Special Features: Telegames And Nonviolent Games

COMPUTE!

$2.50
October
1983
Issue 41
Vol. 5, No. 10
£1.25 UK $3.25 Canada
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

Two Gripping Games With Excellent Graphics For The Commodore 64, VIC-20, And Atari: *Moving Maze* And *Dragonmaster*

***Spiralizer*: A High-Res Drawing Utility For Atari, Apple, VIC-20, Radio Shack Color Computer And Commodore 64**

Merging Programs From Commodore Disks

Atari Multicolor Character Editor

A Sound Tutorial For The Commodore 64

The story of STORY MACHINE.™ Ages 5 to 9.

STORY MACHINE is like a storybook come to life. Using the keyboard, your children write their own fun little stories. The computer then takes what they've written and animates their story on the screen, complete with full color graphics and sound.

STORY MACHINE helps your children learn to write correctly, become familiar with the keyboard, and lets them have fun exercising their creativity at the same time.

KINDERCOMP.™ Numbers, shapes, letters, words and drawings make fun. Ages 3 to 8.

KINDERCOMP is a game that allows very young children to start learning on the computer. It's a collection of learning exercises that ask your children to match shapes and letters, write their names, draw pictures, or fill in missing numbers. And KINDERCOMP will delight kids with colorful rewards, as the screen comes to life when correct answers are given.

As a parent, you can enjoy the fact that your children are having fun while improving their reading readiness and counting skills.

FACEMAKER™ makes faces fun. Ages 4 to 12.

FACEMAKER lets children create their own funny faces on the screen. Once a face is completed, your children will giggle with delight as they make it do all kinds of neat things: wink, smile, wiggle its ears, or whatever their imagination desires.

Plus, FACEMAKER helps children become comfortable with computer fundamentals such as: menus, cursors, the return key, the space bar, simple programs, and graphics. FACEMAKER won't make parents frown because their children will have fun making friends with the computer.

SPINNAKER™
We make learning fun.

Disks for: Apple, Atari, IBM, Commodore 64.
Cartridges for: Atari, Commodore 64

October 1983 Vol. 5, No. 10

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 50 before typing in programs

GUIDE TO ARTICLES AND PROGRAMS

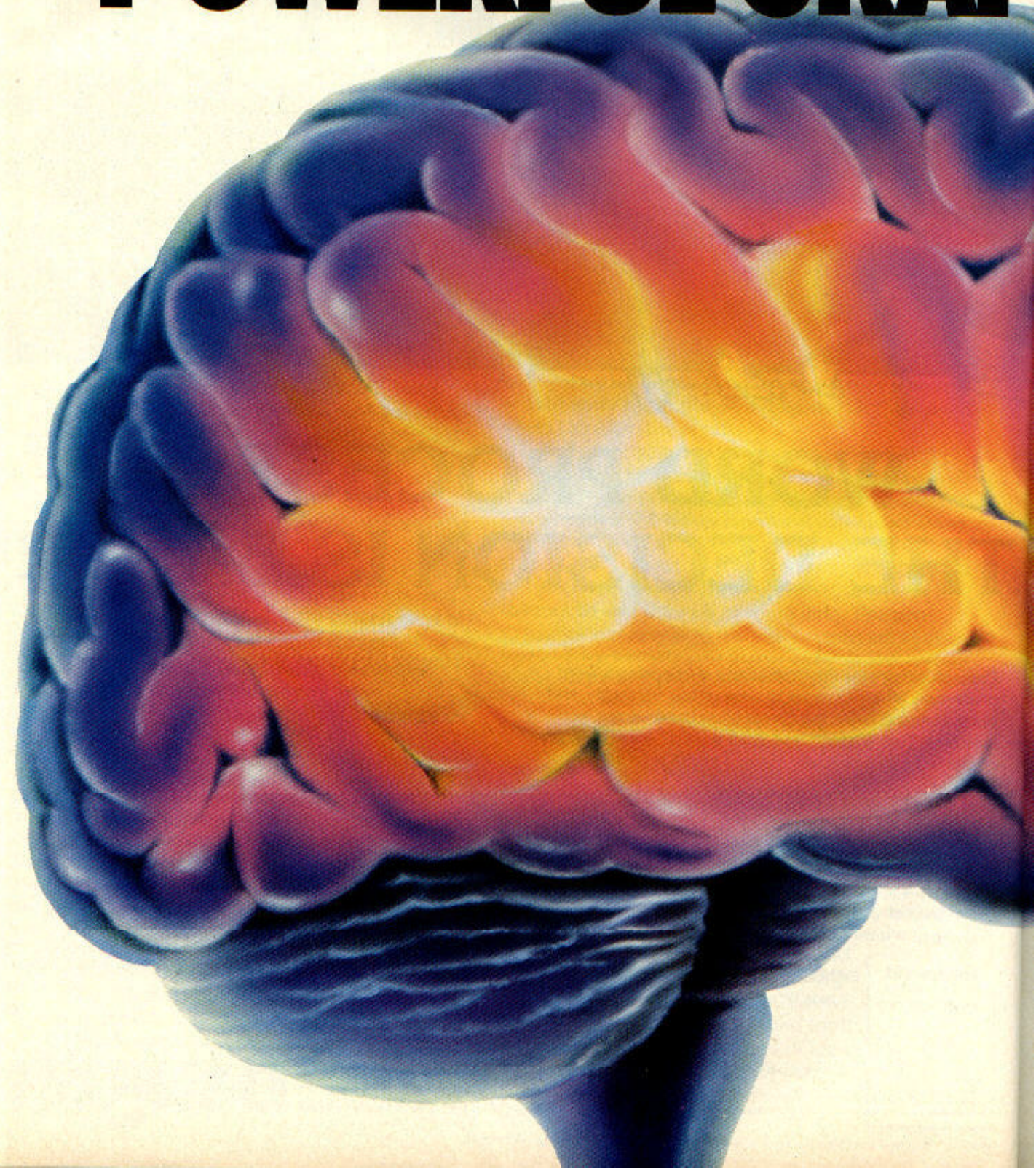
AP Apple **AT** Atari, **P** PET/CBM, **V** VIC-20, **C** Radio Shack Color Computer, **64** Commodore 64, **TS** Timex/Sinclair, **TI** Texas Instruments, *All or several of the above.

COMPUTE! Publications, Inc.

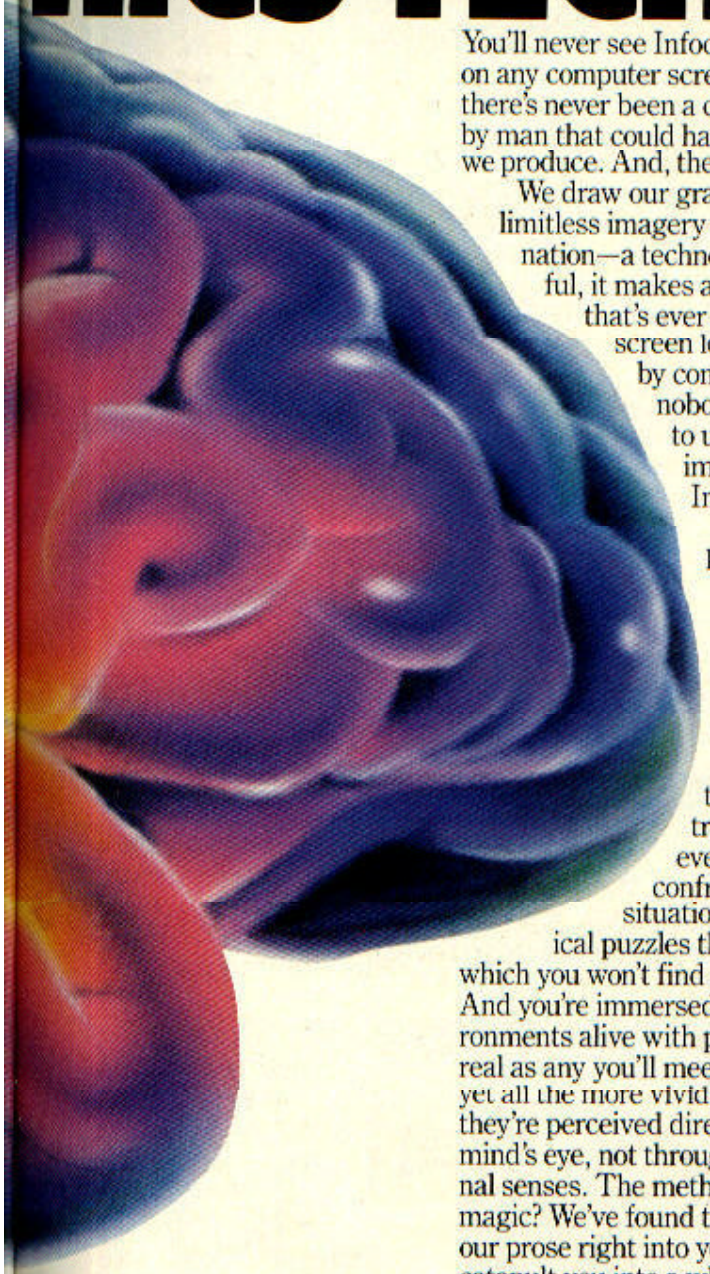
TOLL FREE Subscription Order Line 800-334-0868 (In NC 919-275-9809)

COMPUTE! The Journal for Progressive Computing [illegible]

WE UNLEASH TH
POWERFUL GRAP



E WORLD'S MOST
HICS TECHNOLOGY.



You'll never see Infocom's graphics on any computer screen. Because there's never been a computer built by man that could handle the images we produce. And, there never will be.

We draw our graphics from the limitless imagery of your imagination—a technology so powerful, it makes any picture that's ever come out of a screen look like graffiti by comparison. And nobody knows how to unleash your imagination like Infocom.

Through our prose, your imagination makes you part of our stories, in control of what you do and where you go—yet unable to predict or control the course of events. You're confronted with situations and logical puzzles the like of which you won't find elsewhere. And you're immersed in rich environments alive with personalities as real as any you'll meet in the flesh—yet all the more vivid because they're perceived directly by your mind's eye, not through your external senses. The method to this magic? We've found the way to plug our prose right into your psyche, and catapult you into a whole new dimension.

Take some tough critics' words about our words. SOFTALK, for example, called ZORK® III's prose "far more graphic than any depiction yet achieved by an adventure with graphics." And the NEW YORK TIMES saw fit to print that our DEADLINE™ is "an amazing feat of programming." Even a journal as video-oriented as ELECTRONIC GAMES found Infocom prose to be such an eye-opener, they named one of our games their Best Adventure of 1983.

Better still, bring an Infocom game home with you. Discover firsthand why thousands upon thousands of discriminating game players keep turning everything we write into instantaneous bestsellers.

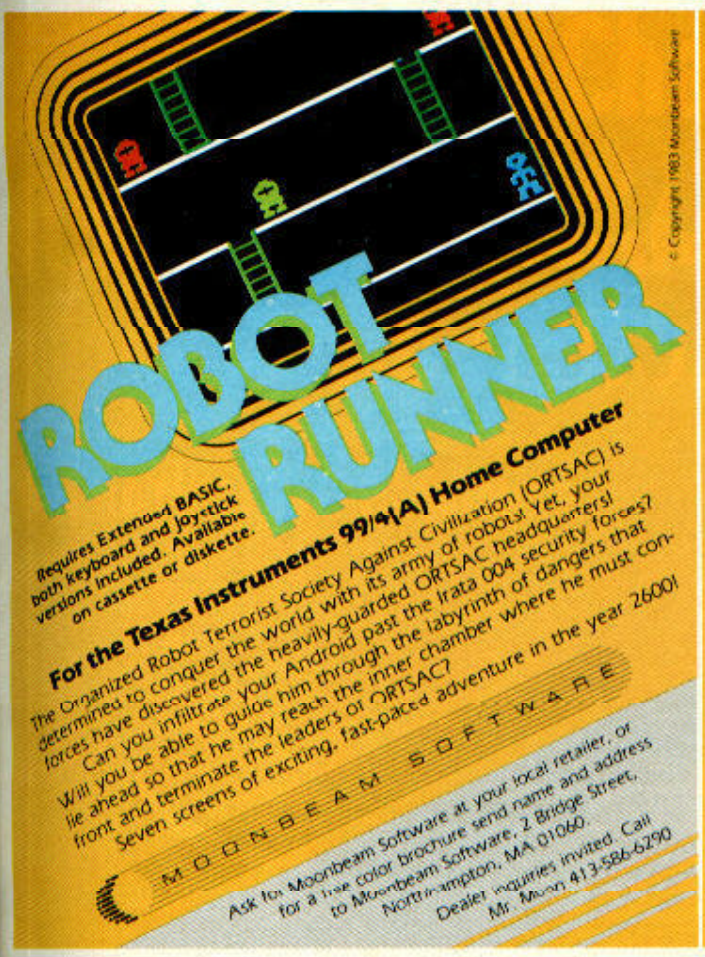
Step up to Infocom. All words. No graffiti. The secret reaches of your mind are beckoning. A whole new dimension is in there waiting for you.

INFOCOM™

The next dimension.

Infocom, Inc., 55 Wheeler St., Cambridge, MA 02138

For your: Apple II, Atari, Commodore 64, CPM 8", DEC Rainbow, DEC RT-11, IBM, NEC APC, NEC PC-8000, Osborne, TI Professional, TRS-80 Model I, TRS-80 Model III.



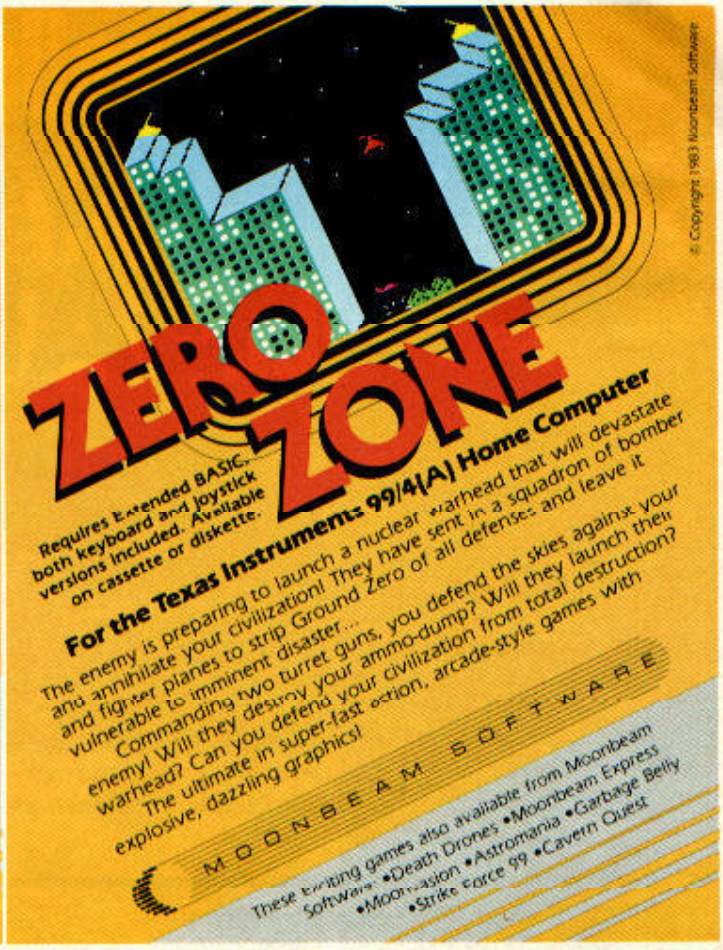
Requires Extended BASIC, both keyboard and joystick versions included. Available on cassette or diskette.

For the Texas Instruments 99/4[A] Home Computer

The Organized Robot Terrorist Society Against Civilization (ORTSAC) is determined to conquer the world with its army of robots! Yet, your forces have discovered the heavily-guarded ORTSAC headquarters! Can you infiltrate your Android past the Irata 004 security forces? Will you be able to guide him through the labyrinth of dangers that lie ahead so that he may reach the inner chamber where he must confront and terminate the leaders of ORTSAC?
Seven screens of exciting, fast-paced adventure in the year 2600!

MOONBEAM SOFTWARE

Ask for Moonbeam Software at your local retailer, or for a free color brochure send name and address to Moonbeam Software, 2 Bridge Street, Northampton, MA 01060.
Dealer inquiries invited. Call Mr. Moon 413-586-6290



Requires Extended BASIC, both keyboard and joystick versions included. Available on cassette or diskette.

For the Texas Instruments 99/4[A] Home Computer

The enemy is preparing to launch a nuclear warhead that will devastate and annihilate your civilization! They have sent in a squadron of bomber and fighter planes to strip Ground Zero of all defenses and leave it vulnerable to imminent disaster...
Commanding two turret guns, you defend the skies against your enemy! Will they destroy your ammo-dump? Will they launch their warhead? Can you defend your civilization from total destruction?
The ultimate in super-fast action, arcade-style games with explosive, dazzling graphics!

MOONBEAM SOFTWARE

These exciting games also available from Moonbeam Software: •Death Drones •Moonbeam Express •Moon Invasion •Astromania •Garbage Belly •Strike Force 99 •Cavern Quest

---

"SMART TERMINAL"

TELECOMPUTING POWER FOR VIC - 20 COMMODORE 64

"Smart-Term" does more than convert your VIC or 64 to a null terminal, it gives you features other programs don't offer, such as:

- A PET to ascii conversion test mode
- Transmits all ascii control characters
- Transmits 122 ascii codes
- Receives 92 ascii codes
- Has repeat key feature
- Allows you to enter and save four permanent messages (up to 80 characters each) for one key, transmission of code, password, names, messages, etc.
- Easy to read, smooth scrolling characters
- Fully menu driven for reliability and ease of use
- Transmits and receives in seperate character colors (selectable)
- Many other features

One of the best telecommunications programs available, "Smart-Term" converts your VIC or 64 into a *SMART TERMINAL*. Requires minimum 3K expansion.

TO ORDER: Specify VIC-20 or 64;
TAPE: $16.95 or
DISK: $18.95
(U.S. funds) + $1.50 P&H

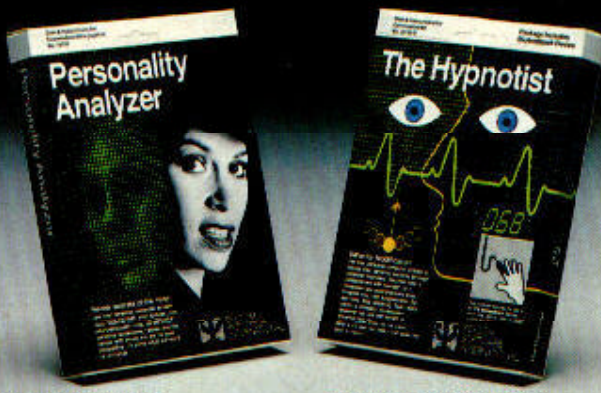
TO ORDER SEND CHECK OR MONEY ORDER. CHECK NEED THREE WEEKS TO CLEAR. NO C.O.D.'s TO U.S. CANADIAN ORDERS ADD 25% FOR CDN. FUNDS.

BYTE — RYTE
P.O. BOX 205, STATION CART.
MONTREAL, QUEBEC
H4K 2J5 CANADA

DEALER INQUIRIES INVITED

VIC-20 and Commodore 64 are registered trademarks of Commodore Business Machines Inc.

Open your mind



© 1983 PATRICK J. WILLIAMS

Reveal secrets of the mind.
Use your Commodore 64 system to analyze yourself, your spouse, your date, relatives and friends. Discover your personality type, career potential, behavior tendencies, values, and the people with whom you will be most compatible. This program requires the use of a "joystick". Price $32.95 Disk ($27.95 Cassette).

Behavior Modification.
Use your Commodore 64 system to change your behavior patterns through computer hypnosis. Discover how to communicate with *yourself*, on a conscious and subconscious level. Program your own post-hypnotic suggestions. The PSI Biofeedback Device is included with this program. Price $87.95 Disk ($79.95 Cassette).

Get this software at your local dealer or order direct from:

PSYCOM SOFTWARE INTERNATIONAL

2118 Forest Lake Drive
Cincinnati, Ohio 45244 USA
Telephone: 513 474-2188

Into the violent world of video games, Romox™ introduces

Two of the Friendliest, Fastest, Fun-packed Family Games Ever!

ANTEATER.™ Ant or Anteater—who will be king of the hill? Somewhere along the twisting, turning tunnels of this marvelous maze game the answer will be found. Until then, gamesters of every age will be caught up in the non-stop action and strategy of Anteater, the game that's crawling with arcade excitement. **Cartridges for the Atari® 400/800/1200XL, Commodore Vic 20™, Commodore 64™, Texas Instruments 99/4A.™**

TYPO.™ Chased by an evil spaceship, your only chance of escape is through "knowledge." Knowledge of typing and spelling. The faster you type, the better your chance of eluding the enemy. Spelling homework can be fun with this fast-paced game that not only develops word power—but increases your typing speed with every fun-filled hour as you master the typewriter-style keyboard of personal computers. **Cartridges for the Atari® 400/800/1200XL, Commodore Vic 20™, Commodore 64™, Texas Instruments 99/4A.™**

ANTEATER and TYPO are only two of the exciting recreational and educational games from Romox™. See our complete selection at your Romox dealer.

Romox Inc., 476 Vandell Way, Campbell, CA 95008.

Romox is a trademark of Romox Inc. Atari is a trademark of Atari, Inc. VIC 20 and Commodore 64 are trademarks of Commodore Business Machines, Inc. TI 99/4A is a trademark of Texas Instruments.

ROMOX™

COUPON FILE

Ken D. McCann and Dale McBane,
Technical Assistant

Not only will this program create easily accessible files for disk or tape storage, but also it will run on any Commodore machine, Atari, TI, Apple, and Color Computer. And with minor adjustments, you can file nearly anything.

This program allows you to file and search for coupons. Coupons may be located by brand name, product, or expiration date, and you may scan the contents of all the files. "Coupon File" could also help you file a great variety of things.

The use of DATA statements as file structures is one of the things that makes this program so versatile. Because the DATA statements are saved with the program, file retrieval is not a problem, even if you don't have disk capability.

The DATA statements were placed before the main program loop to simplify file insertions and deletions. Because the file number and line number for the DATA statements are the same, those of you unfamiliar with programming will find it easy to create files.

Using The Program On Your Computer

There is room for 499 files, assuming you have enough memory to store 499 files. Because of the search routine, each file must have the same format. This is especially true with the date search. 6/30/83, 6-30-83, and 30JUN83 are all different representations of the same date, but for the computer to locate that date, you must choose one format and be consistent.

Coupon File was written to run on any machine which supports BASIC, with one exception. Lines 501, 1000, 1550, 2000, 4000, 6000, and 7100 consist of the statement PRINT "{CLR}". This is COMPUTE!'s listing convention for clear screen on the Commodore 64 and VIC-20. You should substitute the statement to clear the screen on your particular machine (ESC SHIFT < for Atari, CALL CLEAR for TI, etc.).

To make more room for files, you can leave out the instructions. To do this, delete lines 550-555, 630, and 6000-7000, and change line 590 to:

```
590 IF (K$<>"L")*(K$<>"B")*(K$<>"P")*(K$<>"D")*
    (K$<>"C") THEN GOTO 570
```

This versatile program is very easy to use, and it's easy to adapt for other purposes. With a few changes, you can create a program to file nearly anything.

Special Note To Timex/Sinclair Users:

Because your machine's version of BASIC does not contain READ or DATA statements, this program will not run on your machine. You may be able to adapt it to your machine using strings for file storage.

Program Explanation

Lines	
1-499	DATA
500-650	main menu
1000-1700	list all routine
2000-3600	brand search routine
4000-5700	date search routine
6000-7000	instructions
7100-8610	product search routine
9000-9600	display routine
10000-10300	file input routine

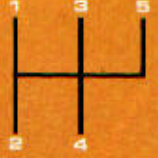
Coupon File

```
1 DATA 1,OXYDOL,DETERGENT,FREE,NONE
2 DATA 2,PLANTERS,MIXED NUTS,25C,31MAR84
3 DATA 3,FREE N' SOFT,FABRIC SOFTNER,20C,
  1JAN84
4 DATA 4,JELLO,PUDDING,15C,15MAY84
5 DATA 5,JENOS,PIZZA,1$ R,4JUL84
6 DATA 6,CHINET,PLATES,20C,21DEC83
7 DATA 7,PEPPIS,PIZZA,60C,15MAY84
8 DATA 8,CHINET,CUPS,20C,20JUL83
9 DATA 9,NABISCO,SHREDDED WHEAT,20C,30JUN
  84
10 DATA 10,HEFTY,TRASH BAGS,25C,31OCT83
11 DATA 11,WHEATSWORTH,CRACKERS,12C,30JUN
   84
12 DATA 12,KRAFT,JELLY,10C,NONE
```

```
13 DATA 13,PHILADELPHIA,CREAM CHEESE,10C,
   NONE
14 DATA 14,PREGO,SPAGETTI SAUCE,20C,30JUN
   84
500 DATA END
501 PRINT "{CLR}":REM CLEAR SCREEN
502 PRINT "COUPON FILE"
503 PRINT
504 LET T=0
520 PRINT "<L>{2 SPACES}LIST ALL ENTRIES"
525 PRINT
530 PRINT "<B>{2 SPACES}BRAND NAME "
533 PRINT
535 PRINT "<P>{2 SPACES}PRODUCT"
537 PRINT
540 PRINT "<D>{2 SPACES}EXPIRATION DATE"
545 PRINT
550 PRINT "<H>{2 SPACES}INSTRUCTIONS"
555 PRINT
560 PRINT "<C>{2 SPACES}COMMAND MODE"
565 PRINT
570 PRINT "CHOICE ";
580 INPUT K$
590 IF (K$<>"L")*(K$<>"B")*(K$<>"P")*(K$<
   >"D")*(K$<>"H")*(K$<>"C")THEN GOTO 57
   0
600 IF K$="L" THEN GOSUB 1000
610 IF K$="B" THEN GOSUB 2000
615 IF K$="P" THEN GOSUB 7100
620 IF K$="D" THEN GOSUB 4000
630 IF K$="H" THEN GOSUB 6000
640 IF K$="C" THEN END
650 GOTO 501
1000 PRINT "{CLR}":REM CLEAR SCREEN
1010 PRINT "COUPON LIST"
1015 PRINT
1100 GOSUB 10000
1200 IF A$="END" THEN RETURN
1550 PRINT "{CLR}":REM CLEAR SCREEN
1600 GOSUB 9000
1700 GOTO 1000
2000 PRINT "{CLR}":REM CLEAR SCREEN
2200 PRINT "ENTER BRAND"
2300 INPUT J$
2400 PRINT
2410 PRINT "BRAND: ";J$
2420 PRINT
2430 PRINT
2500 GOSUB 10000
2600 IF A$="END" THEN GOTO 3300
3000 IF B$<>J$ THEN GOTO 2500
3100 GOSUB 9000
3150 LET T=1
3200 GOTO 2500
3300 IF T=1 THEN GOTO 3600
3350 PRINT J$;" NOT ON FILE"
3360 PRINT
3400 PRINT "HIT RETURN TO CONTINUE"
3450 PRINT
3500 INPUT K$
3600 RETURN
4000 PRINT "{CLR}":REM CLEAR SCREEN
4200 PRINT "ENTER EXPIRATION DATE"
4300 INPUT J$
4400 PRINT
4410 PRINT "DATE: ";J$
4420 PRINT
4430 PRINT
4500 GOSUB 10000
4600 IF A$="END" THEN GOTO 5300
5000 IF E$<>J$ THEN GOTO 4500
5100 GOSUB 9000
5150 LET T=1
5200 GOTO 4500
5300 IF T=1 THEN GOTO 5600
5350 PRINT "NONE EXPIRE ";J$
5360 PRINT
5400 PRINT "HIT RETURN TO CONTINUE"
5410 PRINT
5500 INPUT K$
5600 RESTORE
5700 RETURN
6000 PRINT "{CLR}":REM CLEAR SCREEN
6200 PRINT "TO ENTER A FILE, PLACE"
6350 PRINT "THE COMPUTER IN COM-"
6400 PRINT "MAND MODE (THE MODE"
6450 PRINT "BEFORE YOU TYPE RUN)."
6500 PRINT "TYPE IN THE LINE NUM-"
6550 PRINT "BER,' DATA ',THE FILE"
6600 PRINT "NUMBER, THE BRAND NAME"
6650 PRINT "THE PRODUCT,. THE VALUE"
6700 PRINT "AND EXPIRATION DATE."
6710 PRINT
6720 PRINT
6800 PRINT "HIT RETURN TO CONTINUE"
6900 INPUT K$
7000 RETURN
7100 PRINT "{CLR}":REM CLEAR SCREEN
7200 PRINT "ENTER PRODUCT"
7300 INPUT J$
7400 PRINT
7410 PRINT "PRODUCT: ";J$
7420 PRINT
7430 PRINT
7500 GOSUB 10000
7600 IF A$="END" THEN GOTO 8300
8000 IF C$<>J$ THEN GOTO 7500
8100 GOSUB 9000
8150 LET T=1
8200 GOTO 7500
8300 IF T=1 THEN GOTO 8600
8350 PRINT J$;" NOT ON FILE"
8360 PRINT
8400 PRINT "HIT RETURN TO CONTINUE"
8410 PRINT
8500 INPUT K$
8600 RESTORE
8610 RETURN
9000 PRINT "FILE NUMBER>";A$
9010 PRINT
9100 PRINT "BRAND{6 SPACES}>";B$
9110 PRINT
9150 PRINT "PRODUCT{4 SPACES}>";C$
9160 PRINT
9200 PRINT "VALUE{6 SPACES}>";D$
9210 PRINT
9300 PRINT "EXP. DATE{2 SPACES}>";E$
9310 PRINT
9400 PRINT
9410 PRINT "HIT X TO ABORT TO MENU"
9420 PRINT "HIT RETURN TO CONTINUE"
9500 INPUT K$
9540 IF K$="X" THEN RESTORE
9550 IF K$="X" THEN GOTO 501
9600 RETURN
10000 READ A$
10100 IF A$="END" THEN RESTORE
10110 IF A$="END" THEN RETURN
10200 READ B$,C$,D$,E$
10300 RETURN
```

Two Great Game Cartridges for Your Texas Instruments 99/4A™
Guaranteed to work on all present and future versions of the TI 99/4A™

AMBULANCE™



DRIVING DEMON™

Funware™
A Subsidiary of Creative Software

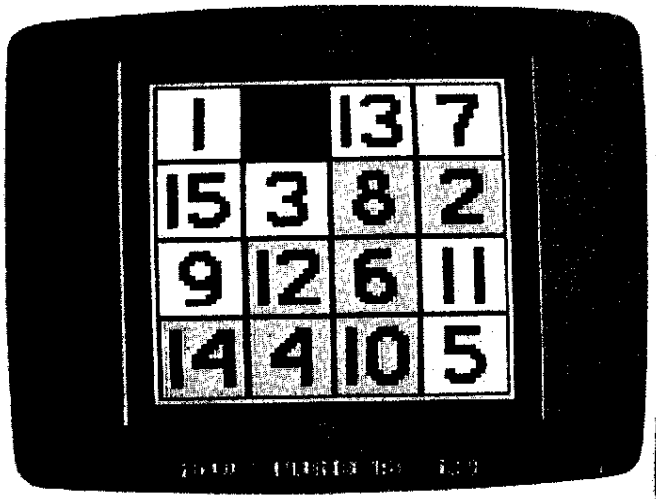
230 E. Caribbean Drive • Sunnyvale, CA 94089

Ambulance™ and Driving Demon™ are trademarks of Funware, Inc.
"Texas Instruments," and "Texas Instruments 99/4A" "TI 99/4A" are trademarks of Texas Instruments

Mosaic Puzzle

Bruce Jordan

Remember that once-popular sliding-squares game? With only one free space, you tried to move the colored plastic tiles around to get a particular sequence or color pattern. Although it was a challenge, this computer variation of the game can be a mind-boggling test of skill and dexterity. Versions for VIC, 64, Atari, and the TI-99/4A.



A game is just underway in the TI version of "Mosaic Puzzle."

"Mosaic Puzzle" is a computer version of those sliding-squares puzzles that used to drive people nuts before the advent of Rubik's Cube. It can run on either an unexpanded or 3K expanded VIC. The object of the game is to arrange the 15 numbered squares (hexadecimal numbers 1-F in this version) into some predetermined order by sliding them around in their frame. The first few moves are easy, but as the game progresses, it gets a lot more complicated. You'll find yourself rearranging everything just to get the last few squares in place.

This version of the game has a timer for up to 23 hours, 59 minutes, 59 seconds, and a chicken switch. It also automatically checks for the winning order and allows you to go back to the puzzle the way you left it or reset it to the beginning arrangement.

When you start the game, you're asked if you wish to set a time limit. If you answer Y for yes, enter the time limit in one line with no spaces or punctuation between the values. For example, for a 1-hour, 23-minute limit, enter 012300.

Next, enter the goal order. This will be the order that you will try to match to win the game. When this is done, the upper half of the screen will clear, and the puzzle will appear. A moment later, the message !GO! will flash on the screen, along with a tone. The controls for moving the squares are as follows:

@ up
?/ down
= right
: left

If you succeed in getting the squares in the goal order, the message YOU WIN! appears on the screen, accompanied by a short tune and the elapsed time. If the time runs out before you are finished, you'll hear an unpleasant sound. If you want to stop the game, press RETURN and the screen will display the elapsed time. You can then restart the game, either as you left it or reset, by hitting RETURN a second time.

Below is a brief description of the program as originally written on the VIC.

Line 1 sets the limit of memory at 7600. This gives a place to store the image of the puzzle.

Lines 2-5 define variables. Note that S, SC, and SS are defined in two consecutive lines. This is done because there are two possible entry points when restarting the game.

Lines 10-66 GET the time limit.

Lines 70-290 GET the goal order and make sure that no number is entered more than once.

Lines 300-365 put the puzzle on the screen, either from the data table or from memory, then

print the go message and start the timer.

Lines 360-560 check the timer, move the number squares, and check for the winning order.

Lines 570-710 print the winning or losing message, display the elapsed time, and play the sound effects.

Lines 720-740 save the position of the blank square, then clear the variables and reset the game.

Lines 745-790 are the DATA statements for the puzzle's beginning order and the winning tune.

Program 1: Mosaic Puzzle – VIC Version

BEGINNING PROGRAMMERS

If you're new to computing, please read "How To Type COMPUTE!'s Programs" and "A Beginner's Guide To Typing In Programs."

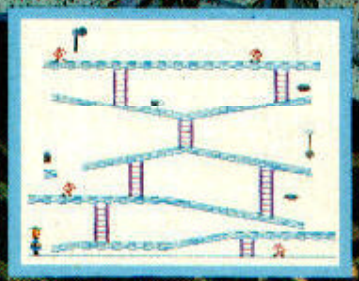
```
1 POKE55,176:POKE56,29:CLR
2 S=7845:SC=38565:SS=7603:GOTO4
3 SS=7603:S=PEEK(SS)*256+PEEK(SS+1):SC=S+
  30720
4 DIMA$(16)
5 PRINT"{CLR}":G=8018:X=0:DX=1:P=38738:V=
  36878:S1=36876:S2=36877:POKEV,15
10 PRINT"TIME LIMIT? {RVS}{GRN}Y{OFF}
   {RVS}{PUR}N{OFF}{BLU}"
20 GETA$:IFA$=""THEN20
30 IFA$<"N"ORA$>"Y"THEN20
40 IFA$="O"ANDA$<>"X"THEN20
50 IFA$="N"THEN70
60 PRINT"{CLR}":INPUT"HRS:MINS:SEC";T$:H=
   1:IFLEN(T$)<>6THEN60
62 IFLEFT$(T$,2)>"23"ORLEFT$(T$,2)<"0"THE
   N60
64 IFMID$(T$,3,2)>"59"ORMID$(T$,3,2)<"0"T
   HEN60
66 IFRIGHT$(T$,2)>"59"ORRIGHT$(T$,2)<"0"T
   HEN60
70 PRINT"{CLR}"TAB(24)"TYPE IN GOAL ORDER
   "
80 PRINT"{DOWN}{3 SPACES}1 2 3 4 5 6 7 8
   {SPACE}9"SPC(8)"A B C D E F {RVS}S
   {OFF}"TAB(49)"IN ANY ORDER"
90 PRINTTAB(118)"GOAL"TAB(30)"[4 Y]"
100 FORK=0TO3:POKE7996+K,100:POKE8106+K,9
    9:POKE38716+K,0:POKE38826+K,0:NEXTK
110 FORK=22TO88STEP22:POKE7995+K,103:POKE
    8000+K,101:POKE38715+K,0:POKE38720+K,
    0:NEXTK
130 FORI=1TO16STEP1
140 GETA$(I):IFA$(I)=""THEN140
150 FORL=I-1TO0STEP-1:IFA$(I)=A$(L)THEN14
    0
160 NEXTL
165 IFA$(I)="S"THEN190
170 IFA$(I)<="0"ORA$(I)=>"G"THEN140
180 IFA$(I)=>":"ANDA$(I)<="@"THEN140
190 B=VAL(A$(I)):C=B+48:IFB=0THEN220
200 POKE38467+2*B,7:POKEG+X,C:POKEP+X,0:X
    =X+DX:IFX=4THENG=G+22:P=P+22:X=0
210 NEXTI:IFI=17THEN300
220 IFA$(I)="A"THENB=1
230 IFA$(I)="B"THENB=2
240 IFA$(I)="C"THENB=3
250 IFA$(I)="D"THENB=4
260 IFA$(I)="E"THENB=5
270 IFA$(I)="F"THENB=6
272 B2=B
275 IFA$(I)="S"THENB=7:B2=32
280 POKE38492+2*B,7:POKEG+X,B2:POKEP+X,0:
    X=X+DX:IFX=4THENG=G+22:P=P+22:X=0
290 NEXTI
300 FORR=0TO132:POKE7702+R,32:POKE38422+R
    ,1:NEXTR
310 PRINT"{HOME}"TAB(29)"PUZZLE"TAB(29)"
    [6 Y]"
320 FORK=0TO3:POKE7754+K,100:POKE38474+K,
    0:POKE7864+K,99:POKE38584+K,0:NEXTK
330 FORK=22TO88STEP22:POKE7753+K,103:POKE
    38473+K,0:POKE7758+K,101:POKE38478+K,
    0:NEXTK
335 IFP1=0THEN340
336 FORK=0TO66STEP22:FORJ=0TO3:POKE7776+J
    +K,PEEK(7605+J+K)
337 POKE38496+J+K,PEEK(7609+J+K):NEXTJ:NE
    XTK:GOTO352
340 READA,B,C:IFA=-1THEN355
350 POKE7776+A,B:POKE38496+A,C:GOTO340
352 READA,B,C:IFA=-1THEN355
353 GOTO352
355 FORT=1TO1500:NEXT
360 POKES1,235:POKEV,15:PRINT"{HOME}"TAB(
    228)"{RVS}{RED}!GO!{OFF}{BLU}"
365 FORT=1TO500:NEXT:PRINT"{HOME}"TAB(228
    )"{4 SPACES}":POKES1,0:TI$="000000"
370 IFH<>1THEN380
375 IFT$<=TI$THENT$=TI$:GOTO600
380 GETB$:IFB$=""THEN370
390 D=ASC(B$):ON-(D=47)-2*(D=58)-3*(D=61)
    -4*(D=64)-5*(D=13)GOTO410,440,470,500
    ,620
400 GOTO370
410 IFPEEK(S-22)=100THEN370
420 POKES,PEEK(S-22):POKESC,PEEK(SC-22):P
    OKES-22,32:POKESC-22,1:S=S-22:SC=SC-2
    2
430 GOSUB530:GOTO370
440 IFPEEK(S+1)=101THEN370
450 POKES,PEEK(S+1):POKESC,PEEK(SC+1):POK
    ES+1,32:POKESC+1,1:S=S+1:SC=SC+1
460 GOSUB530:GOTO370
470 IFPEEK(S-1)=103THEN370
480 POKES,PEEK(S-1):POKESC,PEEK(SC-1):POK
    ES-1,32:POKESC-1,1:S=S-1:SC=SC-1
490 GOSUB530:GOTO370
500 IFPEEK(S+22)=99THEN370
510 POKES,PEEK(S+22):POKESC,PEEK(SC+22):P
    OKES+22,32:POKESC+22,1:S=S+22:SC=SC+2
    2
520 GOSUB530:GOTO370
530 FORM=0TO66STEP22:FORN=0TO3STEP1
540 W=PEEK(7776+M+N):IFW-128>0THENW=W-128
550 IFW<>PEEK(8018+M+N)THENRETURN
560 NEXTN:NEXTM
570 T$=TI$:PRINT"{HOME}"TAB(226)"{RVS}
    {YEL}YOU WIN!{OFF}{BLU}"
580 READN,D:IFN=-1THEN620
585 POKES1,N:FORT=1TOD:NEXT:GOTO580
600 PRINT"{HOME}"TAB(225)"{RVS}{GRN}!YOU
    {SPACE}LOSE!{OFF}{BLU}":POKES2,135:PO
    KES1,128:FORT=1TO500:NEXT
610 POKES2,0:POKES1,0
620 T$=TI$:FORK=0TO66STEP22:FORJ=0TO3:POK
    E7605+K+J,PEEK(7776+K+J)
630 POKE7609+K+J,PEEK(38496+K+J):NEXTJ:NE
```

Stake a Claim on the most exciting New Game for your Atari Home Computer.

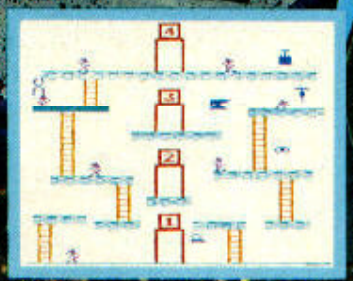
- From the programming team that brought you "Robot Attack", "Defense Command" and many other great Arcade games for your TRS-80™
- 100% machine language
- 16K ROM Cartridge, the largest available anywhere!
- Written specifically for the Atari® — not a converted Apple® game.
- Ten different rounds
- Difficulty adjustment
- High score table
- Demo mode
- Spectacular sound and graphics
- Runs on any 400/800 with at least 16K memory
- Only $49.95

Available also For the 5200

Here are just three of ten rounds in the game:



Round 1: The Mine Shaft.
Sharpen your miner's skill!



Round 3: The Transporters.
Beam yourself up!

Round 10: The Cannon.
Shoot yourself to the top!

BIG FIVE SOFTWARE

P.O. Box 9078-185, Van Nuys, CA 91409 · (213) 782-6861

To order, see your local dealer. If they do not have "Miner 2049er" then send a check or money order to us for $49.95 plus $2.00 for shipping and handling for immediate shipment! (California residents add 6½% sales tax.)

"Atari," "Apple," and "TRS-80" are trademarks of Atari Inc., Apple Computer, and Tandy Corp., respectively.

```
820 IF STRIG(0)=0 THEN 820
830 GOSUB 880:GOSUB 860:GOSUB 870
840 POKE 18,T1:POKE 19,T2:POKE 20,T3
    :GOTO 410
850 END
860 FOR I=0 TO 3:POSITION 2,I+4:PRIN
    T #6;PUZZLE$(I*4+1,I*4+4):NEXT I
    :RETURN
870 FOR I=0 TO 3:POSITION 14,I+4:PRI
    NT #6;GOAL$(I*4+1,I*4+4):NEXT I:
    RETURN
880 GRAPHICS 2+16:SETCOLOR 0,1,4:SET
    COLOR 1,5,4:SETCOLOR 2,8,8:SETCO
    LOR 3,10,6:SETCOLOR 4,0,0
890 POKE 756,CHBASE/256:POSITION 5,2
    :PRINT #6;"hex puzzle"
900 POSITION 2,3:PRINT #6;"____":POS
    ITION 14,3:PRINT #6;"____"
910 FOR I=4 TO 7:POSITION 1,I:PRINT
    #6;"\{4 SPACES}]":POSITION 13,I:
    PRINT #6;"\{4 SPACES}]":NEXT I
920 POSITION 2,8:PRINT #6;"^^^^":POS
    ITION 14,8:PRINT #6;"^^^^"
930 POSITION 1,9:PRINT #6;"puzzle":P
    OSITION 14,9:PRINT #6;"goal":RET
    URN
```

"Mosaic Puzzle," Atari version.

Program 4: Mosaic Puzzle — TI Version

by Rick Rothstein

```
190 CALL MAGNIFY(4):: RANDOMIZE ::
    DIM TILE(16),TEMP(16):: TEMP(16
    )=16 :: FR=153
200 CALL CLEAR :: CALL SCREEN(11)::
     CALL CHARSET :: CALL CHAR(35,"
    0",71,RPT$("0",12)&"FF0000FF")
210 DISPLAY AT(1,9):RPT$("G",12)::
    DISPLAY AT(2,9):"OPTION##MENU"
220 DISPLAY AT(3,9):RPT$("H",12)::
    DISPLAY AT(10,4):"PRESS####FOR"
     :: DISPLAY AT(11,4):"HHHHH####
    HHH"
230 DISPLAY AT(13,6):"1######NUMBER
    #PUZZLE" :: DISPLAY AT(16,6):BEE
    P:"2######LETTER#PUZZLE"
240 WASTE=RND :: CALL KEY(0,K,ST)::
     IF ST=0 THEN 240
250 IF K=49 THEN I=0 ELSE IF K=50 T
    HEN I=1 ELSE IF K=7 THEN 700 EL
    SE CALL SOUND(150,110,0):: GOTO
     240
260 CALL CLEAR :: CALL SCREEN(4)::
    IF I=0 THEN RESTORE 710 ELSE RE
    STORE 790
270 FOR I=80 TO 136 STEP 4 :: READ
    A$,B$ :: CALL CHAR(I,"FFFF"&A$&
    "FFFFFFFF"&B$&"FFFF"):: NEXT I
    :: CALL CHAR(140,RPT$("0",64))
280 CALL CHAR(71,"00000000000000003
    030303030303030000FFFF")
290 CALL CHAR(74,"00003F3F303030300
    000FCFC0C0C0C0C303030303F3F0000
    0C0C0C0CFCFC",78,RPT$("0C",8)&"0
    000000FFFF")
300 RESTORE 870 :: FOR I=38 TO 47 :
    : READ A$ :: CALL CHAR(I,A$)::
    NEXT I
310 CALL COLOR(5,15,2,6,15,2):: CAL
    L HCHAR(2,7,71,20):: DISPLAY AT
    (3,5):"GJ"&RPT$("I",16)&"KG"
320 FOR I=4 TO 16 STEP 4 :: DISPLAY
     AT(I,1):RPT$("####GH"&RPT$("G"
    ,16)&"NG####",4):: NEXT I
330 DISPLAY AT(20,5):"GL"&RPT$("O",
    16)&"MG" :: CALL HCHAR(21,7,71,
    20)
340 GOSUB 890 :: FOR I=1 TO 15 :: T
    ILE(I)=I :: NEXT I
350 FOR J=1 TO 15 :: R=1+INT(RND*(1
    6-J)):: TEMP(J)=TILE(R):: TILE(
    R)=TILE(16-J):: NEXT J
360 N=0 :: FOR I=1 TO 14 :: FOR J=1
     TO 15-I :: IF TEMP(I)>TEMP(I+J
    )THEN N=N+1
370 NEXT J :: NEXT I :: IF N/2<>INT
    (N/2)THEN TEMP(16)=TEMP(15):: T
    EMP(15)=TEMP(14):: TEMP(14)=TEMP
    (16):: TEMP(16)=16
380 FOR I=1 TO 16 :: TILE(I)=TEMP(I
    ):: NEXT I :: N=0 :: SP=16
390 FOR I=22 TO 124 STEP 34 :: FOR
    J=62 TO 164 STEP 34 :: N=N+1 ::
     CALL LOCATE(#TILE(N),I,J):: NE
    XT J :: NEXT I
400 MOVE$="" :: TOTAL=0 :: DISPLAY
    AT(24,1):CHR$(32+6*DIR)&"#####(
    )*+*,-(.+/:##########"&CHR$(32 7
    *NO):: CALL SOUND(150,666,0)::
    GOTO 430
410 CALL SOUND(150,110,0)
420 WASTE=RND :: IF K=73 OR K=78 OR
     K=105 OR K=110 THEN 450
430 CALL KEY(1,KK,ST):: IF KK=18 TH
    EN 670 ELSE CALL JOYST(1,X,Y)::
     IF ABS(X)+ABS(Y)=8 OR X+Y=0 TH
    EN 450
440 IF X/4=(-1)^(1-DIR)THEN 510 ELS
    E IF X/4=(-1)^(2-DIR)THEN 550 E
    LSE IF Y/4=(-1)^(2-DIR)THEN 630
     ELSE IF Y/4=(-1)^(1-DIR)THEN 5
    90
450 CALL KEY(0,K,ST):: IF ST=0 THEN
     430 ELSE IF K=45 THEN 670 ELSE
     IF K=83-HORZ OR K=115-HORZ THE
    N 510
460 IF K=68+HORZ OR K=100+HORZ THEN
     550 ELSE IF K=88-VERT OR K=120
```

TI-99/4A Notes

Rick Rothstein

The TI/99-4A version of "Mosaic Puzzle" requires Extended BASIC. In this version, you have the option of requesting either letters (A-O) or numbers (1-15) within a 4-by-4 frame. When you have entered your choice, the game board with its lettered or numbered blocks appears in a scrambled order. The object of the game is to slide the blocks about, one at a time, to bring them to one of several preselected patterns. Some patterns that you can try to duplicate are given in the table.

Move the lettered or numbered blocks around the game board with a joystick or the keyboard (E, S, D, and X keys). You actually have a choice of moving either the free space (hole) or the labeled blocks. The game is initially set to move the free space, but by pressing I (note the appearance of the left-right arrow symbol in the lower-left corner of the screen), you can move the labeled blocks.

Each move that you make is tallied, and the total number of moves is given at the bottom of the screen. Moves are normally accompanied by a sliding noise (notice the note symbol at the lower-right corner of the screen). If this noise becomes annoying, press N and the noise will cease (the note will also disappear).

At certain times during the game, you may wish to retrace your previous moves. Press - (minus sign) or hit the fire button to step back through each preceding move. With this option, a maximum of 250 moves can be recalled.

Once you've achieved the desired preselected pattern from its scrambled beginnings, you can restore the original game board pattern by pressing FCTN (REDO) and challenge yourself or others to beat your tally.

Other options available to you during the game are:

Keystroke	Description
FCTN (BACK)	Returns to the letter or number option menu
FCTN (BEGIN)	Starts a new game
FCTN (ERASE)	Ends the program

Possible Patterns For Puzzle, TI Version

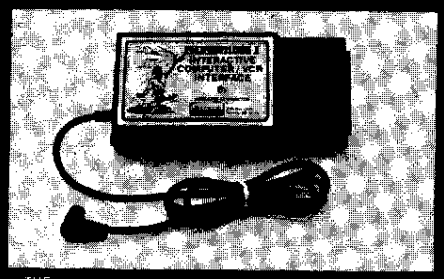
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	1 5 9 13 2 6 10 14 3 7 11 15 4 8 12	7 8 9 10 6 1 2 11 5 4 3 12 15 14 13
Horizontal	Vertical	Spiral
1 2 3 4 12 13 14 5 11 15 6 10 9 8 7	12 2 1 15 7 9 10 4 11 5 6 8 14 13 3	A B C D E F G H I J K L M N O
Peripheral	Adds To Thirty	Horizontal
A E I M B F J N C G K O D H L	G H I J F A B K E D C L O N M	A B C D L M N E K O F J I H G
Vertical	Spiral	Peripheral
F O G B I N D H E L M J A C K	E L F J A M B D O C K N I G H	
Words(1)	Words(2)	

```
    -VERT THEN 590 ELSE IF K=69+VER
    T OR K=101+VERT THEN 630
470 IF K=6 THEN CALL HCHAR(24,4,32,
    26):: GOSUB 890 :: GOTO 380 ELS
    E IF K=7 THEN 700
480 IF K=15 THEN CALL DELSPRITE(ALL
    ):: GOTO 200 ELSE IF K=14 THEN
    CALL HCHAR(24,4,32,26):: GOTO 34
    0 ELSE IF ST=-1 THEN 420
490 IF K=78 OR K=110 THEN NO=1-NO :
    : FR=153+NO*30000 :: CALL HCHAR
    (24,30,39-7*NO):: GOTO 420
500 IF K=73 OR K=105 THEN HORZ=15-H
    ORZ :: VERT=19-VERT :: DIR=1-DI
    R :: CALL HCHAR(24,3,32+6*DIR):
    : GOTO 420 ELSE 410
510 IF SP=1 OR SP=5 OR SP=9 OR SP=1
    3 THEN 410 ELSE SP=SP-1 :: CALL
     POSITION(#TILE(SP),ROW,COL)::
    CALL SOUND(4000,FR,14*NO)
520 IF MINUS=0 THEN MOVE$="L"&MOVE$
     ELSE MINUS=0
530 FOR I=COL TO COL+34 STEP 2 :: C
    ALL LOCATE(#TILE(SP),ROW,I):: N
    EXT I
540 TILE(SP+1)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,30):: GO
    TO 690
550 IF SP=4 OR SP=8 OR SP=12 OR SP=
    16 THEN 410 ELSE SP=SP+1 :: CAL
    L POSITION(#TILE(SP),ROW,COL)::
     CALL SOUND(4000,FR,14*NO)
560 IF MINUS=0 THEN MOVE$="R"&MOVE$
     ELSE MINUS=0
570 FOR I=COL TO COL-34 STEP -2 ::
    CALL LOCATE(#TILE(SP),ROW,I)::
    NEXT I
580 TILE(SP-1)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,30):: GO
    TO 690
590 IF SP>12 THEN 410 ELSE SP=SP+4
    :: CALL POSITION(#TILE(SP),ROW,
    COL):: CALL SOUND(4000,FR,14*NO
    )
```

```
600 IF MINUS=0 THEN MOVE$="D"&MOVE$
    ELSE MINUS=0
610 FOR I=ROW TO ROW-34 STEP -2 ::
    CALL LOCATE(#TILE(SP),I,COL)::
    NEXT I
620 TILE(SP-4)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,30):: GO
    TO 690
630 IF SP<5 THEN 410 ELSE SP=SP-4 :
    : CALL POSITION(#TILE(SP),ROW,C
    OL):: CALL SOUND(4000,FR,14*NO)
640 IF MINUS=0 THEN MOVE$="U"&MOVE$
    ELSE MINUS=0
650 FOR I=ROW TO ROW+34 STEP 2 :: C
    ALL LOCATE(#TILE(SP),I,COL):: N
    EXT I
660 TILE(SP+4)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,30):: GO
    TO 690
670 IF MOVE$="" THEN 410 ELSE MINUS
    =1 :: K=ASC(MOVE$):: MOVE$=SEG$
    (MOVE$,2,250):: TOTAL=TOTAL-2
680 IF K=82 THEN 510 ELSE IF K=76 T
    HEN 550 ELSE IF K=85 THEN 590 E
    LSE IF K=68 THEN 630
690 TOTAL=TOTAL+1 :: DISPLAY AT(24,
    19)SIZE(4)BEEP:TOTAL :: MOVE$=S
    EG$(MOVE$,1,250):: GOTO 420
700 CALL DELSPRITE(ALL):: CALL CLEA
    R :: END
710 DATA FEFEFEFEFEFEFEFEFEFEFEFE,7
    F7F7F7F7F7F7F7F7F7F7F7F7F,F8F0F3F
    FFFFFFEFCF8F1F0F0,1F0FCFCF8F1F3
    F7FFFFF0F0F
720 DATA F8F0F3FFFFFEFEFFFFF3F0F8,1
    F0FCFCF8F1F1F8FCFCF0F1F,FFFFFFF
    CF8F1F0F0FFFFFFFF,8F0F0F4FCFCF0
    F0FCFCFCFCF
730 DATA F0F0F3F3F0F0FFFFFFF3F0F8,1
    F1FFFFF3F1F8FCFCF8F1F3F,FCF8F1F
    3F0F0F1F3F3F1F8FC,1F1FFFFF3F1F8F
    CFCF8F1F3F
740 DATA F0F0F3FFFFFFFFFEFEFEFEFE,0
    F0FCF8F9F1F3F3F7F7F7F7F,FCF8F9F
    9FCF8F1F3F3F1F
8FC,3F1F9F9F3F1F8FCFCF8F1F3F
750 DATA FCF8F1F3F3F1F8FCFFFFF8F8,3
    F1F8FCFCF8F0F0FCF8F1F3F,CFCECCC
    CCCCCCCCCCCCCECF,0F0763F3F3F3F
    3F3F363070F
760 DATA F3F3F3F3F3F3F3F3F3F3F3F3,C
    FCFCFCFCFCFCFCFCFCFCFCFCF,CECCCCC
    FCFCFCFCFCECCCCCC,0703F3F3E3C78
    F1F3F7F0303
770 DATA CECCCCCFCFCFCFCFCFCCCCCE,0
    703F3F3E38707C3F3F30307,CFCFCFC
    FCECCCCCCCFCFCFCF,E3C3831333730
    303F3F3F3F3
780 DATA CCCCCCCCCCCCCFCFCFCCCCCE,0
    707FFFFF0F07E3F3F3E3070F
790 DATA FEFCF8F1E3E7E0E0E7E7E7E7,7
    F3F1F8FC7E70707E7E7E7E7,E0E0E7E
    7E7E0E0E7E7E7E0E0,1F0FC7E7C70F0F
    C7E7C70F1F
800 DATA F8F0E3E7E7E7E7E7E7E3F0F8,0
    F07E7FFFFFFFFFFFFE7070F,E0E0E7E
    7E7E7E7E7E7E7E0E0,3F1F8FC7E7E7E
    7E7C78F1F3F
810 DATA E0E0E7E7E7E0E0E7E7E7E0E0,0
    707FFFFFF1F1FFFFFFF0707,E0E0E7E
    7E7E0E0E7E7E7E7E7,0707FFFFFF1F1
    FFFFFFFFFFF
820 DATA F8F0E3E7E7E7E7E7E7E3F0F8,0
    F07E7FFFFFF8787E7E70707,E7E7E7E
    7E7E0E0E7E7E7E7E7,E7E7E7E7E70707
    E7E7E7E7E7
830 DATA F8F8FEFEFEFEFEFEFEFEF8F8,1
    F1F7F7F7F7F7F7F7F7F7F1F1F,FFFFFFF
    FFFFFFFFFFE7E3F0F8,E7E7E7E7E7E7E7
    E7E7C70F1F
840 DATA E7E7E7E6E4E0E0E0E6E7E7E7,C
    78F1F3F7FFFFF7F3F1F8FC7,E7E7E7E
    7E7E7E7E7E7E7E0E0,FFFFFFFFFFFFFF
    FFFFF0707
850 DATA E7E3E1E0E4E6E7E7E7E7E7E7,E
    7C787072767E7E7E7E7E7E7,E7E3E1E
    0E4E6E6E7E7E7E7E7,E7E7E7E7E76767
    270787C7E7
860 DATA F8F0E3E7E7E7E7E7E7E3F0F8,1
    F0FC7E7E7E7E7E7E7C70F1F
870 DATA 3060FF0000FF060C,0E0908086
    8F8F86,00446C54444444.007C44444
    4447C,0044444444281,007C40784040
    7C
880 DATA 00446454544C44,0044444444
    438.0078447844447R48
    44
890 FOR I=1 TO 16 STEP 2 :: CALL SP
    RITE(#I,76+4*I,16,193,1,#I+1,80
    +4*I,11,193,1):: NEXT I :: RETU
    RN
```


COMPUTE!
The Resource

AT LAST . . . AN INTERACTIVE COMPUTER/VCR INTERFACE FOR THE HOME!



THE PROMETHEUS 1™
INTERACTIVE COMPUTER/VCR INTERFACE

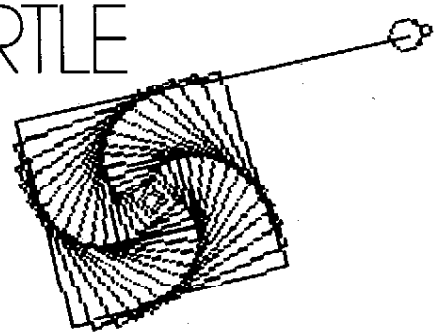
NOW, FOR THE FIRST TIME, YOU CAN HAVE A COMPLETELY INTERACTIVE COMPUTER/VIDEO SYSTEM AT HOME FOR LESS THAN $1,500. THE PROMETHEUS 1™ INTERFACE CONNECTS YOUR COMMODORE 64 OR VIC 20 COMPUTER TO YOUR CONSUMER VCR, GIVING YOU A WHOLE NEW RANGE OF ENTERTAINMENT AND EDUCATIONAL POSSIBILITIES NEVER BEFORE AVAILABLE IN THIS PRICE RANGE. SEND FOR OUR FREE LITERATURE AND OUR CATALOG OF ENTERTAINMENT/EDUCATIONAL INTERACTIVE COURSEWARE, OR SEND CHECK OR MONEY ORDER (PLUS $4.00 POSTAGE AND HANDLING) FOR THE PROMETHEUS 1™ INTERFACE.)

$49.95
Wa. St. Res. add 8% Sales Tax

SEND TO:
VIDEOBROOK™
P.O. BOX 19597
SEATTLE, WA 98109
OR CALL:
(206) 282-3636

INCLUDE YOUR COMPUTER WARRANTY NUMBER AND RECEIVE A FREE PROMETHEUS 1™ POSTER! ALLOW 6-8 WEEKS FOR DELIVERY. ORDER NOW FOR CHRISTMAS.

FRIENDS OF THE TURTLE



David D Thornburg, Associate Editor

Ed Emberley's Drawing Procedures

Part of the appeal of turtle graphics is that it allows complex pictures to be built from simple building blocks. This feature arises from the fact that each shape description or procedure describes the shape itself, independently of its starting point or orientation. For example, once a square is defined with the procedure:

```
TO SQUARE :SIZE
    REPEAT 4 [FORWARD :SIZE RIGHT 90]
END
```

the computer can use this procedure to create a square of any size at any starting location and orientation.

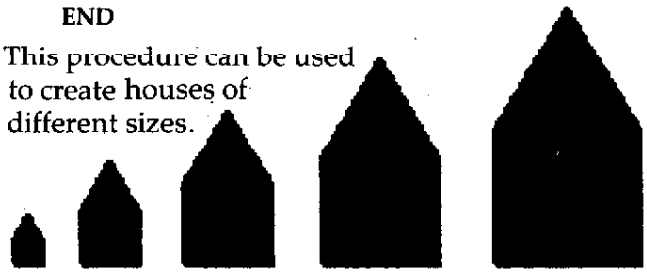
If the user has built up a set of useful geometric procedures, these can be combined to create more complex figures. If one also has a procedure for drawing triangles:

```
TO TRI :SIZE
    REPEAT 3 [FORWARD :SIZE RIGHT 120]
END
```

then a procedure for drawing a house can be created from a combination of a square and a triangle:

```
TO HOUSE :SIZE
    SQUARE :SIZE
    FORWARD :SIZE RIGHT 30
    TRI :SIZE
END
```

This procedure can be used to create houses of different sizes.



Many turtle graphics enthusiasts create libraries of basic figures from which quite interesting pictures can be created.

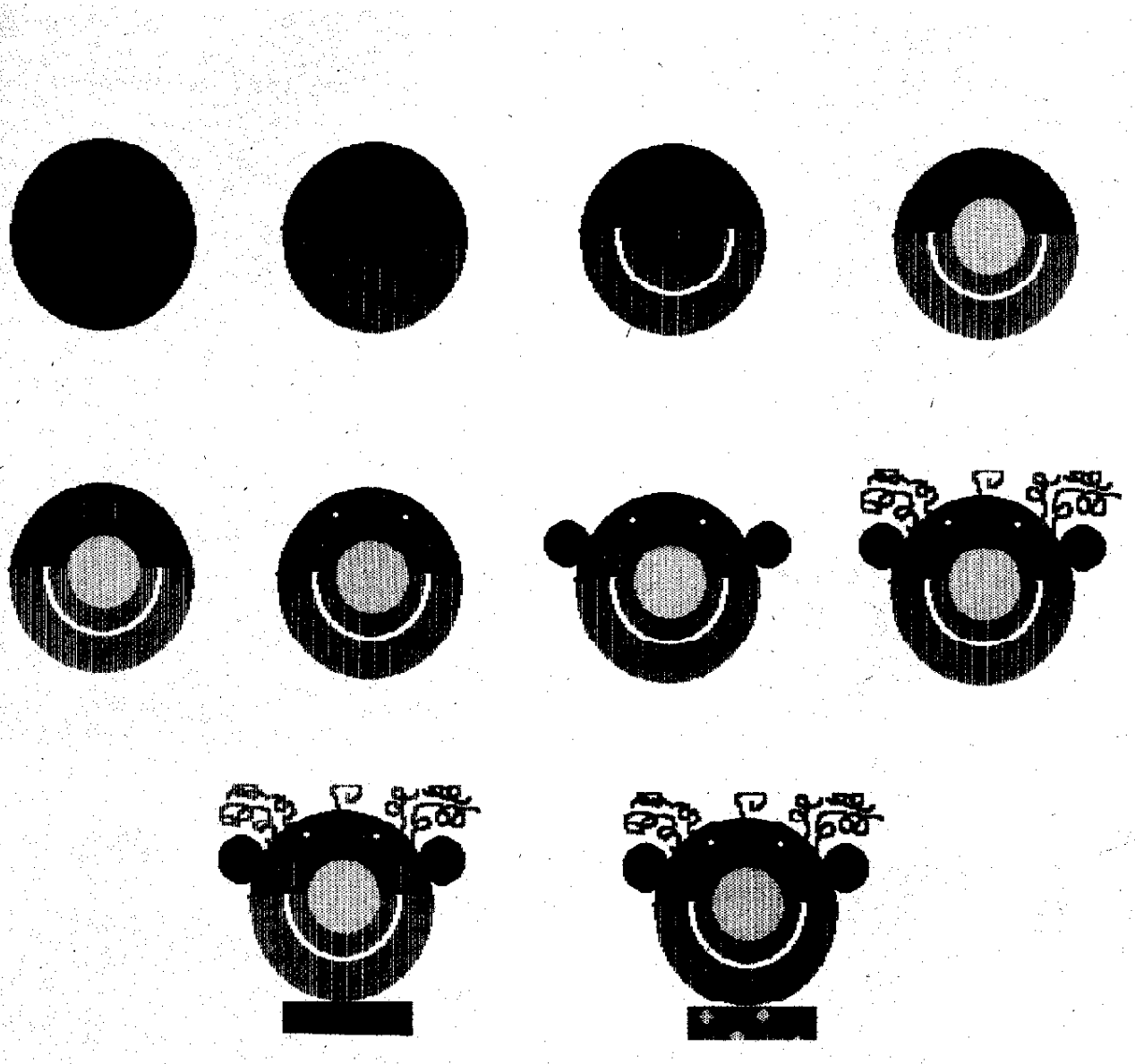
As an active proponent of turtle graphics and procedural problem-solving, I was delighted to find Ed Emberley's independent discoveries along these lines.

Ed Emberley has written several books on illustration for children. His books of particular interest to readers of this column would include: *Ed Emberley's Drawing Book of Animals, Ed Emberley's Drawing Book, Make a World, Ed Emberley's Big Orange Drawing Book,* and *Ed Emberley's Big Purple Drawing Book* (all published by Little, Brown and Co.).

Mr. Emberley's illustration technique is built on the idea that, just as words are created from an alphabet of letters, pictures can be created from an alphabet of shapes. He shows how to create myriad figures using circles, rectangles, arcs, lines, triangles, and other simple pieces. By building the figure piece by piece, the young artist is never overwhelmed by trying to deal with the whole figure at once. The following series of illustrations (courtesy of Mr. Emberley) shows how one can create a clown's head almost entirely from circles and circle parts.

If you were to create this figure using turtle graphics procedures, you would need only procedures for a circle, an arc, a rectangle, and the squiggles for the hair.

Ed Emberley does not normally use a computer to create his illustrations. The clown figures shown on the next page are a happy exception to that, as he created them on an Apple computer using the KoalaPad touch tablet with the Micro



Illustrator software. I am encouraging him to use Logo also to see how he likes it.

Just as Mr. Emberley's books can be a source of inspiration to those of us who build pictures using turtle graphics, they can also be wonderful tools for teaching procedural problem-solving for teaching people how to solve larger problems by breaking them into bite-sized chunks. For this reason I encourage the use of his drawing books by teachers of computer programming. Not only are the children learning to solve problems with procedures, but they are also learning how to create charming illustrations at the same time.

I created the next figure myself to show that almost *anyone* can learn to make pictures in this manner.

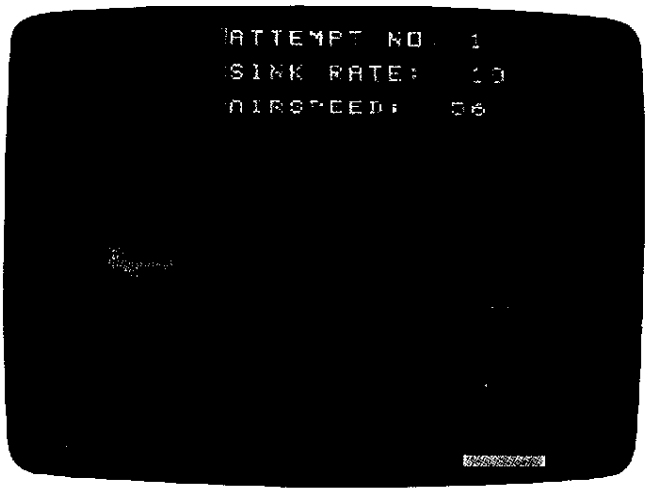
For those of us who have been in the field a long time, the discovery of Mr. Emberley's excellent contributions is refreshing. Clearly, he is a Friend of the Turtle!

Runway 180

Using Sprites In TI Extended BASIC

James Dunn

The efficient, remarkable sprite-handling ability of TI Extended BASIC is clearly evident in this game. The author discusses creating sprites and explores sprite manipulation. There are several valuable pointers here for those interested in graphics, animation, or game programming on the TI.



Your plane is on final approach. "Runway 180," TI version.

Using Sprites In TI Extended BASIC

One of the biggest problems in designing an arcade-type game in BASIC is that BASIC can move only one character at a time, usually slowly and usually not very smoothly. Ideally, we need the ability to move an object independently of the operation of the main program. Once set in motion, the object would continue in motion until acted upon by a new command from the main program. Sprites accomplish this.

Although a sprite is a type of subprogram that runs concurrently with a main program, the main program first must create the sprite, define its shape, and set it in motion. A sprite then continues its motion without requiring continuous control from the main program, except that the main program may at any time test the sprite for position, change the color or pattern, delete, or change its motion.

Included in TI-99/4A Extended BASIC are 11 commands to control sprites: CALL COLOR, CALL CHAR, CALL SPRITE, CALL PATTERN, CALL MAGNIFY, CALL MOTION, CALL POSITION, CALL LOCATE, CALL DISTANCE, CALL COINC, and CALL DELSPRITE. To illustrate the use of these commands, we'll look at an airplane landing game, "Runway 180." Try some examples for yourself to get a feel for sprite programming.

Creating Sprites

Certain considerations must be taken into account before sprites are created. If a special graphics character is to be used for the sprite, the character must be created by use of CALL CHAR. For example, in the game there are three special characters defined for the aircraft. One is with the wheels up (lines 430)-460), one is with the wheels down (lines 510-540), and one is debris after a crash (lines 550-580).

To create a special character, it is necessary to redefine an existing standard character. The standard characters correspond to the numbers 30 through 143 (part of what's called the ASCII number code). The new pattern is created by using CALL CHAR and is referenced by its ASCII number.

Before we choose which ASCII number to use, we must examine some other factors. CALL MAGNIFY can enlarge a sprite to one of four magnification factors. Factor four is used in the game (line 630). This enlarges the sprites to double-size pixels and uses a block of four

sequential characters. The ASCII number used to define the sprite must be evenly divisible by four and represents the upper-left character in the block of four. The next three ASCII numbers represent the lower-left, upper-right, and lower-right characters respectively in the block of four.

The sprite may be colored independently of the other characters in the same character set. In addition, the sprite with the lower sprite number (this is a different number than the ASCII number) will pass in front of (that is, *over*) the higher numbered sprite. Since the aircraft should pass in front of the tower, it should have a lower sprite number for each of its three configurations (line 610).

To set up a list of sprites, first number the lines on a sheet of paper from 30 to 143. Then, beside each number, write what set it belongs to (set 0 to 14). Since you may want to use letters or numbers in a screen display at the same time, mark out ASCII numbers 48 through 57 and 65 through 90. The remaining ASCII numbers can be used to define special characters for graphics and sprites.

For sprites, using CALL MAGNIFY (4), select four sequential numbers starting at one of the numbers evenly divided by four. Now you are ready to use CALL SPRITE.

CALL CLEAR will not remove a sprite from the screen. To completely clear the screen, you must also use CALL DELSPRITE (line 1350).

Sprites In Motion

Now that the sprite has been created, there are two ways of moving it around the screen. Let's call these two methods *absolute* and *relative*. The absolute method uses exact row and column positions via the CALL LOCATE command. The relative method uses row and column motion values via the CALL MOTION command.

The absolute method uses a loop with CALL JOYST to increment row and column variables, and then a CALL LOCATE to move the sprite one step each time the loop is executed. This is analogous to nonsprite methods of animation. The drawback in using this method is that the sprite does not move independently; the main program causes the move. A modified form of this method is used for the stall subroutine (line 1470) and the new approach routine (line 1380).

The relative method is similar, using a loop with CALL JOYST to increment row and column *motion* variables which are used in a CALL MOTION command. This allows the sprites to continue moving independently of the main program. By this method, the runway stripe is moved horizontally only (line 680) and the aircraft vertically only (also line 680).

The sprite's shape may be changed anytime during the program by using CALL PATTERN to substitute a different ASCII character number and therefore a different pattern. When the fire button is depressed (line 1130), the aircraft landing gear comes down (line 1190). The pattern is changed again if the aircraft crashes (line 1720).

Testing For Game Conditions

During the operation of the program, it may become necessary to test for certain conditions. For example, we see if the aircraft has touched down on the runway (line 690), if the tower has reached the left side of the screen (line 700), or if the aircraft is going off the top of the screen (line 710). CALL COINC is used to test for these conditions.

However, there is a problem with this method. Since the main program tests for coincidence only when CALL COINC is executed and since the sprite moves independently of the main program, it is quite possible to miss an exact coincidence when it occurs. For this reason a tolerance factor is included in CALL COINC. So the test is really for a range of + or - tolerance. If the tolerance is too large, coincidence can be returned too early. If the tolerance is too small, coincidence can be missed altogether. How large the tolerance should be depends upon two things: the speed of the sprite and the speed of the loop which is testing for coincidence.

The test for the tower reaching the left side of the screen is in both the main loop (line 700) and the stall loop (line 1480). The tolerance in the stall loop is much smaller because the execution speed is so fast and the sprite moves so slowly that coincidence is actually read twice before the sprite leaves the tolerance range. Trial and error is the only way to find out how large the tolerance should be.

However, after programming this game, it is obvious that very fast-moving sprites will require tolerance ranges that will make arcade-style, fast-action games nearly impossible in Extended BASIC. The problem is that the coincidence test is executed from the main program. If it were part of the sprite subprogram instead, it would be possible to keep the tolerance very small.

CALL POSITION and CALL DISTANCE both suffer from the same problem as CALL COINC. By the time a position or distance can be computed and returned to the main program, the sprite has moved elsewhere. But it is possible to stop the sprite by using a CALL MOTION before using CALL POSITION or CALL DISTANCE (line 1330), then to restart whatever motion is required.

Despite a few shortcomings, the sprite capabilities in Extended BASIC are remarkable. For true arcade-type play, machine language is still necessary, but Extended BASIC sprites will carry the programmer a lot closer to this goal.

Runway 180

```
130 CALL CLEAR :: CALL SCREEN(5)::
    CALL COLOR(1,16,1,2,16,1,3,16,1
    ,4,16,1,5,16,1,6,16,1,7,16,1,8,
    16,1)
140 DISPLAY AT(10,9):USING "RUNWAY
    180"
150 FOR B=0 TO 30 STEP 2 :: CALL SO
    UND(-10,110,30,110,30,2500,30,-
    8,B):: CALL SOUND(-10,110,30,11
    0,30,4000,30,-8,B):: NEXT B
160 CALL CLEAR :: DISPLAY AT(10,9):
    USING "PRESS" :: DISPLAY AT(12,
    9):USING "I-FOR INSTRUCTIONS"
170 DISPLAY AT(14,14):USING "OR" ::
     DISPLAY AT(16,9):USING "G-FOR
    GAME"
180 CALL KEY(0,K,S):: IF S<>1 THEN
    180
190 IF K=71 THEN 330
200 IF K=73 THEN 220
210 PRINT "ALPHA LOCK MUST BE ON" :
    : PRINT :: PRINT "TRY AGAIN" ::
     FOR DELAY=1 TO 200 :: NEXT DEL
    AY :: GOTO 160
220 CALL CLEAR :: PRINT "YOU ARE PI
    LOTING A JET" :: PRINT :: PRINT
     "AIRCRAFT WHICH HAS BEEN " ::
    PRINT :: PRINT "CLEARED TO LAND
     ON": :
230 PRINT "RUNWAY 180." :: PRINT ::
     PRINT :: GOSUB 310
240 CALL CLEAR :: PRINT "USE YOUR J
    OYSTICK TO CONTROL" :: PRINT ::
     PRINT "SINK RATE AND AIRSPEED.
    ": :
243 PRINT "JOYSTICK CONTROL-" :: PR
    INT
245 PRINT "LEFT: ACCELERATE" :: PRI
    NT "RIGHT: BRAKE" :: PRINT "UP:
     DECREASE SINK RATE"
247 PRINT "DOWN: INCREASE SINK RATE
    " :: PRINT
250 PRINT "FIREBUTTON CONTROLS LAND
    ING" :: PRINT :: PRINT "GEAR."
    :: PRINT :: PRINT :: GOSUB 310
    :: CALL CLEAR
260 PRINT "TO RECOVER FROM A STALL"
     :: PRINT :: PRINT "INCREASE AI
    RSPEED ABOVE 60." :: PRINT ::
    PRINT "IF YOU CANNOT STOP BEFO
    RE": :
270 PRINT "TOWER REACHES LEFT SIDE
    OF" :: PRINT :: PRINT "SCREEN,
    INCREASE AIRSPEED" :: PRINT
280 PRINT "TO 60 AND LIFT OFF FOR "
     :: PRINT :: PRINT "ANOTHER PAS
    S." :: PRINT :: PRINT :: GOSUB
    310 :: CALL CLEAR
290 PRINT "YOU MAY HAVE FOUR PASSES
    " :: PRINT :: PRINT "AT THE RUN
    WAY....." :: PRINT :: PRINT "BE
    WARE OF THE WIND SHIFTS!" :: PR
    INT :: PRINT
300 PRINT "GOOD LUCK!!!!" :: PRINT
    :: PRINT :: PRINT :: PRINT :: G
    OSUB 310 :: GO TO 330
310 PRINT :: DISPLAY AT(24,1):USING
     "HIT ANY KEY TO CONTINUE"
320 CALL KEY(0,R8,S8):: IF S8<>1 TH
    EN 320 ELSE RETURN
330 A1=1
340 REM INITIALIZE
350 A=0 :: B=-75 :: LC=0 :: CALL SC
    REEN(2)
360 CALL CLEAR :: CALL CHAR(33,"FFF
    FFFFFFFFFFFFF"):: CALL COLOR(1,
    8,1)
370 LC=0 :: FOR Z=1 TO 16 :: CALL H
    CHAR(Z,1,33,32):: NEXT Z
380 CALL CHAR(42,"FFFFFFFFFFFFFFFF"
    ):: CALL COLOR(2,13,1)
390 FOR Z=17 TO 20 :: CALL HCHAR(Z,
    1,42,32):: NEXT Z
400 RANDOMIZE
410 REM DEF CHAR
420 CALL CHAR(96,"00000000FFFFFFFF
    FFFFFFF0000000000000000FFFFFFFF
    FFFFFFFF")
430 CALL CHAR(120,"0030181C3F1F0700
    ")
440 CALL CHAR(121,"000000")
450 CALL CHAR(122,"00000000FCFF8000
    ")
460 CALL CHAR(123,"00000000")
470 CALL CHAR(104,"0000000071F151F
    ")
480 CALL CHAR(105,"0203030203030203
    ")
490 CALL CHAR(106,"00008080E0F8A8F8
    ")
500 CALL CHAR(107,"C040C0C040C0C0C0
    ")
510 CALL CHAR(124,"0030181C3F1F0705
    0000")
520 CALL CHAR(126,"00000000FCFF8884
    0000")
530 CALL CHAR(125,"00000000")
540 CALL CHAR(127,"00000000")
550 CALL CHAR(128,"0000000021F3B00
    ")
560 CALL CHAR(129,"00000000E56C300
    ")
570 CALL CHAR(130,"00000000")
580 CALL CHAR(131,"00000000")
590 REM DRAW DISPLAY
600 CALL SPRITE(#1,96,2,180,1,0,B):
    : CALL COLOR(#1,16)
610 CALL SPRITE(#2,120,2,10,245,A,0
    ):: CALL COLOR(#2,7)
620 CALL SPRITE(#3,104,2,110,250,0,
    -2)
630 CALL MAGNIFY(4)
640 FOR C5=1 TO 40 :: CALL LOCATE(#
    2,10,C5):: NEXT C5 :: GOSUB 870
650 REM MAIN LOOP
660 GOSUB 1120 :: GOSUB 890
670 IF J=0 THEN 690
680 CALL MOTION(#1,0,B,#2,A,0)
690 CALL COINC(#2,170,40,9,T)
700 CALL COINC(#3,110,1,4,DA)
710 CALL COINC(#2,240,40,9,E):: IF
    E=-1 THEN A=1 :: GOSUB 890 :: G
    OTO 680
720 IF DA=-1 THEN 1320
730 IF T<>-1 THEN 660
740 CALL MOTION(#2,0,0)
750 IF A>1 THEN GOSUB 920 :: GOSUB
```

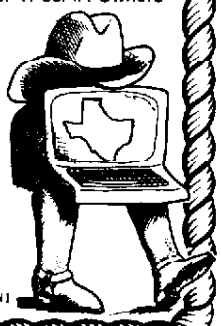
```
   960 :: GOTO 1660
760 IF LG=0 THEN 1660
770 GOTO 1760
780 REM UPDATE DISPLAY
790 IMAGE SINK RATE: ###
800 IMAGE RUNWAY ENDS ### YDS
810 IMAGE AIRSPEED: ###
820 IMAGE TOUCH DOWN
830 IMAGE SINK RATE TOO HIGH
840 IMAGE AIRSPEED TOO HIGH
850 IMAGE CRASH LANDING
860 IMAGE STALL WARNING!
870 DISPLAY AT(1,10)SIZE(20):USING
    "ATTEMPT NO. #":A1
880 RETURN
890 DISPLAY AT(3,10)SIZE(20):USING
    790:A
900 DISPLAY AT(5,10)SIZE(20):USING
    810:-B
910 RETURN
920 DISPLAY AT(7,5)SIZE(20):USING 8
    30
930 RETURN
940 DISPLAY AT(7,5)SIZE(20)BEEP:USI
    NG 840
950 DISPLAY AT(9,5)SIZE(20):USING "
    BOUNCE" :: RETURN
960 DISPLAY AT(9,5)SIZE(20):USING 8
    50
970 RETURN
980 CALL HCHAR(7,5,33,27):: DISPLAY
     AT(9,5)SIZE(20):USING 820
990 RETURN
1000 DISPLAY AT(9,5)SIZE(20):USING
     "WARNING "
1010 DISPLAY AT(11,5)SIZE(20):USING
      800:RE
1020 RETURN
1030 CALL HCHAR(7,5,33,27):: RETURN
1040 CALL HCHAR(9,5,33,27):: RETURN
1050 CALL HCHAR(11,5,33,27):: RETUR
     N
1060 DISPLAY AT(9,5)SIZE(20):USING
     "LIFT OFF" :: CALL HCHAR(11,5,
     33,27):: RETURN
1070 DISPLAY AT(3,10):USING "END OF
      RUNWAY " :: DISPLAY AT(5,10):
     USING "NEW APPROACH" :: DISPLA
     Y AT(7,10):USING "NECESSARY"
1080 RETURN
1090 PRINT "THAT'S 5 PASSES AT THE"
      :: PRINT :: PRINT "RUNWAY. TU
     RN IN YOUR" :: PRINT :: PRINT
     "PILOT LICENSE AND PUT";
1100 PRINT "SOMEONE ELSE IN THE" ::
      PRINT :: PRINT "COCKPIT" :: P
     RINT :: RETURN
1110 DISPLAY AT(7,9)BEEP SIZE(20):U
     SING 860 :: RETURN
1120 REM JOYST/ LANDING GEAR
1130 CALL KEY(1,RV,ST):: IF RV=18 A
     ND LG=0 THEN 1190
1140 CALL JOYST(1,X,Y):: IF X=0 AND
      Y=0 THEN GOSUB 1210 :: RETURN
1150 A=A-Y/4 :: B=B+X/4
1160 IF ABS(A)>127 THEN A=127*SGN(A
     )
1170 IF B>-50 THEN 1430
1180 J=1 :: RETURN
1190 CALL PATTERN(#2,124)
1200 A=A+3 :: B=B+20 :: LG=1 :: GOT
     O 1160
1210 REM COMPLICATIONS
1220 CP=INT(RND*16)
1230 IF CP=1 THEN B=B-1 :: GOTO 130
     0
1240 IF CP=6 THEN B=B+1 :: GOTO 130
     0
1250 IF CP=10 THEN A=A-1 :: GOTO 12
     80
1260 IF CP=15 THEN A=A+1 :: GOTO 12
     80
1270 J=0 :: RETURN
1280 IF ABS(A)>127 THEN A=127*SGN(A
     )
1290 GOTO 1310
1300 IF B<-127 THEN B=-127
1310 J=1 :: RETURN
1320 REM NEW APPROACH
1330 CALL MOTION(#2,0,0):: CALL POS
     ITION(#2,R4,C4)
1340 IF A1>4 THEN 1400
1350 CALL DELSPRITE(#1,#3):: CALL C
     LEAR
1360 GOSUB 1070
1370 CALL PATTERN(#2,120)
1380 FOR X=C4 TO 255 :: CALL LOCATE
     (#2,INT(R4),X):: R4=R4-(R4/(25
     5-C4)):: NEXT X
1390 A1=A1+1 :: GOTO 340
1400 CALL DELSPRITE(ALL):: CALL CLE
     AR
1410 GOSUB 1090
1420 FOR DELAY=1 TO 900 :: NEXT DEL
     AY :: GOTO 1970

1430 REM STALL
1440 GOSUB 1110
1450 CALL MOTION(#2,0,0)
1460 CALL POSITION(#2,SR,SC)
1470 CALL LOCATE(#2,SR,SC)
1480 CALL COINC(#2,170,40,2,T)
1490 CALL COINC(#3,110,1,2,DE):: IF
      DE=-1 THEN A1=A1+1 :: GOSUB 8
     70 :: IF A1>4 THEN 1400
1500 IF T=-1 THEN 1660
1510 SR=SR+4
1520 CALL KEY(1,RV,ST)
1530 IF RV=18 AND LG=1 THEN 1610
1540 CALL JOYST(1,X,Y):: IF X=0 AND
      Y=0 THEN 1470
1550 B=B+X/4
1560 REM
1570 IF B<-60 THEN 1640
1580 CALL MOTION(#1,0,B)
1590 GOSUB 890
1600 GOTO 1470
1610 CALL PATTERN(#2,120)
1620 A=A-3 :: B=B-22 :: LG=0
1630 GOTO 1560
1640 GOSUB 1030
1650 RETURN
1660 REM CRASH
1670 CALL MOTION(#1,0,0,#2,0,0,#3,0
     ,0,#4,0,0)
1680 CALL SOUND(1000,-7,0)
1690 FOR P=1 TO 10
1700 CALL SCREEN(2)
```

```
1710 CALL SCREEN(1A):: NEXT P :: CA
     LL SCREEN(2)
1720 CALL PATTERN(#2,128)
1730 FOR DELAY=1 TO 400 :: NEXT DEL
     AY
1740 CALL DELSPRITE(ALL)
1750 GOTO 1970
1760 REM TOUCHDOWN/BRAKE/T&G
1770 GOSUB 980 :: IF B<-55 THEN 194
     0
1780 CALL JOYST(1,X,Y):: B=B+X/2
1790 IF B>-1 THEN 1880
1800 CALL MOTION(#1,0,B)
1810 CALL COINC(#3,110,1,4,DA)
1820 IF DA=-1 THEN RE=0 :: GOSUB 10
     10 :: GOTO 1660
1830 CALL DISTANCE(#3,110,1,RQ)
1840 RE=INT(SQR(RQ)):: GOSUB 1000 :
     : GOSUB 900
1850 CALL KEY(1,RV,ST):: IF RV=18 A
     ND B<-60 THEN GOSUB 1060 :: A=
     A-2 :: GOTO 187
0
1860 GOTO 1780
1870 CALL MOTION(#2,A,0):: FOR DELA
     Y=1 TO 200 :: NEXT DELAY :: GO
     TO 650
1880 REM SCORING
1890 CALL MOTION(#1,0,0,#2,0,0,#3,0
     ,0,#4,0,0)
1900 CALL DELSPRITE(ALL):: CALL CLE
     AR
1910 PRINT "CONGRATULATIONS !":
1920 PRINT "YOUR SCORE IS :";(RE/A1
     )*10:
1930 GOTO 1990
1940 A=A-2 :: CALL MOTION(#2,A,0)::
      GOSUB 940
1950 FOR DELAY=1 TO 20 :: NEXT DELA
     Y
1960 A=A+2 :: GOSUB 1030 :: GOSUB 1
     040 :: GOTO 650
1970 REM PLAY AGAIN
1980 CALL CLEAR
1990 PRINT "PLAY AGAIN (Y/N)?"
2000 CALL KEY(2,RV,SV)
2010 IF SV=0 THEN 2000
2020 IF RV=15 THEN 2050
2030 IF RV=18 THEN 330
2040 GOTO 1990
2050 END
```


COMPUTERIZED GIVEAWAY

TEXAS STYLE A Contest for TI 99/4A Owners



HOWDY PARDNER! How would you like a chance to expand your computer system or software library ABSOLUTELY FREE? For the mere cost of a stamp and postcard, you cowboys and cowgirls can have a chance to win a valuable peripheral or hundreds of dollars worth of software. Be sure to throw your hat into the ring to round up one of these fabulous prizes. Enter as often as you like. Don't delay! For an official entry form send your name and address to: TEXware Associates, 350 First North St., Wellington, IL 60973, or circle the reader service card number.

RETAIL VALUE OF PRIZES IS OVER $1,750.00

Sponsored By: TEXware Associates

In Cooperation With: Doryt Systems, Inc. Dhein's Computer Connection

The Doryt Connection: Because you shouldn't have to pay for something you don't need.

Paraprint 18A can save TI-99/4A users up to $300.

Our Paraprint 18A interfaces between the TI-99/4A and any parallel printer, eliminating the need for the TI Peripheral Expansion Box and the RS-232 Interface Card. It plugs directly into the TI-99/4A, is fully compatible and provides daisy chain connection for TI peripherals. And our price is only $105, so you save up to $300 in the bargain!

Doryt also provides savings on additional 32K memory. Ours costs only $175 and you don't have to buy the Peripheral Expansion Box. Plus it plugs directly into the TI-99/4A, is fully compatible with all TI software and provides daisy chain connection for peripheral units.

As a further service, Doryt can provide an economical printer for the system.

Buy the Doryt 18A, 32K memory, printer, and our interconnecting cable and save an additional $50 on the total package! Call collect (516) 676-7950.

Doryt Systems, Inc.

14 Glen Street
Glen Cove, N.Y. 11542
(516) 676-7950

Dealer inquiries invited.

C. Regena

Playing Music On The TI

Musicians, rejoice! Here is a computer that plays music. You can play a single tone to tune an instrument or get a pitch for a song, or you can play complex rhythms with three-part harmony. Compose to your heart's content and let the computer perform your masterpiece.

Nonprofessional musicians – if you'd like to learn more about music, your computer can be a great teacher. Learn the notes on a keyboard, learn to read music, learn definitions, or learn to distinguish chords. The computer with music capabilities makes learning fun.

For those of you who prefer not to write your own programs, the TI *Music Maker* command module is available. Here's a quick review. You may compose music by choosing various types of notes or rests (quarter, eighth, half, etc.) and placing them on the staff. Choose notes for accompaniment if you wish. Build a song a measure at a time. The computer makes sure the timing works out correctly. Oh yes, you can choose your key signature, time signature, and tempo. At any time you can play or edit your composition, then save it on cassette or disk if you like.

Another section of the module is made especially for nonmusicians. You may draw lines up and down the screen at different levels for a "sound graph," then hear the computer play relational tones. Add second and third voices if you wish. This command module is really quite versatile with many options and can help you learn about music.

CALL SOUND

To program your own music on the TI, use the CALL SOUND statement. The basic form is

CALL SOUND(*duration, frequency, volume*)

The *duration* is a numeric expression (number, variable, or algebraic expression which will evaluate to a number) which is the number of milliseconds you wish to play the tone. For example, 1000 would be one second. The number may be from 1 to 4250 or from -4250 to -1.

The *frequency* is a numeric expression that indicates what tone to play. The frequency is the cycles per second and may be from 110 to 44733, which is from low A on the bass staff to out-of-human-hearing range. The "Musical Tone Frequencies" table in the Appendix of the *User's Reference Guide* lists the musical notes with the corresponding frequencies. Note that you can specify numbers that are between the normal musical tones.

The *volume* is a numeric expression that indicates loudness. The volume may vary from 0 to 30, where 0 is the loudest. The volume also depends on the audio setting of your monitor or television, but you can control relative volumes of the tones with this parameter.

Try this command:

```
CALL SOUND(500,440,2)
```

The computer plays the tone of A (440) for 500 milliseconds (half a second) at a volume level of 2.

Now, if you want to tune your band instrument, just run this program.

```
100 CALL SOUND(4250,440,0)
110 GOTO 100
```

You may specify one, two, or three notes to be played in one CALL SOUND statement. Each statement has one duration, then a frequency with a volume for each note desired. Here is an example of the three notes in the C major chord:

```
CALL SOUND(1000,262,6,330,4,392,2)
```

The chord will play for 1000 milliseconds. The notes played are C at a volume 6, E at a volume 4, and G at a volume 2. Try a few chords with different frequency and volume numbers.

If you play a solo instrument, you might enjoy programming the computer to play the accompaniment chords. Tune your instrument with the computer, then you can play with the computer as your accompanist.

Using Sheet Music

If you use three tones in the CALL SOUND statement, they may be in any order. I like to use the first frequency and volume as the melody tone, then the second and third frequencies and volumes as the accompaniment tones. This way I can keep track of which number is the melody. Also, if I start to run out of memory in a piece, I can go back to the CALL SOUND statements and delete accompaniment tones by keeping only the first frequency and volume in each statement.

You may work from a copy of written music to try out the musical capabilities of the TI. The top note is usually the melody. You may choose any two notes written directly under the melody note for the accompaniment or the other two notes in your CALL SOUND statement. To emphasize the melody, use a louder volume for the melody note and softer volumes for the accompaniment notes. For example:

```
CALL SOUND(400,262,1,196,6,159,8)
```

If you have two CALL SOUND statements together which specify the same frequencies and volumes, the notes may sound like one long note rather than two separate notes. To make the notes sound distinct, just change the volume number for one of the notes:

```
300 CALL SOUND(200,262,2,196,6,165,8)
310 CALL SOUND(200,262,3,196,6,165,8)
```

To make a bass note sound tied or held while two different melody notes are played, keep the frequency and the volume numbers the same in both statements:

```
500 CALL SOUND(300,262,2,165,8)
510 CALL SOUND(300,330,2,165,8)
```

Other statements may be executed while a note is being played. You may define graphics, draw graphics, or make calculations between CALL SOUND statements. This feature allows you to have fun choreographing pictures with music to present a musical dramatization. You do need to experiment so you don't get too many statements between the music statements or there will be gaps in the music.

A note will keep playing for its specified duration, and the computer will execute statements until either the duration runs out or another CALL SOUND statement is encountered. If another CALL SOUND statement needs to be executed, the computer waits until the first duration is finished before starting the next sound. If you prefer to have the computer go ahead with the next sound statement, use a negative number for the second statement's duration. Here is an example.

```
100 CALL SOUND(200,392,2)
110 CALL SOUND(200,330,2)
120 CALL SOUND(200,262,2)
130 CALL SOUND(200,330,2)
140 CALL SOUND(400,392,2)
150 END
```

The computer starts with the tone of G and plays for 200 milliseconds. Next the tone of E plays for 200 milliseconds, then C for 200 milliseconds, then E for 200 milliseconds, then G for 400 milliseconds. During the last note the program will end, but the note will keep playing for the 400 milliseconds.

Now change to negative durations in lines 110-140:

```
100 CALL SOUND(200,392,2)
110 CALL SOUND(-200,330,2)
120 CALL SOUND(-200,262,2)
130 CALL SOUND(-200,330,2)
140 CALL SOUND(-400,392,2)
150 END
```

This time, the computer starts by playing G. As soon as the computer comes to line 110, a CALL SOUND statement with a negative duration, the computer immediately starts the new sound – no matter what the previous duration was. Line 140 starts the sound of G as soon as the computer comes to that statement, then continues the sound for 400 milliseconds since there is not a following sound statement with a negative duration. Try running these two programs to hear the difference.

A technique I like to use in programming music is to use a variable name for the duration, and specify the numeric value of that duration variable near the beginning of the program. For example, I often use T for "tempo" or "time" or M for "metronome marking" or N for "note." If I use T to represent the duration for a quarter note, then T/2 would be an eighth note and 4*T would be a whole note. You can get exact timing in your music and let the computer calculate the durations.

Note: Avoiding using Q for "quarter note," especially on the TI-99/4, because the key combination of SHIFT Q is "quit." This is comparable to the FCTN (quitting on the TI-99/4A). An accidental SHIFT Q will wipe out your program and return to the title screen. With a shifted parenthesis before the variable and a shifted comma after the variable, it's too easy to get an accidental SHIFT Q.

Variable Durations

Another advantage to using a variable duration is that you can write your song in terms of the variable, then change the tempo of the song by changing only one line (the line defining the duration) rather than each CALL SOUND statement. Here is a short example.

```
100 T=400
110 CALL SOUND(T,262,2)
120 CALL SOUND(T,294,2)
```

```
130 CALL SOUND(2*T,330,2)
140 CALL SOUND(3*T/4,349,2)
150 CALL SOUND(T/4,392,2)
160 CALL SOUND(T/2,440,2)
170 CALL SOUND(T/2,494,2)
180 CALL SOUND(T*4,523,1)
190 END
```

100 Duration of quarter note = 400
110 Quarter note
120 Quarter note
130 Half note
140 Dotted eighth note
150 Sixteenth note
160 Eighth note
170 Eighth note
180 Whole note

RUN the program. Now change line 100 to T = 800. The song is twice as long, but each note stays in the exact proportion. Change line 100 to T = 200. The song is faster, but still in proportion.

If you need to learn a song with a difficult rhythm, program the computer to play the song. Use a variable such as T for the duration. You can set the duration to a slower note, then as you learn the song you can speed it up by changing just the one line.

You may prefer to use variables for the different kinds of notes in this manner:

```
100 T=400
110 E=T/2
120 H=T*2
130 CALL SOUND(H,523,2)
140 CALL SOUND(E,494,3)
150 CALL SOUND(E,440,3)
160 CALL SOUND(T,392,2)
```

100 Quarter note duration
110 Eighth note
120 Half note

You may also want to set up a list of variables for the note names before you use them in CALL SOUND statements:

```
100 T=400
110 C=262
120 D=294
130 E=330
140 CALL SOUND(T,E,2)
150 CALL SOUND(T,D,2)
160 CALL SOUND(T,C,2)
```

You may also use a variable for the volume, such as CALL SOUND(T,D,V).

Just as in other programming, you can use FOR-NEXT loops and GOSUB and GOTO statements to help write your music. For example, if you have a musical phrase between repeat bars, you can use a FOR-NEXT loop to play it twice. If you have a common phrase used several times within a song, use a GOSUB procedure.

Beethoven Medley

The following program, "Ludwig," illustrates the use of CALL SOUND statements to create a medley of familiar Beethoven pieces. Line 120 sets the duration of a quarter note to 400 milliseconds for the first tune, an excerpt from "Ode to Joy" of the Ninth Symphony. Lines 170-660 play this melody. In between the CALL SOUND statements are graphics statements. Lines 180-340 define graphics characters and colors, then later CALL HCHAR and CALL VCHAR statements draw a picture. The CALL SOUND statements in lines 170-400 illustrate the "tied" bass note, or a bass note held while two melody notes are played. Most of the notes are quarter notes, but line 610 has a dotted quarter note, line 650 has an eighth note, and line 660 has a half note.

Line 860 resets the duration variable T to 200 milliseconds. This time T represents an eighth note for phrases from "Ecossaises." The excerpt here is taken from music that is within repeat bars but has a first ending and a second ending. The common part of the repeat is in the subroutine at lines 1860-2230. Line 890 GOSUB 1860 plays the common phrase, then lines 920-980 play the first ending. Line 1010 repeats the common phrase with GOSUB 1860, then lines 1040-1100 contain the second ending.

Lines 1260-1420 play the third melody, "Für Elise." This example shows GOSUB commands within a FOR-NEXT loop. The subroutine for the common notes is contained in lines 2240-2420.

The final melody (lines 1430-1840) is an excerpt from the second movement of Beethoven's Fifth Symphony. Line 1430 defines the new duration T to be 800 milliseconds for an eighth note at an andante tempo. U is defined as three-fourths of an eighth note, or a dotted sixteenth note. T/4 is used for a thirty-second note. Character 128 is defined as a graphic musical note, and the embedded CALL HCHAR statements among the CALL SOUND statements place the notes on the screen.

Line 1850 (GOTO 1850) holds the picture on the screen. Press CLEAR (FCTN 4 on the TI-99/4A or SHIFT C on the TI-99/4) to stop the program.

If you prefer to save the typing time, you can obtain a copy of this program by sending $3, a stamped, self-addressed mailer, and a blank tape or disk to: REGENA, P.O. Box 1502, Cedar City, UT 84720. Please specify the name of the program.

Ludwig

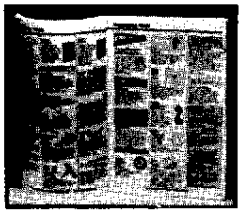
```
100 REM  BEETHOVEN MEDLEY
110 REM
120 T=400
130 CALL CLEAR
140 CALL SCREEN(3)
150 PRINT TAB(6);"BEETHOVEN MEDLEY"
    ;
160 CALL COLOR(1,2,8)
170 CALL SOUND(T,330,2,131,6)
180 CALL CHAR(96,"FFFFFFFFFFFFFFFF"
    )
```

```
190 CALL CHAR(97,"FF7F3F1F0F070301"
    )
200 CALL SOUND(T,330,3,131,6)
210 CALL CHAR(98,"FFFEFCF8F0E0C080")
220 CALL CHAR(104,"FFFFFFFFFFFFFFFF
    ")
230 CALL SOUND(T,349,3,131,6)
240 CALL CHAR(105,"0103070F1F3F7FFF
    ")
250 CALL CHAR(106,"80C0E0F0F8FCFEFF
    ")
260 CALL SOUND(T,392,2,131,6)
270 CALL CHAR(120,"00003C3E3E1E0F03
    ")
280 CALL CHAR(121,"003878F8F8F0C0BC
    ")
290 CALL SOUND(T,392,3,147,6)
300 CALL CHAR(113,"FF7F3F1F0F070301
    ")
310 CALL CHAR(114,"FFFEFCF8F0E0C080"
    )
320 CALL SOUND(T,349,3,147,6)
330 CALL COLOR(10,8,8)
340 CALL COLOR(11,3,11)
350 CALL SOUND(T,330,3,147,6)
360 CALL VCHAR(13,15,104,7)
370 CALL VCHAR(13,16,104,7)
380 CALL VCHAR(13,17,104,7)
390 CALL VCHAR(13,18,104,7)
400 CALL SOUND(T,294,3,147,6)
410 CALL HCHAR(19,14,105)
420 CALL HCHAR(19,19,106)
430 CALL HCHAR(20,13,105)
440 CALL HCHAR(20,14,104,7)
450 CALL HCHAR(20,21,106)
460 CALL SOUND(T,262,2,165,6)
470 CALL HCHAR(21,11,105)
480 CALL HCHAR(21,12,104,10)
490 CALL HCHAR(21,22,106)
500 CALL SOUND(T,262,3,165,7)
510 CALL HCHAR(22,9,105)
520 CALL HCHAR(22,10,104,14)
530 CALL HCHAR(22,24,106)
540 CALL SOUND(T,294,2,175,6)
550 CALL HCHAR(23,7,105)
560 CALL HCHAR(23,8,104,18)
570 CALL HCHAR(23,26,106)
580 CALL SOUND(T,330,2,176,5)
590 CALL CHAR(115,"80A2A2AAEEEFFFFF
    ")
600 CALL CHAR(99,"80A2A2AAEEEFFFFF"
    )
610 CALL SOUND(T*1.5,294,2,196,7)
620 CALL HCHAR(24,1,99,32)
630 CALL CHAR(122,"0F3F7F7D790101")
640 CALL CHAR(123,"FEFE9EC0E0E0E0E0"
    )
650 CALL SOUND(T/2,262,3,196,7)
660 CALL SOUND(2*T,262,4,165,7,131,
    8)
670 CALL COLOR(9,3,8)
680 CALL COLOR(10,11,8)
690 CALL HCHAR(24,6,115,22)
700 CALL HCHAR(12,15,113,2)
710 CALL HCHAR(12,17,114,2)
720 CALL HCHAR(11,13,97)
730 CALL HCHAR(11,14,96,6)
740 CALL HCHAR(11,20,98)
750 FOR I=10 TO 4 STEP -1
760 READ A,B
770 CALL HCHAR(I,A,97)
780 CALL HCHAR(I,A+1,96,B)
790 CALL HCHAR(I,A+B+1,98)
800 NEXT I
810 DATA 12,9,9,14,7,18,6,20,4,24,5
    ,26,1,28
820 CALL HCHAR(1,1,96,94)
830 CALL HCHAR(3,31,98)
840 CALL HCHAR(2,32,32)
850 CALL HCHAR(1,32,98)
860 T=200
870 CALL COLOR(2,16,3)
880 CALL COLOR(12,3,3)
890 GOSUB 1860
900 CALL HCHAR(3,6,120)
910 CALL HCHAR(3,7,121)
920 CALL SOUND(T,466,3,117,8)
930 CALL HCHAR(4,6,122)
940 CALL HCHAR(4,7,123)
950 CALL SOUND(T,831,4,698,8)
960 CALL HCHAR(5,10,120)
970 CALL HCHAR(5,11,121)
980 CALL SOUND(T*2,831,3,698,7,233,
    9)
990 CALL HCHAR(6,10,122)
1000 CALL HCHAR(6,11,123)
1010 GOSUB 1860
1020 CALL HCHAR(2,15,120)
1030 CALL HCHAR(2,16,121)
1040 CALL SOUND(T,349,3,294,7,117,9
     )
1050 CALL HCHAR(3,15,122)
1060 CALL HCHAR(3,16,123)
1070 CALL SOUND(T,466,3)
1080 CALL HCHAR(4,26,120)
1090 CALL HCHAR(4,27,121)
1100 CALL SOUND(T*2,466,2,294,6,233
     ,8)
1110 CALL HCHAR(5,26,122)
1120 CALL HCHAR(5,27,123)
1130 CALL COLOR(12,16,3)
1140 CALL HCHAR(6,20,120)
1150 CALL HCHAR(6,21,121)
1160 CALL HCHAR(7,20,122)
1170 CALL HCHAR(7,21,123)
1180 CALL HCHAR(8,16,120)
1190 CALL HCHAR(8,17,121)
1200 CALL HCHAR(9,16,122)
1210 CALL HCHAR(9,17,123)
1220 CALL HCHAR(2,22,120)
1230 CALL HCHAR(2,23,121)
1240 CALL HCHAR(3,22,122)
1250 CALL HCHAR(3,23,123)
1260 FOR I=1 TO 2
1270 GOSUB 2240
1280 CALL SOUND(T,415,3)
1290 CALL SOUND(T,494,3)
1300 CALL SOUND(T,523,2,110,15)
1310 CALL SOUND(T,165,4)
1320 CALL SOUND(T,220,4)
1330 CALL SOUND(T,330,3)
1340 GOSUB 2240
1350 CALL SOUND(T,523,3)
1360 CALL SOUND(T,494,4)
1370 CALL SOUND(T,440,4,110,14)
1380 CALL SOUND(T,165,10)
1390 CALL SOUND(T,220,7)
1400 NEXT I
1410 CALL SOUND(T,330,6)
1420 CALL SOUND(T*3,440,6)
```

```
1430 T=800
1440 U=T*3/4
1450 CALL CHAR(128,"080C0A0A0878F87
     ")
1460 CALL COLOR(13,2,6)
1470 CALL SOUND(1,9999,30)
1480 CALL SOUND(U,156,6)
1490 CALL COLOR(1,2,6)
1500 CALL COLOR(9,3,6)
1510 CALL COLOR(10,11,6)
1520 CALL SOUND(T/4,208,5)
1530 CALL SOUND(T,262,3)
1540 CALL HCHAR(17,4,128)
1550 CALL SOUND(U,262,4)
1560 CALL HCHAR(15,8,128)
1570 CALL SOUND(T/4,233,4)
1580 CALL SOUND(U,208,3)
1590 CALL HCHAR(13,12,128)
1600 CALL SOUND(T/4,262,4)
1610 CALL SOUND(T+U,175,3,139,10)
1620 CALL HCHAR(13,21,128)
1630 CALL SOUND(T/4,220,3)
1640 CALL SOUND(U,233,3)
1650 CALL HCHAR(15,25,128)
1660 CALL SOUND(T/4,262,2)
1670 CALL SOUND(U,277,2,233,8)
1680 CALL HCHAR(17,29,128)
1690 CALL SOUND(T/4,262,3)
1700 CALL SOUND(U,233,2,196,8)
1710 CALL SOUND(T/4,277,2)
1720 CALL SOUND(U,196,2,156,8)
1730 CALL SOUND(T/4,233,2)
1740 CALL SOUND(U,165,3,131,8)
1750 CALL SOUND(T/4,196,3)
1760 CALL SOUND(T+U,262,2)
1770 CALL SOUND(T/4,233,4)
1780 CALL SOUND(U,220,4,175,10)
1790 CALL SOUND(T/4,175,4)
1800 CALL SOUND(U,233,2,117,10)
1810 CALL SOUND(T/4,277,3)
1820 CALL SOUND(U,196,4,156,10)
1830 CALL SOUND(T/4,156,4)
1840 CALL SOUND(2*T,208,2)
1850 GOTO 1850
1860 CALL SOUND(T,392,3,156,8)
1870 CALL HCHAR(2,2,42)
1880 CALL SOUND(T,466,3)
1890 CALL HCHAR(4,27,42)
1900 CALL SOUND(2*T,466,2,233,6,196
     ,8)
1910 CALL HCHAR(6,14,42)
1920 CALL SOUND(T,523,3,392,6,156,8
     )
1930 CALL HCHAR(8,11,42)
1940 CALL SOUND(T,466,3)
1950 CALL HCHAR(2,26,42)
1960 CALL SOUND(T*2,466,2,392,6,196
     ,8)
1970 CALL HCHAR(3,4,42)
1980 CALL SOUND(T,622,1,392,6,156,8
     )
1990 CALL HCHAR(2,19,42)
2000 CALL SOUND(T,466,2)
2010 CALL HCHAR(7,23,42)
2020 CALL SOUND(T*2,466,1,392,5,196
     ,8)
2030 CALL HCHAR(3,12,42)
2040 CALL SOUND(T,523,1,392,5,156,8
     )
2050 CALL HCHAR(9,19,42)
2060 CALL SOUND(T,466,3)
2070 CALL HCHAR(6,7,42)
2080 CALL SOUND(T*2,466,2,392,5,196
     ,8)
2090 CALL HCHAR(5,24,42)
2100 CALL SOUND(T,349,1,294,5,117,8
     )
2110 CALL HCHAR(5,17,42)
2120 CALL SOUND(T,466,3)
2130 CALL HCHAR(2,9,42)
2140 CALL SOUND(T*2,466,2,294,6,175
     ,8)
2150 CALL HCHAR(4,20,42)
2160 CALL SOUND(T,392,2,311,5,117,8
     )
2170 CALL HCHAR(2,30,42)
2180 CALL SOUND(T,466,3)
2190 CALL SOUND(T*2,466,2,311,7,176
     ,8)
2200 CALL SOUND(T,415,3,349,6,117,8
     )
2210 CALL SOUND(T,466,4)
2220 CALL SOUND(T*2,466,3,349,6,208
     ,8)
2230 RETURN
2240 CALL SOUND(T,659,6)
2250 CALL SOUND(T,622,6)
2260 CALL SOUND(T,659,6)
2270 CALL SOUND(T,622,5)
2280 CALL SOUND(T,659,4)
2290 CALL SOUND(T,494,3)
2300 CALL SOUND(T,587,4)
2310 CALL SOUND(T,523,5)
2320 CALL SOUND(T,440,6,110,15)
2330 CALL SOUND(T,165,8)
2340 CALL SOUND(T,220,6)
2350 CALL SOUND(T,262,4)
2360 CALL SOUND(T,330,4)
2370 CALL SOUND(T,440,4)
2380 CALL SOUND(T,494,4,131,15)
2390 CALL SOUND(T,165,4)
2400 CALL SOUND(T,208,4)
2410 CALL SOUND(T,330,4)
2420 RETURN
2430 END
```


FREE CATALOG! 264

Features Precision Tools and Equipment for Computers



Page after page of hard-to-find precision tools, test equipment, computer accessories and equipment. Send for your free copy today!

Name _____

Company _____

Address _____

City _____

State _____ Zip

Jensen Tools Inc.
7815 S. 46th St.
Phoenix, AZ 85040

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari Retirement Planning

In Craig Cole's suggested improvements to the Atari version of Retirement Planner (COMPUTE!, April 1983) which appeared in the July "Readers' Feedback" section (p. 16), line 70 should have read:

```
70 Y=Y+1
```

TI Goblin

As this game from the July issue (p. 72) was presented, your goblin could not reach the faces in the rightmost column of the screen. This can be corrected by changing line 760 to read:

```
760 COL=COL+SGN(31-COL)
```

To make the game display the proper high score, delete lines 270, 280, and 290, and add the following line:

```
1125 IF S>HS THEN 1130 ELSE 1140
```

Thanks to Canadian reader Luc Cousineau and others who pointed out this problem.

Fortress Of Adnil

In the program for this Timex/Sinclair game (July 1983, p. 92), the GOTO statements were missing from the following two lines:

```
8040 IF INKEY$="" THEN GOTO 8020
8209 IF INKEY$="" THEN GOTO 8207
```

Roadblock

Readers who have had trouble typing in this long machine language program for the Atari may want to add the following lines to the BASIC loader (July 1983, p. 108) to help check for typing errors in their DATA statements:

```
15 LN=1010
20 FOR L=0 TO 16
25 FOR C=0 TO 69
30 D=PEEK(L*70+13824+C):IF D>150 T
   HEN 40
35 T=T+D:NEXT C
40 READ CK:IF CK<>T THEN PRINT "ERRO
   R IN LINES ";LN;"-";LN+90:STOP
45 PRINT "LINES ";LN;"-";LN+90;" OK"
50 T=0:LN=LN+100:NEXT L
55 A=USR(14788)
3000 DATA 3797,8372,7521,6620,6660,8
     073
3010 DATA 7319,7083,6853,6626,6631,8
     059
3020 DATA 7685,7702,7451,7720,7014
```

Circles

For the machine language circle-drawing routine presented in this Atari graphics article from the July issue to work properly, the following lines must be added to Program 7 (p. 168), the BASIC loader for the routine:

```
28004 RESTORE 29500
28005 FOR I=1577 TO 1584:READ A:POKE
      I,A:NEXT I
29500 DATA 128,64,32,16,8,4,2,1
```

Timex/Sinclair Screenscrolls

Reader Daniel Froats notes that the screen scrolling programs from the July issue (p. 216) will work with the basic 2K of memory if the following two lines are entered before running the program:

```
POKE 16389,n
CLS
```

Replace n with any number greater than 90.

Commodore 64 Video Tour

Jim Butterfield writes that readers whose 64s have the newer ROM sets may experience problems with the BASIC programs from Parts VI (July 1983 p. 218) and VII (August 1983, p. 182) of his series on the 64's video capabilities. This is because the ROM routines leave the raster interrupt in a non-standard state. To make the programs work, add the following line:

```
90 POKE 53265,27
```

VIC Bitmapping

Line 100 of Program 1 from this article on VIC high resolution graphics (July 1983, p. 248) should read:

```
100 POKEV+15,17*T-9:FORI=.TO255:POKEW+I,I
    :POKER+I,C-1:NEXT:FORI=.TOL*2*πSTEPB
```

The "shifted up-arrow" key combination on the VIC gives the symbol for pi, which acts as a constant with the value of pi when used in calculations.

First Math

A bug in the VIC, 64, and Apple versions of this educational game from the August issue (p. 92) sometimes causes a "division by zero" error when playing in the addition, subtraction, or multiplication modes. To prevent this, add the following line:

VIC or 64 Version (Programs 1 or 2)

```
123 IF A$<>CHR$(47) THEN 140
```

Apple Version (Program 5)

```
395 IF A$ <> ("/") THEN 430
```

©

Ck commodore
SUPER-MART
★★★★★★★★★★★★★★★
$$$ SAVE TIME & MONEY $$$
HANNA ENTERPRISES
1303 COLUMBIA DR. suite 207
Richardson, Texas 75081
OUTSIDE OF TEXAS CALL
1-800-527-1738
TO ORDER CALL (214)
231-2645
MasterCard & Visa accepted
add 3% surcharge for credit cards
F.O.B. Dallas, Texas
9:30 a.m. - 6:30 p.m. (m-f)
10:30 a.m. - 2:30 p.m. sat.

COMMODORE COMPUTERS

B-500-128k	**$777.20**
PET 64	**$627.50**
8032	**$627.50**
8096	**$876.25**
9000 Super Pet	**$1071.25**
c-64	**$233.00**
Executive 64 portable	**CALL**

COMMODORE DISK DRIVES

1541 (170k)	**$250.50**
2031	**$311.00**
4040 (340k)	**$657.85**
8050 (1mg)	**$981.90**
8250 (2mg)	**$1226.50**
9060 hard disk (5mg)	**$2040.00**
9090 hard disk (7.5mg)	**$2290.00**

COMMODORE PRINTERS

1525 (30cps)	**$238.75**
1526 (100cps)	**$343.95**
4023 CBM (100cps)	**$330.95**
8023 (160cps)	**$537.95**
6400 daisy wheel CBM	**$1417.75**
new! 1520 plotter printer	**$178.50**

COMMODORE PERIPHERALS

1701 color monitor	**$249.95**
c1600 modem	**$59.30**
c1650 automatic modem	**$94.50**
Datasette 1530	**$58.50**
CBM 64k memory board	**$246.95**
Super Pet upgrade board	**$488.95**
Cables PET-IEEE	**$34.95**
Cables IEEE-IEEE	**$42.95**

COMMODORE SOFTWARE FOR CMB

wordpro4+ or 5+	**$305.00**
Visicalc (Expanded)	**$195.00**
BPI G/L A/R A/P Inv. etc.	**$320 ea.**
MANAGER (database)	**$195.00**
c64 SOFTWARE:	
Easy script	**$99.95**
Easy mail	**$49.95**
Word/name machine	**$29.95**
Logo	**$99.95**
Pilot	**$99.95**
Music machine	**$29.95**
Music composer	**$29.95**

the rear of the VIC-20 cabinet.

The expansion board is available for $129.95.

Legend Valley Computer Systems
1474 Naughtingham
Newark, OH 43055

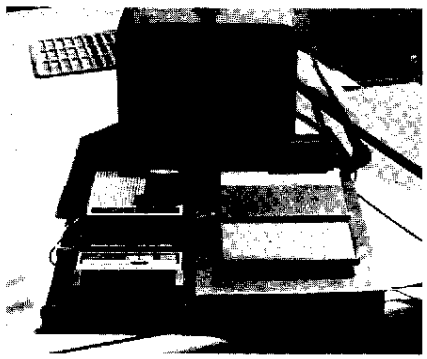
Traveling Timex

Car Comp is an accessory for the Timex/Sinclair computer that turns it into a traveling companion.

Car Comp is a rigid platform to which the computer and cassette recorder can be attached. It can be used anywhere with AC power or a 12-volt cigarette-lighter outlet.

Car Comp, which is produced by L&G Enterprises, sells for $59.95.

L&G Enterprises
Box 6854
Silver Spring, MD 20906



Car Comp secures a Timex/Sinclair and cassette deck as well as allowing power to be drawn from a standard auto cigarette-lighter outlet.

TI-99/4A Assortment

Western Properties Investment Company has produced a line of products designed for the TI-99/4A computer. The programs include a word processor, a data base, and a spreadsheet.

The word processor, *Printer Book*, is designed to handle up to two pages of text. Control of the printer is achieved through use of the CTRL key. The program can be used in conjunction with one of Western Properties' *File Book* programs to merge records from a data base with text. *Printer Book*'s 14 menu options include record merging, merge to screen or printer, and automatic multi-letter printing.

File-Book III handles up to 100 records with 6 items per record. The program includes full editing, search and sort capabilities as well as output to screen, printer or tape.

Income and Expense Spreadsheet IV is an accounting spreadsheet composed of 2 income and 50 expense categories. The program produces monthly charts of each of the 52 categories and an annual chart. Data is saved to tape with a cassette routine that is four times normal speed.

Printer Book and *File-Book III* are available for $39.95. *Income and Expense Spreadsheet* sells for $43.95. The programs run on Extended BASIC, but require no memory expansion. A printer is optional for the data base and spreadsheet programs.

Western Properties Investment Company
Software Division
P.O. Box 9602
Marina Del Rey, CA 90295

VCR Interface For VIC And 64

The Videobook Corporation has introduced Prometheus 1, an interface for the Commodore 64 and VIC-20 computers and home video cassette recorders.

Prometheus 1 allows you to produce interactive videotape courseware. It will connect a VIC-20 or Commodore 64 to various Panasonic, Magnavox, Canon, and Hitachi video cassette recorders.

The interface, which sells for $49.95, is being marketed in conjunction with Videobook's *Comp-U-Tutor Computer/VCR*

keted by Commodore for the Commodore 64 computer.

The program, which can handle home budgets, financial planning, small business accounting, educational projects, and statistics, will be available on disk for less than $100.

Commodore
Computer Systems Division
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

TI Joystick

The Prostick 2002 is a direct replacement joystick for the TI-99/4 and TI-99/4A computers.

The joystick, which sells for $29.95, requires no additional interface for connection with the TI. It includes a 4-way/8-way switchable gateplate that allows 8-way action to be disabled when playing games that are limited to vertical and horizontal movement.

The Prostick 2002 has two firing buttons located on the top end of the base, allowing both right- and left-handed play. The fire buttons are designed to be controlled by the index finger for faster response and decreased fatigue during play.

Newport Controls
15425 Los Gatos Boulevard
Los Gatos, CA 95030
(408) 358-3439

Data Base Manager For VIC And 64

Jini Micro Systems, creator of several data base managers for Commodore computers, has released *Mini Jini*, a record keeper for the VIC and 64.

The program is available in cartridge format, and files can be saved to either tape or disk. It will handle between 35 and 500 records, depending on available memory.

Mini Jini will accept up to 10 fields of information per record, can sort by any field, and can search by record number, name, or phrase. When used with a printer, the program can generate reports and mailing labels.

Mini Jini includes a math function to perform calculations on file data, and it can be used in conjunction with word processing programs to produce personalized letters and custom reports.

The program sells for $89.95. Data files with sample records set up for dozens of applications are available on disk for $14.95, or tape for $9.95.

Jini Micro Systems, Inc.
Box 274
Riverdale, NY 10463

TI-99/4A Cartridge Connection

Romox is producing the GamePort expansion module for the TI-99/4A. The module, which sells for $39.95, plugs into the computer's I/O port and accesses the 9900 CPU directly.

The GamePort is designed to circumvent the reported plan of Texas Instruments to modify its internal software to accept only cartridges programmed in TI's patented GROM format.

The GamePort module, which accepts both GROM and standard ROM cartridges, includes an 8K ROM, plus RAM memory.

Romox, Inc.
501 Vandell Way
Campbell, CA

Grade And Attendance Manager

GradeCalc is a grade and attendance management package designed to relieve teachers of time-consuming, record-keeping tasks.

The grade section can average grades using a variety of methods, including percentage

COMMODORE 64 SOFTWARE

THE MAILER: A list and label program ideal for mailing lists. Disk. **$30.00**

SUPER MAILER: A deluxe version. Adds sorts, searches, alphabetizing and more. Disk. **$50.00**

DATA-BYTES: A data-storage and retrieval program. Prints hard copies. Integrates with the mailer. Disk. **$30.00**

THE EDITOR: Integrates with mailer and data bytes programs. Add sorts, enhanced editing features and more. Disk. **$30.00**

SALES 64: Generates personalized sales orders, receipts, invoices & packing slips on economical plain paper. Tractor feed compatible. Disk. **$30.00**

TRS80 Model 100 version $30.00 (tape) Deluxe version $50.00 (tape)

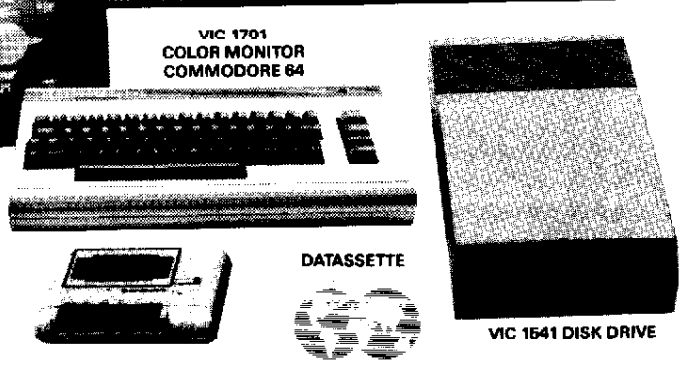
SUSIE SOFTWARE
709 Wilshire Drive, Mt. Prospect, IL 60056 (312) 394-5165

GENERIC GAMES
* * *
STAR THIEF
BY JAMES NITCHALS
(APPLE II DISK)
ONLY $9.95

COD ORDERS MUST PAY SHIPPING AND HANDLING. DEALER INQUIRIES INVITED. FOR ADDITIONAL INFO ON NEW RELEASES SEE THE READER INFORMATION CARDS IN THE BACK OF THIS MAGAZINE.
SEND CHECK OR MONEY ORDER TO:

N C D
P.O. Box 947
Solana Beach, CA 92075

SAVE



PICK YOUR BEST COMMODORE COMPUTER PACKAGE AND SAVE EVEN MORE...

C 64	$229.00		C 64	$229.00
VIC 1541	239.00	or	VIC 1530	65.00
VIC 1701	249.00		VIC 1701	249.00
Total	$700.00		Total	$530.00
SAVE	$ 17.00		SAVE	$ 13.00

Everything you need to support your COMMODORE VIC20 and C64

Cardco, Creative Software, EPYX, HES, UMI, WICO, Romox.

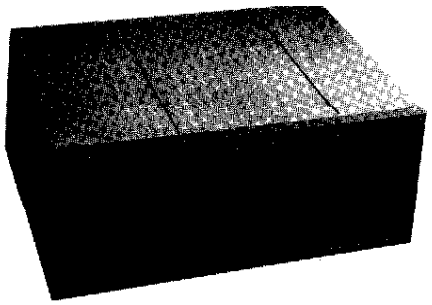
Micro Software International Inc

PractiCalc 20
For the Commodore VIC-20
16K RAM required

Tape (CV2T10)	$28.00
Disk (CV2D10)	33.00

PractiCalc 64-For the Commodore 64

Tape (C60T11)	$37.00
Disk (C60D11)	39.00



GET A T.I. PERIPHERAL EXPANSION BOX FREE

SAVE

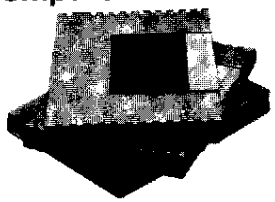
We will give you T.I.'s Peripheral Expansion Unit—a $249.98 retail value—FREE—when you buy any three of the below listed items—OR—when you buy $350.00 worth of in-stock software (no substitutions).

PHP 1220 RS 232 Card	$135.95
PHP 1240 Disk Contoller Card	189.95
PHP 1250 Expansion System Disk Drive	300.95
PHP 1260 Memory Expansion Card 32K	227.95
PHP 1270 P-Code Card	189.95
PHP 3111 TI Writer	75.95
PHP 3113 Microsoft Multiplan	75.95

GET A T.I. SPEECH SYNTHESIZER FREE

T.I. will give you a Speech Synthesizer—a $99.95 retail value—FREE—when you buy six solid state software cartridges. Offer expires January 31, 1984. Ask for Coupon.

We carry in stock all hardware and software for TI home computers.



QUANTITIES LIMITED!

TEXAS INSTRUMENTS

SAVE

1733 Marietta Blvd. N.W. Atlanta, Georgia 30318

Order Toll Free 1-800-241-2883
In Georgia (404)-351-8468

VIC & APPLE

DISCOUNT SOFTWARE (game, utility, educ., applications) for the VIC-20 and APPLE 11+ or APPLE 11e (1 disk, 48K). Write for a full catalogue — 30-day warranty on faulty merchandise — please pay in your country's currency. VIC on cassette only.

	U.S.A. - CAN.	U.K.
VIC-20		
SNAKMAN	$9.00	£4.00
CRICKET	9.00	4.00
KRAZY KONG	9.00	4.00
ESCAPE	9.00	4.00
APPLE		
CHOPLIFTER	14.00	8.50
CASTLE WOLFN.	14.00	8.50
SNEAKERS	12.00	7.50
RASTER BLASTER	14.00	8.50

DISCOUNT SOFTWARE
P.O. Box 1489
Niagara-on-the-Lake, Ont.,
Canada L0S 1J0

FREE FREE FREE for C64/V20*

One 3 piece set of the finest quality, hand made computer covers for the C64/V20*, printer and disk drive. **A $23.95 value!**...When you enter your one year subscription to **SOFTYPE**...Why spend hours of typing?...Avoid frustrations!!! Let **SOFTYPE** do it for you!...**for as little as $3.99**; we will send you one menu driven tape or diskette** a month with all the public domain programs for your C64/V20* printed in the monthly issues of this magazine. Order now!!!...before the Christmas rush.

- ☐ C64* 1 year subscription to **SOFTYPE**
- ☐ V20* (with FREE covers) $47.88
- ☐ Set of three covers... Only $23.95 (Texas add 6% sales tax)
- ☐ One month of **SOFTYPE**... $3.99

**Add $1.75 for disk version per issue

Send check or money order to:
SOFTYPE
1807 Cobble Creek
Houston, Texas 77073

*TRADEMARK OF COMMODORE BUSINESS MACHINES. YOU PAY ONLY FOR THE SERVICE AND TAPE. THE PROGRAMS ARE FREE.

MAKE YOUR OWN CARTRIDGES!

- Complete System • Programs 8K Cartridge to "AUTO-RUN" (opt) at Power-up — YOUR BASIC or Machine-Code Prgm.
- Prototype GAMES
- EDUCATIONAL use — NO Load Delay
- OS-3721 plugs into VIC-20
- Expansion socket on board
- Carts. Erasable & BLK Locatable

OS-3723 ZIF socket module Allows programming 2764, 2732A, 2732, 2716, 2564, 2532, 2516 EPROMS	$24.50
OS-37A System (Pgmr., 8K Cart)	$88.50
OS-37B System (Pgmr., ZIF mod) (Software Tape incl. Disk add $2)	$88.50
OS-3722 Blank 8K Carts.	$24.50
Blank 2764 EPROMS	$11.50

OTTO SYSTEMS
8135 ENGINEER ROAD
SAN DIEGO, CA 92111
(619) 569-5665

Add $4 Shipg. & Hndlg. — CA res. 6% tax

ATTENTION
TEXAS INSTRUMENTS TI-99/4A OWNERS

We have hundreds of 3rd party independent software programs on cassette and disk ready to run on your TI-99/4A. Games, business, and educational programming at discount prices as low as $8.95 ea. Plus all TI hardware and software at incredibly low, low prices, including the new TI-99/2 and CC-40 computers. We also have dust covers, heavy duty joysticks with TI adapters, and many more accessories. Call or write now for a FREE listing. We ship your order U.P.S. the same or next business day to insure fast service. Visa and MasterCard accepted (NO service charges) or C.O.D. is okay.

THE MUSIC WORKSHOP
59 E. Tioga St.
Tunkhannock, PA 18657
CALL 1-717-836-4522

KINDER KONCEPTS

30 PROGRAMS FOR KINDERGARTEN CHILDREN.
READING READINESS, MATH CONCEPTS, PERCEPTION, COLORS, SHAPES, ETC.

Disks $39.50 for 5 programs*
All 30 on 1 disk with Backup — $175.00

For COMMODORE PET® and 64 ™ and APPLE II +

We also carry "Master Grades", "Multiple Choice", "Football Scout" and other useful programs for teachers.

Please add $2.00 per disk for shipping & handling.

MIDWEST SOFTWARE
BOX 214 • FARMINGTON, MI 48024
Phone: 313-477-0897 (4:00 pm - 11:00 pm)

Please send for complete descriptions and ordering information.
* Sample disk with 2 programs for $10.00.
® and TM trademarks of Commodore Business Machines and Apple Computer Company.

GEMINI OR EPSON PRINTER CONFIGURATION PROGRAM

If you have trouble understanding all those software codes that configure your printer, then this is the program for you. Just turn on your printer, boot the program and choose from the menu more than a dozen print types. We also include a preview of other functions available on the Gemini/Epson printers. Program is available for Apple II+, Ataris and Vic-20. Specify computer type, memory size and either disk or tape when ordering. Send check or money order for $1.95(disk)$14.95(tape) to: DOWN AND OUT SOFTWARE, 33 Storm Drive, Holtsville,NY 11742.

APPLE·ATARI·COMMODORE·TI OWNERS
PROTECT YOUR INVESTMENT
CLEAR ANTI-STATIC VINYL

Dust Covers

Full One Year Guarantee

COVERS AVAILABLE FOR YOUR SPECIFIC MODELS
ATARI-COMMODORE-TI
COMPUTER AND ACCESSORY COVERS EACH $9.95
APPLE
COMPUTER AND ACCESSORY COVERS EACH $11.95
WE HAVE OVER 1,000 MAKES AND MODELS
PLEASE STATE MAKE AND MODEL WHEN ORDERING
ALSO AVAILABLE: ANTI-STATIC MATS 18"x24" $24.95
IDEAL FOR HOME OR OFFICE USE

Call or Write for Brochure
COM-COV
217/787-9372
113 EXETER COURT
SPRINGFIELD, IL 62704
ILLINOIS RESIDENTS ADD 5% SALES TAX
ALL ORDERS ADD $2 FOR SHIPPING AND HANDLING
PLEASE ALLOW 4-6 WEEKS FOR DELIVERY

WHERE DOES IT GO ?

This package makes short work of tracking 35 expenses and 7 incomes (which may be changed, deleted or added too). Suggestions and instructions for its use are provided.

Daily or weekly records may be stored and then entered on supplied monthly forms.

Household financial record keeping becomes easy and pleasant. (The results may surprise you.)

TI-99/4A Cassette Package $12.95
(Fla. Add 5% Tax).....+$ 1.50 Postage

Free Additional Information

I. S. A.
9808 N. W. 67th Court
Tamarac, Florida 33321

1/2 PRICE INTRODUCTION
Commodore & TI

Spreadsheet	**19.95**
Data Base	**19.95**
Word Processor	**9.95**
Payroll	**9.95**
Mailing List	**9.95**
Inventory	**9.95**
Games Pak*	**19.95**
Education Pak*	**19.95**
Business Pak*	**19.95**

(*Valu-paks have 5 programs)

VALORUM
441 Clyde Ave.
Mtn. View, CA 94043
(415) 968-8500