-C PORT (up to 19,200 baud)
- UP TO 16 UNITS CAN BE CHAINED TOGETHER
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- CONNECTS TO KEYBOARD OR E.I.
- ONLY $179.95



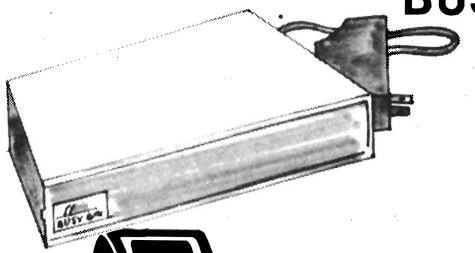
CHATTER BOX

- FULL 8-BIT PARALLEL PORT
- RS-232-C PORT (up to 19,200 baud)
- ACOUSTIC MODEM
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- CONNECTS TO KEYBOARD OR E.I.
- INCLUDES TERMINAL SOFTWARE
- ONLY $259.95



DISK-80

- DISK CONTROLLER (up to 4 drives)
- DATA SEPARATOR
- INCLUDES 16K OF RAM
- PROVISION FOR AN ADDITIONAL 16K RAM
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- Available September 1980
- ONLY $299.95



BUSY BOX

- 8 1/4" x 6 2/5" PLASTIC CASE
- CABLE AND CONNECTOR FOR TRS-80 KEYBOARD E.I.
- OPERATING MANUAL
- INCLUDES BASIC LISTING FOR CONTROL ROUTINE
- LEVEL II MINUMUM SYSTEM
- ONLY $109.95

*TRS-80 is a trademark of Tandy Corp.

HARDSIDE™ 6 SOUTH ST., MILFORD, NH 03055

To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

VISA

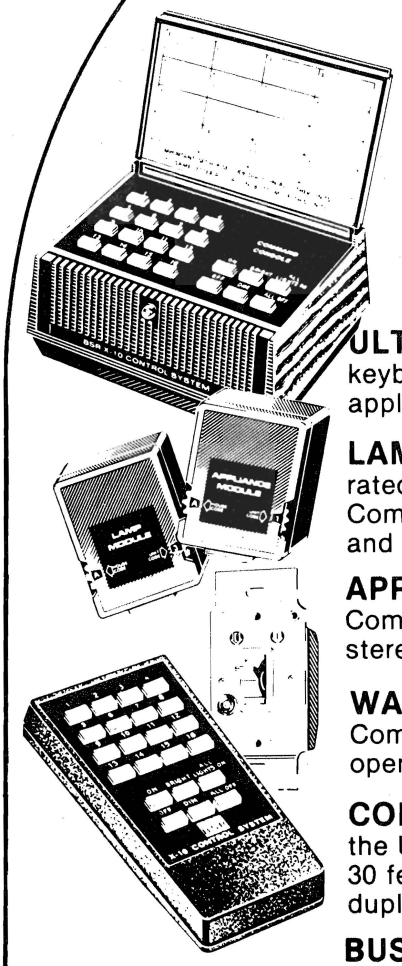
The Software Exchange & HardSide (Div of Robitaille & Sons, Enterprises, Inc.), SoftSide Publications

The Controller

ELECTRONIC SECURITY AND CONVENIENCE AT YOUR FINGERTIPS

BSR

SYSTEM X-10



ULTRA SONIC COMMAND CONSOLE . . . Its keyboard permits remote control operation of lights and appliances in up to 16 separate locations. (List $49.95) $42.99

LAMP MODULE . . . Will control any incandescent lamp rated up to 300 watts from control signals received from Command Console. Functions include on and off, brighten and dim. (List $16.99) $14.99

APPLIANCE MODULE . . . Receives signals from the Command Console to turn an appliance on and off (e.g.: TV, stereo, fan, etc.). (List $16.99) $14.99

WALL SWITCH MODULE . . . Receives signals from the Command Console to control lights or lamps normally operated by a wall switch. (List $16.99) $14.99

CORDLESS CONTROLLER . . . Transmits signals to the Ultra Sonic Command Console from any distance up to 30 feet. All functions of the Command Console are duplicated on the Cordless Controller. (List $24.99) $19.99

BUSY BOX . . . Connects to your micro-computer and transmits signals to the Ultra Sonic Command Console from any distance up to 30 feet. Your computer can now be programmed to control all functions of the BSR X-10 Home Control System, including dimming of lights. This unit connects directly to your TRS-80 (List $109.95) or APPLE (List $114.95) $99.95

S-100 Busy Box (List $119.95) $114.95

SPECIAL CONTROLLER STARTER KIT . . . Includes BSR X-10 Ultra Sonic Controller, 2 Lamp Modules, 1 Appliance Module, 1 Cordless Controller and Busy Box (specify for TRS-80 or APPLE). (List $240.90) $199.95

TRS-80 4 YEAR HOME CONTROL PROGRAM . . . $19.95

(Above prices do not include postage and handling)

HARDSIDE 6 SOUTH ST., MILFORD, NH 03055

To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

The Software Exchange & HardSide (Div. of Robitaille & Sons Enterprises, Inc.) SoftSide Publications

VISA

The Software Exchange Order Form

DESCRIPTION	MEMORY	TYPE	PRICE

SPECIFY: TRS-80, APPLE, ATARI or PET. Do not use for hardware.

Check
Money Order

CHARGE ACCOUNT NUMBER

ADD handling charges	**$1.00**
Additional charges	
TOTAL	

Signature_____ Exp. Date_____ Inter. No._____

Name_____

Address_____

City_____ State_____ Zip_____

MAIL TO: The Software Exchange
6 South St.
Milford, NH 03055
(In NH call 673-5144)

or, CALL TOLL FREE
for Orders Only
1-800-258-1790

Special prices in effect 60 days from mailing

Level II software available on disk for a $5.00 **(per order)** medium charge. This extra fee is for any number of programs transferred to disk from tape when you order. If the order exceeds the capacity of a single disk, we absorb the extra cost.

Be sure to include handling charge and any additional charges when figuring your total. All orders are shipped within 48 hours.

All prices are subject to change without notice. We are not responsible for typographical errors, including incorrect prices.

ALL SOFTWARE GUARANTEED TO LOAD AND RUN. If you experience difficulties, simply return the tape or disk properly protected for free replacement. Send to the attention of Bette Keenan, Customer Service Representative: please enclose a brief note and your name and mailing address with the software.

ALL SOFTWARE SOLD ON AN AS-IS BASIS WITHOUT WARRANTY. TSE assumes no liability for loss or damage caused or alleged to be caused directly or indirectly by equipment or products sold or exchanged by them or their distributors, including but not limited to any interruption in service, loss of business or anticipatory profits or consequential damages resulting from use or operation of such equipment or software.

HARDSIDE ORDER FORM

NAME

STREET

CITY STATE ZIP TEL. NO.

VISA/MASTER CHARGE EXP. DATE

QUAN.	PART NO.	PRODUCT DESCRIPTION	AMOUNT

MAIL TO:

HARDSIDE
6 South Street
Milford, NH 03055

UPS Zone Weight

Total Retail	
Handling Charge	
Insurance	
Shipping Charge	
TOTAL	
Amount Enclosed	
COD Balance	

IMPORTANT

Use this form for hardware orders only. To avoid delay, be sure to fill in all the necessary information. Please PRINT CLEARLY. Hardside's terms and conditions are stated in this catalog. We are not responsible for typographical errors, including incorrect prices. Prices are subject to change without notice. We reserve the right to limit quantities.

UPS CHART

ADDITIONAL CHARGES:

For each COD received for collection - $1.25
For each Address Correction - $1.25
For each Acknowledgment of Delivery (AOD) - 22 cents.
For each package with a declared value over over $100.00 - 25 cents for each additional $100 or fraction thereof.

WEIGHT AND SIZE LIMITS:

Maximum Weight per package 50 POUNDS
Maximum Weight of all packages from one shipper to one consignee in one day - 100 pounds.
Maximum Size per package - 108 INCHES IN LENGTH AND GIRTH COMBINED
Minimum charge for a package measuring over 84 inches in length and girth combined will be equal to charge for a package weighing 25 pounds.

United Parcel Service — Effective May 1, 1980

COMMON CARRIER RATE CHART

Including 1.3% Fuel Surcharge

ANY FRACTION OF A POUND OVER THE WEIGHT SHOWN TAKES THE NEXT HIGHER RATE

WEIGHT NOT TO EXCEED	RATES TO ZONE 2	ZONE 3	ZONE 4	ZONE 5	ZONE 6	ZONE 7	ZONE 8	WEIGHT NOT TO EXCEED	RATES TO ZONE 2	ZONE 3	ZONE 4	ZONE 5	ZONE 6	ZONE 7	ZONE 8
1 lb.	$1.08	$1.10	$1.13	$1.18	$1.23	$1.28	$1.34	26 lb.	$3.23	$3.84	$4.66	$5.58	$6.92	$8.32	$9.95
2 "	1.16	1.22	1.28	1.35	1.45	1.56	1.68	27 "	3.32	3.95	4.80	5.75	7.15	8.60	10.29
3 "	1.26	1.33	1.42	1.53	1.68	1.84	2.03	28 "	3.40	4.06	4.94	5.94	7.37	8.88	10.64
4 "	1.34	1.44	1.56	1.70	1.90	2.13	2.37	29 "	3.49	4.17	5.09	6.11	7.61	9.17	10.98
5 "	1.43	1.54	1.70	1.87	2.14	2.40	2.71	30 "	3.58	4.27	5.22	6.28	7.83	9.44	11.33
6 "	1.51	1.65	1.84	2.06	2.36	2.68	3.06	31 "	3.67	4.39	5.36	6.46	8.06	9.72	11.67
7 "	1.60	1.76	1.99	2.23	2.59	2.97	3.40	32 "	3.75	4.50	5.50	6.64	8.29	10.01	12.01
8 "	1.68	1.87	2.13	2.41	2.82	3.25	3.75	33 "	3.84	4.61	5.64	6.82	8.52	10.29	12.36
9 "	1.77	1.99	2.27	2.58	3.05	3.54	4.09	34 "	3.92	4.72	5.78	6.99	8.74	10.58	12.70
10 "	1.85	2.09	2.40	2.76	3.27	3.81	4.44	35 "	4.01	4.82	5.93	7.16	8.98	10.85	13.05
11 "	1.94	2.20	2.54	2.94	3.50	4.09	4.78	36 "	4.09	4.93	6.07	7.34	9.20	11.13	13.39
12 "	2.03	2.31	2.68	3.11	3.73	4.38	5.13	37 "	4.18	5.04	6.21	7.52	9.43	11.42	13.74
13 "	2.12	2.42	2.83	3.29	3.96	4.66	5.47	38 "	4.26	5.16	6.35	7.70	9.65	11.70	14.08
14 "	2.20	2.53	2.97	3.46	4.18	4.94	5.81	39 "	4.36	5.27	6.49	7.87	9.89	11.98	14.43
15 "	2.29	2.63	3.11	3.64	4.42	5.22	6.16	40 "	4.44	5.37	6.63	8.04	10.11	12.26	14.77
16 "	2.37	2.75	3.25	3.82	4.64	5.50	6.50	41 "	4.53	5.48	6.77	8.23	10.34	12.54	15.11
17 "	2.46	2.86	3.39	3.99	4.87	5.78	6.85	42 "	4.61	5.59	6.91	8.40	10.57	12.82	15.46
18 "	2.54	2.97	3.54	4.17	5.10	6.07	7.19	43 "	4.70	5.70	7.05	8.58	10.80	13.11	15.80
19 "	2.63	3.08	3.68	4.35	5.33	6.35	7.54	44 "	4.78	5.81	7.19	8.75	11.02	13.39	16.15
20 "	2.71	3.18	3.81	4.52	5.55	6.63	7.88	45 "	4.87	5.92	7.33	8.92	11.25	13.67	16.49
21 "	2.81	3.29	3.95	4.70	5.78	6.91	8.23	46 "	4.95	6.03	7.48	9.11	11.48	13.95	16.84
22 "	2.89	3.40	4.09	4.87	6.01	7.19	8.57	47 "	5.04	6.14	7.62	9.28	11.71	14.23	17.18
23 "	2.98	3.52	4.23	5.05	6.24	7.48	8.91	48 "	5.13	6.25	7.76	9.46	11.93	14.52	17.52
24 "	3.06	3.63	4.38	5.23	6.46	7.76	9.26	49 "	5.22	6.36	7.90	9.63	12.17	14.80	17.87
25 "	3.15	3.73	4.52	5.40	6.70	8.03	9.60	50 "	5.30	6.46	8.03	9.81	12.39	15.07	18.21

COMING IN THE SEPTEMBER ISSUE

S-80

Dr. Livingston, in search of . . .
Adventure in the dark continent!

Lone Star Corral
Can you outsmart a horse?

APPLE

Stereo Music Generator
Why settle for monaural sound?

Foosball
Paddle yourself silly! ! !

ATARI

Ricochet
Strain your powers of deduction!

Sleuth
Mind boggling

Super Barricade
Crossover adds to the challenge!

PLUS . . .

columns, articles, and other features!

Don't miss a single issue of the great new SoftSide!

The BONE FONE is a miniaturized AM/FM stereo system that drapes over your shoulders like a scarf. The BONE FONE gives new meaning to the word portable. You don't carry a BONE FONE around with you. You wear it. Ski with it, cycle, jog or just take a leisurely stroll through the woods with it. It's the hands-free stereo that brings beautiful music to every indoor or outdoor activity without disturbing those around you and without covering your ears.

The BONE FONE stereo is covered with a sleeve made of Lycra Spandex - the same material used to make expensive swim suits, so it's easily washable. You simply remove the sleeve, dip it in soapy water, rinse and let dry. The entire system is also protected against damage from moisture and sweat making it ideal for jogging or bicycling.

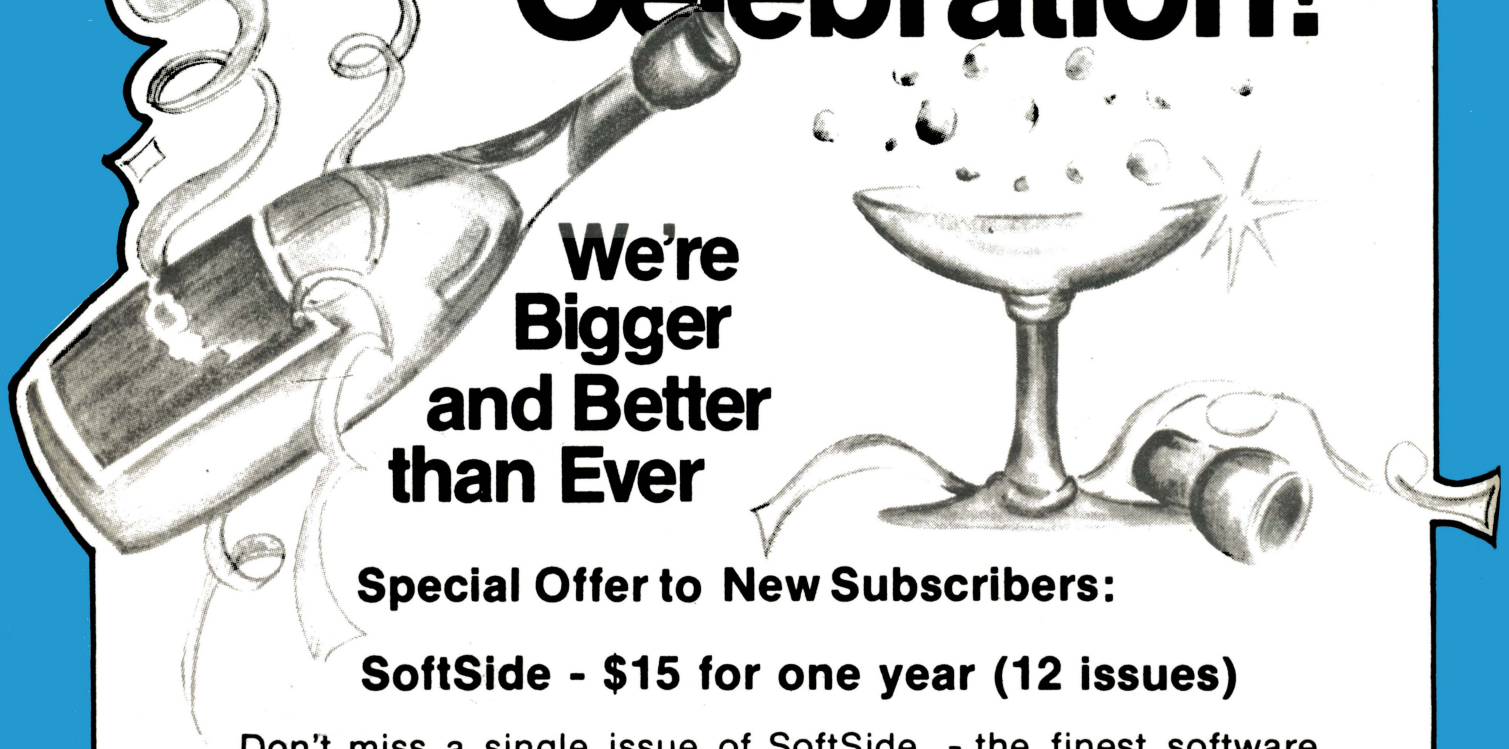
The BONE FONE has been selling nation-wide for $70. HARDSIDE's special price is ONLY $56.95 plus $2.50 handling (we pay the UPS charges).

9902.....$56.95

Bone Fone

TO ORDER
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

Join Our SoftSide Celebration!



We're Bigger and Better than Ever

Special Offer to New Subscribers:

SoftSide - $15 for one year (12 issues)

Don't miss a single issue of SoftSide - the finest software source for the Radio Shack TRS-80*, APPLE II*, and ATARI*!

Save hours of typing — buy SoftSide's programs for your computer already on cassette or diskette, with magazine. Try our low introductory price for one year (12 issues).

To order, call TOLL FREE **1-800-258-1790** (in NH call 673-5144) or mail the following information to: SoftSide Publications, Subscriptions, 6 South Street, Milford, NH 03055.

☐ SoftSide: $15., one yr.,(12 issues)

☐ SoftSide: Apple on cassette $69.

☐ SoftSide: Apple on diskette $129.

☐ SoftSide: S-80 on cassette $69.

☐ SoftSide: S-80 on diskette $129.

SoftSide: Atari on cassette and diskette will be available soon!

I own:
- ☐ Apple
- ☐ TRS-80
- ☐ Atari
- ☐ Pet
- ☐ Other _____

Exp. Date _____ Interbank #(M/C only) _____

Signature _____

Name _____

Address _____

_____ State _____ Zip _____

Subscribe to SoftSide today!

Introductory offer valid until November 1, 1980. Not applicable to overseas subscribers. Write for subscriptions rates.

*TRS-80, Apple and Atari are trademarks of Tandy, Apple Computer Co., and Warner Communications, respectively.