INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

DATE JANUARY/FEBRUARY, 1981 VOLUME NO. II, no. 1

TO BE A GOOD PROGRAMMER TODAY IS AS MUCH A PRIVILEGE AS IT WAS TO BE A LITERATE MAN IN THE SIXTEENTH CENTURY

Andri Ershov

Welcome to the second year of Interaction. The quote above probably expresses the reason a lot of you bought an Interact. As an introduction to computers. However Interaction is a users' newsletter, not a course in computer literacy. I can sympathize with those who are starting programming but the newsletter must try to cover all aspects of the Interact.

For those just starting, I recommend you read Crash Course In Microcomputers by Louis E. Frenzel, Jr. , Howard W. Sams & co. For those new to Basic programming, I recommend Basic Primer by M. Waite and M. Pardee , Howard W. Sams & Co. or Basically Speaking from MicroVideo. MicroVideo's new Basic manual is a good buy even if you already know Basic and have read the Interact owner's manual. It has lots of interesting information useful to anyone programming the Interact.

For those wanting to get into machine language, which includes me, I recommend SCELBI 8080 Software Gourmet Guide and Cookbook by Robert Findley , Scelbi Consulting, inc. and Z80 & 8080 Assembly Language Programming by Kathe Spracklen , Hayden .

None of these books, with the exception of Basically Speaking, are written specifically for the Interact and many things may differ from the way the Interact works but I've found the majority of information to be sound and usuable.

PRODUCT REVIEWS

JRA SYSTEMS RS232 interface & COMPRINT 912 printer
reviewed by Geoff Hall, San Jose, CA.

This is to report my experience of the above products, both supplied by Micro Video.

The interface installation instructions describe how to transfer the 8080 CPU from the main circuit board to the interface board. The latter is then supposed to be mounted in place by engaging the 40 pins protruding from its underside into the vacated 8080 socket. This proved to be easier said than done and I spent considerable time rocking the board around and producing interesting video patterns but not the proper "L FOR LOAD" message. Worse yet, I discovered all this struggling had cracked off several fragile wires connecting the keyboard. This caused me to add some unprintable expletives to the crazy video display! A week later, after making repairs, I was again futilely rocking the interface board back and forth. Finally, I "piggy-backed" the interface onto the 8080, by soldering the 40 wires, and then found it easy to install. Once this was done the interface and RS232 Basic interpreter worked fine and have proved trouble-free. No doubt the installation difficulties I encountered were at least partly due to my own clumsiness. I do, however, recommend the installation method I used.

The COMPRINT 912S printer is supplied by Micro Video with RS232 connector all ready to plug into the Interact. It is only necessary to set pins for the correct baud rate,etc., as detailed in the 64 page instruction manual which I found to be comprehensive and easily understandable. The manual also describes the software commands to control the printer. The printer comes with 1 roll of the special aluminized paper and I found this sufficient to keep me going while awaiting a further supply. There have been a number of magazine reviews of the COMPRINT 912, including onComputing,Fall 1980, so I will only add that it performs well with the Interact and RS232 interface. After months of hand-writing program listings, its a real pleasure to key in the RS232 Basic command LLIST and have the COMPRINT make a beautiful neat listing. During printing it is fast and quiet and the aluminized paper makes excellent Xerox copies.

Adding these products to my Interact has greatly expanded the range of possible applications, such as report-generating, computer art,etc.,and in my opinion is well worth the modest cost.

RS-232 PORTS : MICROVIDEO VS. SLAGH

by Stephen Cook

Now that I have used both ports for several months, I now feel I have sufficient information about both to make a comparison. My comparison will lean more towards an explanation of the Slagh unit as there is another review of the MicroVideo unit also in this issue. The software used with the two ports also determines the capabilities of both ports.

The Slagh port I purchased was a kit. The complete construction took under 2 hours (bear in mind that I am an audio service technician by trade and solder PC boards daily). The only difficulty encountered was making sure jumpers went in the right holes. The second unit I assembled (for a friend) had a better pictorial to use for jumper placement.

Installation of the Slagh and Microvideo ports is similar. Both require removal of the 8080 CPU chip and installation of a free standing port board with long wire wrap socket pins into the 8080 socket and reinstallation of the 8080 into the port board. The Slagh instructions warns about breaking the keyboard cable which did on one unit. I then replaced the entire cable with a better quality stranded cable (it required less than 1 foot of 15 conducter ribbon cable). The Slagh instructions also recommend removing the main pc board to ease installation of the RS232 board. The MicroVideo unit's cable ends in a female DB25 socket which is glued with a plastic block to the cabinet. The Slagh unit's cable ends with a male DB25 plug with about 18 inches of free cable outside the cabinet.

Electronically, the two RS232 ports are very different. The MicroVideo unit uses an 8250 asynchronous communication IC. The 8250 IC features a dual port, however no documentation is supplied on how to implement the second port. The user must obtain an 8250 data sheet from Intel and then trace the board to determine connections. No schematic is supplied, though all cable connections are given in charts showing printer and modem hookups.

The Slagh unit uses a 6850 asynchronous communication IC. A schematic is included in the documentation. The schematic and a hookup table clearly show which function is connected to each connector pin. Information is given in both documentations on how to set baud rate, parity, and other parameters.

The MicroVideo port requires the use of RS232 Basic for printer use. The main difference between this Basic and Level II basic is the addition of LPRINT and LLIST commands and the fact that the token values for Basic commands and functions are different. What this means is that Level II or Graphics Basic programs will not RUN or LIST correctly in RS232 Basic. If you have a Level II program and you want a listing, you must first use RS232 Ezedit and the TRANSLATE command to change that program to RS232 Basic. The only addition in RS232 Ezedit is the TRANSLATE command. Of course you can write a program in RS232 Basic. But if you want to give a copy to someone else, they must also have a MicroVideo port as RS232 programs will not RUN in Level II, even if they don't use LPRINT commands, because of the

RS-232 PORTS, cont.

different token values. RS232 Ezedit does not have a de-translate function and if a Level II version is required an entire program must be retyped. The MicroVideo port does drive my printer well, though not above 4800 baud (to be expected, too fast!) and not below 300 baud. I have had trouble getting it to run at 110 baud which is the standard teletype speed. The MicroVideo port supplies RS232-c standard signals and not the 20 ma. current loop that is required for a teletype so the 110 baud rate is not important anyway.

The Slagh port does run at 110 baud and also supplies the 20 ma. current needed for a teletype as well as the RS232 standard signals. Why the interest in teletypes? I don't own one but they are the best bargain in computer printers on the market. They may be noisy and slow but good used machines can be bought at Ham radio swap meets for only $150. Reconditioned teletypes cost only $200-300. My own printer is a COCOCL 48 column plain paper printer. It bought it specifically for newsletter listings as it uses only 3-7/8 inch or smaller adding machine paper and it features multiple type sizes and has full handshaking and baud rate capabilities. It is based on the same printer mechanism as the LRC and Atari printers.

Printer use of the Slagh port requires Level II Basic and an overlay tape to change Basic. Mark Slagh and others have found many printer functions already in Level II basic and the overlay tape adds full printer capabilities to Level II. The first printer overlay replaces DUMMY with a PORTS command so that a command A$=PORTS(A$,n) initializes the port and n determines the baud rate and output to screen or printer. By incorporating the PORTS command in a program to turn on and off the output to the printer, the Slagh port becomes as versatile as RS232 Basic and its separate PRINT and LPRINT commands. Also because more is known to owners about Level II Basic, we can also change the line length. This is something I have not figured out for RS232 Basic. This is important to me and other owners who have less than an 80 column printer. When printing a line with the MicroVideo port and RS232 Basic, my printer loses everything past the 48th character (the printer's maximum line length) because a carriage return- line feed is needed to end one line and start the next. The problem becomes even worse when I use a larger type size such as the 32 character size in the newsletter listings.

The one problem with the Slagh port and printer was that when running to the printer only (no screen output) without handshaking, such as with a teletype, the computer got ahead of the printer; sometimes the first character of a line would be printed as the head was returning from the right hand end of the previous line. The new Slagh printer overlay corrects this slight problem with a change in commands to PORT a,b,c,d where a is the baud rate factor, b determines the word format (length, parity, stop bits), c is the printer line length, and d is the number of nulls to send at the end of a line, allowing a teletype time to do a carriage return-line feed.

RS-232 PORTS, cont.

I have used both ports as a terminal with my Novation modem. The main difference is the software used. MicroVideo's port uses the Communicator by Long Playing Software. This is an excellent program which does backspacing, allows changing control characters, and can be altered directly to change word and baud parameters. The Slagh terminal program does work but parameters must be changed through Basic, it does not erase a backspaced character, and only escape (CHR$27) is available. Slagh is coming out with a new terminal program and if there is an improvement I'll pass that information along when I get the new program.

I hope this analysis has proved sufficient. Both ports work in all aspects. There is a price difference but I don't consider that significant unless you're skilled enough to assemble the board yourself. There is also the question of future compatability with MicroVideo's memory expansion (which is still unfinished as far as I can determine. I have no direct contact with MicroVideo.) The Slagh port is memory address relocatable and should be able to be rewired and work with any memory expansion unit.

INTERACTION ON TAPE

Due to lack of time, I can no longer produce Interaction Basic Programs on Tape. Because of the unexpected demand for ready to run programs, however, I have found someone else to undertake this job. If you would like a data cassette with all the Basic programs in a particular issue (1980 or 1981), specify issue and year (each issue sent on an individual cassette) and send $5.00 to

George Leggett 52895 Bunker Hill Blvd. New Baltimore, MI 48047
Please make check payable to George Leggett

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

SUBMITTING A PROGRAM

If you have a program you would like to submit for publication, please contact me first and I'll send you a data cassette to record the program and send back to me. This will save me typing in a program you've already done. Program instruction (and articles) do not necessarily have to be typed but if not write legibly.

ADDITION PRACTICE

by Terry O'Brien 6500 Lauge Rd Birch Run, MI 48415

This is an addition practice program for children (preschool thru kindergarten) using the subroutine by Bob Draganski ("Big Letters") from Interaction #5

I would like to have put something a little more artistic on the screen for a "reward" for correct answers but I'm not much of an artist. When an incorrect answer is given the correct solution is shown and then the program waits for any to be pressed (this is not stated on the screen).

The listing is from a Base 2 printer using Slagh Systems Services U80 port. I had some problems caused by my poor solder work but Mark Slagh was very helpful in correcting the problems and I am very pleased with the operation of the port. The Base 2 printer is a very good piece of equipment and I recommend it to Interact owners.

```
10 PRINTCHR$(8)
20 CLS:WINDOW18:OUTPUT"ADDITION",30,55,3
30 OUTPUT"PRACTICE",30,40,3
40 FORI=1TO500:NEXT Pause
50 INPUT"HOW MANY PROBLEMS";N
60 N(1)=(N(2)+N(3)):TONE110,124
70 F=1:O=40:Z=0:C=10:CLS
80 IFN(1)=NGOTO100
90 GOTO150
100 CLS:OUTPUT"CORRECT ",20,55,3:OUTPUTN(2),70,55,3
110 OUTPUT" WRONG  ",20,40,3:OUTPUTN(3),70,40,3
120 INPUT"TRY AGAIN ";0$
130 IF0$="Y"OR0$="YES"THENN(2)=0:N(3)=0:GOTO50
140 WINDOW 77,END
150 D=-50:K=0:T=0
160 IFT=3THEN0=0:D=D+40
170 IFT>6THENGOTO60
180 T=T+1
190 ONTGOTO230,200,230,210,610,220
200 A$(T)=" +":GOTO250
210 A$(T)=" ÷":GOTO250
220 PRINT%%%  - ?
230 A=(RND(1)*10):B=INT(A):A$(T)=STR$(B)
240 IFT>6GOTO520
250 D=D+30:K=K+6
```

ADDITION PRACTICE, cont.

```
260 IFT>5AND2>0THEN2=0
270 OUTPUTA$(T),2,K,1
280 FORY=KTO(K-5)STEP-1
290 FORX=CTO(C-5)STEP-1
300 IFPOINT(X,Y)=1THEN GOSUB400
310 NEXT:NEXT
320 IFT=9GOTO340
330 IFF>9THENC=15:T=9:GOTO280
340 IFE=FGOTO450
350 IFF>1ANDE=0GOTO520
360 IFE>0GOTO520
370 Q=Q-30
380 IFT=9GOTO60
390 GOTO160
400 REM BIG LETTERS
410 XB=5*X+C
420 YB=5*Y+Q
430 OUTPUTCHR$(1),XB,YB,3
440 RETURN
450 PRINTCHR$(7);CHR$(8);CHR$(7);CHR$(8);CHR$(7)
460 TONE 124,100
470 CLS:OUTPUT"CORRECT !",30,50,3
480 R=R+1
490 E=0
500 IFR>5THENE=0:N(2)=N(2)+1:R=0:GOTO60
510 GOTO450
520 TONE300,100 wRong tone
530 CLS
540 OUTPUT"ANSWER",40,65,3
550 M=0
560 OUTPUTA$(1),25,50,3:OUTPUT" + ",35,50,3:OUTPUTA$(3),45,50,3
570 M=VAL(A$(1))+VAL(A$(3))
580 OUTPUT" = ",55,50,3:OUTPUTM,65,50,3


590 G$=INSTR$(1)
600 E=0:N(3)=N(3)+1:GOTO60
610 INPUT" ANSWER PLEASE ?";A$(T)
620 E=VAL(A$(T))
630 Z=5          ↖3
640 F=VAL(A$(1))+VAL(A$(3))
650 I$=STR$(D)
660 GOTO250
Ok
```

FREE ADVERTISING

If you have something us-able on or with an Interact that you want sell or trade, you can advertise free if you're a subscriber. Do you have extra or excess tapes, a program that is good enough to sell but not good enough to sell to MicroVideo, or accesories or documentation you've done yourself? I'll give you 5 free lines (including your name and address) in an issue of Interaction if space is available. The total free ads will be limited to one page maximum per issue. All ads must be submitted typed, ready to run, on a 8½ x 11 sheet of white paper.

REVERSE
ƎƧЯƎVƎЯ
REVERSE

by Stephen Cook

Can you change a jumbled list of nine numbers into their proper numerical order? In less than 20 moves? In less than 15 moves? You get the right order by reversing part of the list.

For example, if you start with 376295841

and reverse 3, you get 673295841

```
LIST
1 REM <<REVERSE>>
2 REM INTERACT LEVEL II VERSION
3 REM BY STEPHEN COOK - 8/80
5 CLEAR:CLS
10 PRINT"    REVERSE"
20 PRINT:PRINT"    ESREVER"
30 PRINT:PRINT"    REVERSE"
40 PRINT:PRINT:PRINT
130 DIMA(20)
150 N=9
160 PRINT"DO YOU WANT THE  RULES
 (Y-N) ?"
170 A$=INSTR$(1)
180 IFA$="N"THEN210
185 CLS
190 GOSUB710
210 A(1)=INT((N-1)*RND(1)+2)
220 FORK=2TON
230 A(K)=INT(N*RND(1)+1)
240 FORJ=1TOK-1
250 IFA(K)=A(J)THEN230
260 NEXTJ:NEXTK
280 WINDOW24
290 PRINT:PRINT"HERE WE GO...":P
RINT"THE LIST IS"
310 T=0
320 GOSUB610
330 PRINT"HOW MANY SHALL I REVER
SE?"
340 M$=INSTR$(1)
345 R=ASC(M$)-48
350 IFR=0THEN520
360 IFR<=NTHEN390
370 PRINT"OOPS! TOO MANY!  I CAN
 REVERSE AT MOST";N:GOTO330
390 T=T+1
410 FORK=1TOINT(R/2)
420 Z=A(K)
430 A(K)=A(R-K+1)
440 A(R-K+1)=Z
450 NEXTK
460 GOSUB610
480 FORK=1TON
490 IFA(K)<>KTHEN330
500 NEXTK
510 PRINT"YOU GOT IT IN";T:PRINT
"MOVES!!!"
520 FORP=1TO1250:NEXTP:PRINT
530 PRINT"TRY AGAIN (Y-N) ?"
540 A$=INSTR$(1)
550 IFA$="Y"THEN210
560 PRINT:PRINT"HOPE YOU HAD FUN
.":GOTO999
610 CLS:PRINT:FORK=1TON
620 X=9*K
630 OUTPUTA(K),X,50,2
640 NEXTK
650 PRINT:RETURN
710 PRINT"TO WIN REVERSE,  ALL Y
OU HAVE TO  DO IS ARRANGE A"
720 PRINT"LIST OF NUMBERS  (1 TO
 9) IN      NUMERICAL ORDER"
730 PRINT"FROM LEFT TO     RIGHT
."
735 PRINT:FORP=1TO1250:NEXTP
740 PRINT"TO MOVE,YOU TELL ME HO
W MANY      NUMBERS (COUNTING"
750 PRINT"FROM LEFT TO     RIGHT
) TO REVERSE."
850 FORP=1TO1250:NEXTP:CLS:RETUR
N
999 WINDOW77:END
OK
```

* * *

LIFE PROGRAM for 8 or 16k Interact. Input through left joystick. 1 - 2 seconds per generation. Satisfaction guaranteed. Send $6. to

R. Schnapp 8062 Gold Coast Drive San Diego, CA 92126

PHONE NUMBER WORDS

by Stephen Cook

This program will print out a list of all the letter combinations of any phone number inputted. Of course 1 and Ø have no letter equivalent so they remain numerals. The program as written can be used with either port for printer output or display on the screen . Use screen command for Slagh port. The output for a printer will print only one combination on a line, so be prepared to waste a lot of paper or you will have to reformat the print statements.

```
LIST
1 REM "PHNDS"
2 REM PHONE NUMBER WORDS
3 REM FROM G.YOB  'PET' CREATIVE
   COMPUTING - 9/80
4 REM   INTERACT  VERSION BY  S.
   COOK  9/80
10 CLS
15 PRINT"DO YOU WANT"
20 PRINT"PRINTER (P)"
25 PRINT"OR"
30 PRINT"SCREEN (S) ?"
40 O$=INSTR$(1)
80 DIMD$(9),D(9),P(7),W$(7)
90 FORJ=0TO9:READD$(J):D(J)=LEN(
D$(J)):NEXT
100 DATA"0Z","1","ABC","DEF","GH
I","JKL","MNO","PRS","TUV","WXY"
110 INPUT"YOUR PHONE NUMBER";P$
120 REM GENERATE A STRING
130 IFLEN(P$)<>8THEN110
140 IFMID$(P$,4,1)<>"-"THEN110
150 PRINTP$:PRINT
160 P$=LEFT$(P$,3)+RIGHT$(P$,4)
170 FORJ=1TO7
180 P(J)=ASC(MID$(P$,J,1))-48
190 NEXTJ
200 PRINTP$
210 REM MAKE A WORD
220 FORA=1TOD(P(1))
230 W$(1)=MID$(D$(P(1)),A,1)
240 FORB=1TOD(P(2))
250 W$(2)=MID$(D$(P(2)),B,1)
260 FORC=1TOD(P(3))
270 W$(3)=MID$(D$(P(3)),C,1)
280 FORD=1TOD(P(4))
290 W$(4)=MID$(D$(P(4)),D,1)
300 FORE=1TOD(P(5))
310 W$(5)=MID$(D$(P(5)),E,1)
320 FORF=1TOD(P(6))
330 W$(6)=MID$(D$(P(6)),F,1)
340 FORG=1TOD(P(7))
350 W$(7)=MID$(D$(P(7)),G,1)
360 W$="":FORJ=1TO7:W$=W$+W$(J):
NEXT
370 IFO$="S"THENPRINTW$
375 IFO$="P"THENLPRINTW$
380 REM
390 NEXTG,F,E,D,C,B,A
Ok
```

MEMORY TEST

by Jim Dinkey JIRO Enterprises 3380 Cork Oak Way Palo Alto, CA 94303

This is a memory test program I wrote when I couldn't find anything else that told me what I needed to know: where the problem was. To execute it, all you do is load in the MicroVideo Monitor (or HILO Monitor), then this program and branch to its beginning. As long as everything is OK, the pattern on the screen will be continually changing. If there is an error, the program halts with the offending information available in memory. The information may be read out by reloading the Monitor. If you don't want to punch it in yourself, I have cassettes for $4.95 .

MEMORY TEST, cont.

```
LINE  ADDR  B1 B2 B3 B4          8080/8085 ASSEMBLER VER 1.0MR

   1                      %          ORG 7E30H
   2                      %          NAME ROB
   3                      %          ASEG
   4  7E30  F3            %          DI              INHIBIT ALL Z-80 INTERRUPTS
   5  7E31  2A 62 7E      %  PASS    LHLD FROML          INITIALIZE 'FROM'
   6  7E34  22 66 7E      %          SHLD CURRL        TO 'CURR'
   7  7E37  3A 65 7E      %  MORE    LDA  TO          IS 'CURR' ADDR = 'TO' ADDR?
   8  7E3A  2A 66 7E      %          LHLD CURRL        OBTAIN HIGH-ORDER POSITION
   9  7E3D  BC            %          CMP  H              NO BRANCH IF H-O SAME.
  10  7E3E  C2 4F 7E      %          JNZ  TEST        NOT FINISHED WITH LOOP.
  11  7E41  3A 64 7E      %          LDA  TOL         CHECK SECOND BYTE OF
  12  7E44  BD            %          CMP  L             ADDRESS.
  13  7E45  C2 4F 7E      %          JNZ  TEST        FAILS - DO ANOTHER BYTE.
  14  7E48  21 68 7E      %  EQUAL   LXI  H,DATA       'TO' = 'CURR' - QUIT.
  15  7E4B  34            %          INR  M           END OF PASS--INCREMENT DATA
  16  7E4C  C3 31 7E      %          JMP  PASS          AND RECYCLE AT NEXT MEMORY SECTION.
  17                      %  ;-----------------------------------------------------
  18  7E4F  2A 66 7E      %  TEST    LHLD CURRL       HL RECEIVES CURRENT ADDRESS.
  19  7E52  3A 68 7E      %          LDA  DATA        'DATA' TO ACCUMULATOR.
  20  7E55  77            %          MOV  M,A         STORE 'DATA' INTO MEMORY.
  21  7E56  BE            %          CMP  M           DOES THE BYTE CHECK OUT OK?
  22  7E57  C2 61 7E      %          JNZ  HALT          BAD LOCATION.
  23                      %  ;-----------------------------------------------------
  24  7E5A  23            %          INX  H           INCREMENT 'CURR' POINTER.
  25  7E5B  22 66 7E      %          SHLD CURRL       SAVE THE ADDRESS.
  26  7E5E  C3 37 7E      %          JMP  MORE        DO IT ALL OVER AGAIN.
  27                      %  ;-----------------------------------------------------
  28  7E61  76            %  HALT    HLT              FAILED TEST - LOOK IN LOCATIONS BELOW.
  29                      %  ;=====================================================
  30  7E62  00            %  FROML   DB   00H           ;START LOCATION - SET TO 'LAST'
  31  7E63  00            %  FROM    DB   00H
  32  7E64  03            %  TOL     DB   03H           ;ENDING LOC - USUALLY HIGH MEMORY
  33  7E65  00            %  TO      DB   00H
  34  7E66  00            %  CURRL   DB   00H           ;LOCATION OF FAILURE
  35  7E67  00            %  CURR    DB   00H
  36  7E68  55            %  DATA    DB   55H           ;THE BIT PATTERN THAT FAILED
  37  7E69                %          END
```

PRACTICAL PROGRAMS

Many of the respondents to the questionaire in the last issue requested more practical programs. Unfortunately, that is a problem for me as I bought an Interact for personal enjoyment and playing games. In fact I can think of very few applications that I could not do just as easily with a calculator or a set of index cards, especially when you consider the time to load Basic, then load a program, and finally input or load data. So this is a request for programs with practical uses for the Interact. If you have something that might be of general use, please consider submitting it for publication.

HAMMURABI

by Bob Draganski 14301 Harrison Livonia, MI 48154

An economic simulation game. By careful management you can see how to starve the least number of people by balancing the amount of grain you feed them and plant for next year. Or you can try to increase your wealth in land and grain without starving so many people that you're kicked out of office. Poor harvests, rats, disasters and plague combine to make whatever goal you pick difficult. This program shows some of the special tricks that can be done with an Interact to keep a text game interesting.

```
LIST
1 REM HAMMURABI
2 REM BY B. DRAGANSKI
5 CLEAR(150)
10 CLS:COLOR3,1,4,0:Y=50:POKE192
15,25:A$="HAMMURABI"
11 Z=0
13 OUTPUT"HIT CR",36,50,1:OUTPUT
"HIT CR",36,50,0:IFPEEK(14336)=2
51THEN15
14 Q=RND(1):GOTO13
15 RESTORE
20 FORX=2TO5:OUTPUT"H A M M U R
A B I",X,Y,3:Y=Y-1:NEXT
21 OUTPUT"H A M M U R A B I",X,Y
,1:GOSUB55
25 DATA168,33,124,45,97,58,80,14
0,97,58,80,250
26 FORE=1TO6:READA,B:TONEA,B:NEX
T
40 GOSUB50:GOTO59
50 POKE19215,25:CLS:POKE24777,4:
PRINT" ":POKE24777,235:RETURN
55 FORI=1TO2000:NEXT:RETURN
59 POKE6144,32
60 PRINT"TRY YOUR HAND AT":PRINT
"GOVERNING ANCIENT":PRINTTAB(5);
"SUMERIA":PRINT
70 PRINT"FOR A TEN-YEAR":PRINT"T
ERM IN OFFICE":GOSUB55
94 D1=0:P1=0
95 P=95:S=2800:H=3000:E=H-S:Y=3:
A=H/Y:I=5:Q=1
210 POKE6144,8:D=0
215 GOSUB50:PRINTA$;": I BEG":PR
INT"TO REPORT THAT":Z=Z+1:PRINT"
IN YEAR";1980+Z
217 PRINTD;"MEN STARVED":PRINTI;
"CAME IN CITY":P=P+I:IFQ>0THEN22
9
228 ONINT(5*RND(1))+1GOSUB600,61
0,620,630,640:GOSUB1500
229 OUTPUT"HIT CR",36,11,1
230 I$=INSTR$(1):GOSUB50:PRINT"P
OPULATION:":PRINTP;"PEOPLE":PRIN
T"CITY NOW OWNS"
231 PRINTA;"ACRES"
```

HAMMURABI, cont.

```
235 PRINT"YOU HARVESTED:":PRINTY
;"BUSHELS/ACRE":PRINT"RATS ATE:"
:PRINTE;"BUSHELS"
237 OUTPUT"HIT CR",36,11,1
240 I$=INSTR$(1):GOSUB50:POKE614
4,1:PRINT"YOU NOW HAVE:":PRINTS;
"BUSHELS"
241 IFZ=11THEN860
310 C=INT(10*RND(1)):Y=C+17
311 I$="LAND TRADING AT---"+STR$
(Y)+"--BUSHELS/ACRE"
312 SOUND3,156:FORN=1TO20:OUTPUT
MID$(I$,N,17),6,56,1
313 OUTPUTMID$(I$,N,17),6,56,0:N
EXT:OUTPUTMID$(I$,N-1,17),6,56,1
314 PRINT:PRINT:SOUND7,4096
320 Q=0:PRINT"HOW MANY ACRES":PR
INT"DO YOU WISH TO":INPUT"BUY";Q
:IFQ<0THEN850
321 IFY*Q<=STHEN330
323 GOSUB50:GOSUB720:CLS:GOTO311
330 IFQ=0THEN340
331 A=A+Q:S=S-Y*Q:C=0:GOTO400
340 Q=0:INPUT"SELL";Q:IFQ<0THEN8
50
342 IFQ<ATHEN350
343 GOSUB50:GOSUB720:CLS:GOTO311
350 A=A-Q:S=S+Y*Q:C=0
400 GOSUB50:PRINT"HOW MANY BUSHE
LS":PRINT"DO YOU WISH TO":PRINT"
FEED YOUR PEOPLE"
411 INPUTQ:IFQ<0THEN850
420 IFQ<=STHEN425
421 GOSUB710:GOTO400
425 S=S-Q:C=1
430 PRINT"HOW MANY ACRES":PRINT"
DO YOU WISH TO":PRINT"PLANT WITH
 SEED":INPUTD
441 IFD=0THEN511
442 IFD<0THEN850
445 IFD<=ATHEN450
446 GOSUB50:GOSUB720:GOTO430
450 IFINT(D/2)<=STHEN455
452 GOSUB50:GOSUB710:GOTO430
455 IFD<=10*PTHEN510
460 GOSUB50:PRINT"YOU ONLY HAVE"
:PRINTP;"PEOPLE TO":PRINT"TEND T
HE FIELDS":GOSUB55
470 GOTO430
510 S=S-INT(D/2)
511 GOSUB800
515 Y=C:H=D*Y:E=0:GOSUB800:IFINT
(C/2)<>C/2THEN530
525 E=INT(S/C)
530 S=S-E+H:GOSUB800:I=INT(C*(20
*A+S)/P/100+1):C=INT(Q/20)
531 Q=INT(10*(2*RND(1)-.3)):IFP<
CTHEN210
534 POKE6144,0
552 D=P-C:IFD>.45*PTHEN560
553 P1=((Z-1)*P1+D*100/P)/Z:P=C:
D1=D1+D:GOTO215
560 GOSUB50:PRINT"YOU STARVED":P
RINTD;"PEOPLE IN":PRINT"ONE YEAR
!!"
561 PRINT"DUE TO THIS MIS-":PRIN
T"MANAGEMENT YOU":PRINT"HAVE BEE
N IMP-":PRINT"PEACHED."
562 PRINT"YOU ALSO HAVE":PRINT"B
EEN DECLARED":PRINT"NATIONAL FIN
K!":GOTO990
600 D$="TERRORISTS STRUCK! KILLI
NG "+STR$(INT(P/6)):P=P-INT(P/6)
:RETURN
610 D$=" ROCK-SLIDE..."+STR$(INT
(P/5))+" PEOPLE REPORTED DEAD"
611 P=P-INT(P/5):RETURN
620 D$=" SUMERIA DAM BROKE!! DEA
TH TOLL "+STR$(INT(P/4))+" PEOPL
E"
621 P=P-INT(P/4):RETURN
630 D$=" TORNADO HIT DOWNTOWN SU
MERIA! KILLING "+STR$(INT(P/3))
631 P=P-INT(P/3):RETURN
640 P=INT(P/2):D$="HORRIBLE PLAU
GE STRUCK!! HALF THE PEOPLE DIED
":RETURN
710 PRINTA$;" YOU":PRINT"ONLY HA
VE";S:PRINT"BUSHELS OF GRAIN":GO
SUB55:GOSUB50:RETURN
720 PRINTA$;" THINK":PRINT"AGAIN
 YOU ONLY":PRINT"OWN";A;"ACRES":
GOSUB55:RETURN
800 C=INT(RND(1)*5)+1:RETURN
850 PRINTA$;" I CAN":PRINT"NOT D
O AS YOU":PRINT"WISH I QUIT":GOT
0990
860 GOSUB50:PRINT"IN YOUR TEN-YE
AR":PRINT"TERM IN OFFICE":PRINTP
1;"%";" OF THE"
862 PRINT"POPULATION STAR-":PRIN
T"VED PER YEAR ON":PRINT"THE AVE
RAGE,I.E."
863 PRINT"A TOTAL OF";D1
```

HAMMURABI, cont.

```
864 PRINT"PEOPLE DIED":L=A/P:I$=
INSTR$(1):GOSUB50:PRINT"YOU STAR
TED WITH"
866 PRINT"10 ACRES PER":PRINT"PE
RSON AND ENDED":PRINT"WITH";L:PR
INT"ACRES PER PERSON"
870 I$=INSTR$(1):GOSUB50
880 IFP1>33THEN561
885 IFL<7THEN561
890 IFP1>10ORL<9THEN940
895 IFP1>3ORL<10THEN960
900 PRINT"A FANTASTIC PER-":PRIN
T"FORMANCE!!!":PRINT"NOT EVEN I
COULD"
902 PRINT"HAVE DONE ANY":PRINT"B
ETTER":GOTO990
940 PRINT"YOUR HEAVY-HAND-":PRIN
T"ED PERFORMANCE":PRINT"SMACKS O
F NERO"
941 PRINT"AND IVAN IV.":PRINT"TH
E REMAINING":PRINT"PEOPLE FIND Y
OU":PRINT"AN UNPLEASANT"
942 PRINT"RULER AND FRANK-":PRIN
T"LY HATE YOUR GUTS":GOTO990
960 PRINT"YOUR PERFORMANCE":PRIN
T"WASN'T TO BAD":PRINT"BUT YOU C
OULD":PRINT"HAVE DONE A"
962 PRINT"WHOLE LOT BETTER":PRIN
TINT(P*.8*RND(1));"PEOPLE":PRINT
"WOULD LIKE TO"
964 PRINT"SEE YOU ASSASS-":PRINT
"INATED BUT":I$=INSTR$(1):GOSUB5
0
965 PRINT"WE ALL HAVE OUR"
966 PRINT"LITTLE PROBLEMS"
990 FORI=1TO10:TONE30,300:NEXT
1000 PRINT"AGAIN(Y/N):":I$=INSTR
$(1):IFI$="Y"THEN10
1100 STOP
1500 I$="*** BULLETIN ***":FORN=
1TO5:OUTPUTI$,6,30,2:TONE40,400:
1510 OUTPUTI$,6,30,0:NEXT
1515 SOUND3,109
1521 D$=D$+"                    "
1530 L=LEN(D$):FORN=1TOL
1531 I$=RIGHT$(I$,16)+MID$(D$,N,
1)
1540 OUTPUTI$,6,30,2:OUTPUTI$,6,
30,0:NEXT
1545 SOUND7,4096
1550 D$="  ":I$="":RETURN
2000 FORI=1TO10:Q=INT(10*(2*RND(
1)-.3)):PRINTQ:NEXT
OK
```

* * *

**** ELIZA - THE ROBOT DOCTOR ****

This well-known BASIC game actually simulates a conversation between a patient at the keyboard and the robot doctor. Not a copy of existing programs of the same type (e.g. ELIZA and DOCTOR), it employs similar string algorithms for keyword recognition and conjugation as these, but has been optimized for Interact BASIC Level II. The program formulates seemingly intelligent replies and questions in response to statements typed in by the human. Often humorous and probably cheaper and just as effective as a real doctor. Requires 16K machine. For cassette, listing and program description, send $10.00 to J.A. Miller, P.O. Box 455 Melbourne, Florida, 32901.

COMPONENT LOCATION DRAWING - Far superior to the one in the Interact maintainence package. It is 20" by 30" and is available for $9.95

WEST COAST INTERACT SERVICE - A policy of not over $75.00 in labor (plus parts) to fix an Interact "hard case" with most fees less. Write for shipping instructions.

both available from :
JIRO ENTERPRISES 3380 Cork Oak Way Palo Alto, CA 94303

A BASIC NOTE —

USING RELATIONALS IN ALGEBRAIC EXPRESSIONS

by Harry Holloway PO Box 2263 Ann Arbor, MI 48106

One of the challenges of using the Interact is getting a substantial Basic program into the meager 4.7K of space that is available. By now many users will have found the two most important ingredients of program compression. First, eliminate all unnecessary spaces and, second, put as many statements as possible on each numbered line (at the cost of being unable to use Ezedit). This note describes a third trick that is useful, but virtually undocumented.

Interact Basic treats a relational expression, such as (X< Y), like a variable that has the values -1 if true and Ø if false. The relationals may be used along with regular algebraic expressions in assignment statements (i.e., A=B or LET A=B). This feature can be used to write fairly compact assignment statements that will replace sequences of conditional statements. For example, suppose we need to set

```
X=3 if Y=5
X=10 if Y=7
X=35 if Y= anything else
```

The obvious way is to write

```
1ØØ   X=35
11Ø   IF Y=5 THEN X=3
12Ø   IF Y=7 THEN X=1Ø
```

We could shorten this a bit by combining lines 1ØØ and 11Ø with a colon between the two statements, but the third statement on line 12Ø will have to be on a separate line because otherwise it would only execute when Y=5 . Now, instead, we try using relationals in an algebraic expression and get

```
1ØØ   X=35+25*(Y=7)+32*(Y=5)
```

Comparing the space requirements of these versions we find that, after eliminating all unnecessary spaces, the first version requires 36 bytes. After lines 1ØØ and 11Ø are combined, this requirement is reduced to 32 bytes. However, the single line relational form only needs 27 bytes and the actual saving will probably be greater because we can add another statement or two to line 1ØØ .

As an example that occurs often, suppose we have a pair of displacements, X and Y, from which we need to calculate an angle (TH) using the ATN function. The problem is that ATN always returns an angle in the range $\pm \pi/2$ and we need to set the angle in the correct quadrant. The solution is to define somewhere

```
PI = 4 * ATN (1)
```

and then calculate

```
TH=ATN(Y/X)-PI*(X<Ø)+2*PI*(X>Ø)*(Y<Ø)
```

Try doing this by the more obvious method and you'll see what I mean.

EGGHEAD

by John Worrall Box 87 Avonmore, Ontario, CANADA K0C 1C0

A variation of mastermind. The only problem we've found in an otherwise self-prompting program is a tendency of people to assume that the clues (R for a right letter and C for a right letter in the correct order) are given by the machine in an order matching the players' guess. I.E., a guess of ABCD which gets you the results of CR is assumed to mean that the A is definitely in the first position and the B is the other right letter. This is not necessarily true. Possible, but not likely. Nitpickers might like to add a paragraph in the instructions to try and clarify this point but it seems to take more than it's worth. Enjoy!

```
LIST
10 CLS:OUTPUT"EGGHEAD--",5,20,1
20 PRINT;"A GUESSING GAME"
30 PRINT:PRINT:FORI=1TO1000:NEXT
I
40 CLS:CLEAR100:PRINT;"PLEASE TY
PE IN   YOUR NAME.":INPUT0$:CLS
50 PRINT;"THANKS":PRINT0$:PRINT;
"PLEASE PICK A   NUMBER FROM 1-
100":INPUTZ:Y=RND(-Z)
60 CLS:PRINT;"INSTRUCTIONS? Y/N"
70 A$=INSTR$(1)
80 IFA$="Y"GOTO820
90 CLS:RESTORE:FORK=1TO10
100 DATAA,B,C,D,E,F,G,H,I,J
110 READE$(K):NEXTK
120 H$=" ":I$=" ":P$=" ":R$=" ":
S$=" ":T$=" ":U$=" ":V$=" "
130 A$=E$(INT(RND(1)*10)+1)
140 B$=E$(INT(RND(1)*10)+1)
150 C$=E$(INT(RND(1)*10)+1)
160 D$=E$(INT(RND(1)*10)+1)
170 IFA$=B$THEN130
180 IFA$=C$THEN130
190 IFA$=D$THEN130
200 IFB$=C$THEN130
210 IFB$=D$THEN130
220 IFC$=D$THEN130
230 Z$=A$+B$+C$+D$
240 X=0
250 H1$="___":I1$="___":J1$="___
":P1$="___":R1$="___"
260 S1$="___":T1$="___":U1$="___
":V1$="___"
270 FORN=1TO10
280 PRINT"GUESS NUMBER ";N
290 G$=INSTR$(4)
300 OUTPUTG$,81,11,1
310 PRINT
320 PRINT"RESULTS: "
330 K$=MID$(G$,1,1):L$=MID$(G$,2
,1):M$=MID$(G$,3,1)
340 O$=MID$(G$,4,1)
350 IFASC(K$)>74THEN790
360 IFASC(L$)>74THEN790
370 IFASC(M$)>74THEN790
380 IFASC(O$)>74THEN790
400 IFK$=A$THENGOSUB1190
410 IFL$=B$THENGOSUB1190
420 IFO$=D$THENGOSUB1190
430 IFK$=B$THENGOSUB1170
440 IFK$=C$THENGOSUB1170
450 IFK$=D$THENGOSUB1170
460 IFL$=A$THENGOSUB1170
470 IFL$=C$THENGOSUB1170
480 IFL$=D$THENGOSUB1170
490 IFM$=A$THENGOSUB1170
500 IFM$=B$THENGOSUB1170
510 IFM$=D$THENGOSUB1170
520 IFO$=A$THENGOSUB1170
530 IFO$=B$THENGOSUB1170
540 IFO$=C$THENGOSUB1170
550 IFR=0THENPRINT;"TOTALLY WRON
G!!":A1$="WRONG":FORI=1TO500:NEX
TI:GOTO1020
560 IFR<>0THENPRINT:R=0
570 IFG$=Z$THEN730
580 IFN<9THENINPUT"PLEASE TYPE
IN   YOUR RESULTS";A1$:GOTO1020
590 IFN=10THEN600
600 CLS
610 NEXTN:PRINT;"TSK TSK"
620 FORJ=1TO3
630 SOUND0,3400
640 FORI=1TO55
650 SOUND0,1
```

EGGHEAD, cont.

```
660 NEXTI
670 NEXTJ
680 PRINT;"YOU BLEW IT":PRINT;Q$
690 PRINT
700 PRINT;"THE CODE WAS":PRINT;Z
$
710 PRINT
720 GOTO960
730 IFN<6THENPRINT;"WOW! YOU'RE"
:PRINT;"SMARTER THAN I   THOUGHT
."
740 IFN=6THENPRINT;"QUITE GOOD,"
:PRINT;"ACTUALLY (COULD  BE BETT
ER)":PRINT;Q$
750 IFN=7THENPRINT;"ONLY FAIR":P
RINT;Q$
760 IFN=8THENPRINT;"SUURELY YOU
CAN  DO BETTER THAN THAT!"
770 IFN>8THENPRINT;"THAT'S TERRI
BLE! MY PET ROCK CAN  DO BETTER
THAN THAT!"
780 GOTO960
790 PRINT:PRINT;"REMEMBER, THE"P
RINT;"LETTERS ARE A TO J ONLY."
800 FORI=1TO700:NEXTI
810 CLS:GOTO280
820 CLS:PRINT;"HI! "Q$:PRINT:PRI
NT;"EGGHEAD IS A ":PRINT;"GUESSI
NG GAME.":PRINT:PRINT
830 FORI=1TO700:NEXTI
840 CLS:PRINT;"I HAVE CHOSEN 4
DIFFERENT LETTERSOF THE ALPHABET
"
845 PRINT;"FROM A TO J."
850 FORI=1TO2000:NEXTI
860 CLS:PRINT:PRINT;"YOU MUST GU
ESS   WHICH 4 LETTERS"
870 PRINT;"I HAVE CHOSEN...":FOR
I=1TO900:NEXTI
880 PRINT:PRINT;"AND PLACE THEM
  IN THEIR CORRECT ORDER."
890 PRINT:PRINT;"HIT ANY KEY TO
  CONTINUE.":D$=INSTR$(1)
900 CLS:PRINT;"YOU HAVE 10 TRIES
TO GUESS THE FOURLETTERS AND THE
IR"
910 PRINT;"CORRECT POSITIONS"
920 PRINT:PRINT;"HIT ANY KEY TO
  CONTINUE":D$=INSTR$(1)
930 CLS:PRINT;"R= CORRECT LETTER
":PRINT:PRINT;"C= CORRECT LETTER
AND CORRECT ORDER"
940 PRINT:PRINT;"HIT ANY KEY TO
  CONTINUE.":D$=INSTR$(1)
950 GOTO90
960 PRINT:PRINT;"WOULD YOU LIKE
TOHAVE ANOTHER GO  AT IT? Y/N"
970 D$=INSTR$(1)
980 IFD$="Y"THEN90
990 CLS:PRINT;"ALRIGHT":PRINT;Q$
:PRINT:PRINT;"ANYTIME YOU":PRINT
;"CHANGE YOUR MIND,"
1000 PRINT;"I'LL BE WAITING."
1010 END
1020 X=X+1
1030 IFX=1THENH$=G$:H1$=A1$
1040 IFX=2THENI$=G$:I1$=A1$
1050 IFX=3THENJ$=G$:J1$=A1$
1060 IFX=4THENP$=G$:P1$=A1$
1070 IFX=5THENR$=G$:R1$=A1$
1080 IFX=6THENS$=G$:S1$=A1$
1090 IFX=7THENT$=G$:T1$=A1$
1100 IFX=8THENU$=G$:U1$=A1$
1110 IFX=9THENV$=G$:V1$=A1$
1120 CLS:PRINT;H$:OUTPUTH1$,60,1
7,2:PRINT;I$:OUTPUTI1$,60,17,2
1130 PRINT;J$:OUTPUTJ1$,60,17,2:
PRINT;P$:OUTPUTP1$,60,17,2:PRINT
;R$:OUTPUTR1$,60,17,2
1140 PRINT;S$:OUTPUTS1$,60,17,2:
PRINT;T$:OUTPUTT1$,60,17,2:PRINT
;U$:OUTPUTU1$,60,17,2
1150 PRINT;V$:OUTPUTV1$,60,17,2
1160 GOTO610
1170 PRINT"R":SOUND3,1334:FORI=1
TO50:NEXTI:SOUND7,4096:R=1
1180 RETURN
1190 PRINT"C":R=1:SOUND0,32:FORI
=1TO50:NEXTI:SOUND7,4096
1200 RETURN
Ok
```

DEADLINE

I must apologize for this issue being a little late. A touch of the flu kept me from meeting my deadline. The next issue should be on time, that is the end of March. If you have any ads or articles for the March-April issue, they must be in my hands by March 7 for consideration for publication.

COMPUTERFEST '81

In January, the Detroit Interact Group became a part of the Midwest Affiliation of Computer Clubs (MACC). One of the MACC's main annual functions is a computer show. Computerfest '81 will be held May 29, 30, and 31 at Franklin University in Columbus, Ohio. It will feature speakers, workshops, seminars, manufacturers' exhibits, door prizes, and an Interact users' group meeting, as well as other users' meetings. I would like to make that weekend an Interact owners get-together. We will have a Detroit Interact Group exhibit and I hopefully will be sponsoring an Interact party.
For more information on Computerfest '81 or to preregister ($3.50 , $4.00 at the door), send a self addressed stamped envelope to
COMPUTERFEST '81 PO Box 20205 Columbus, OH 43220

REPAIR HINTS

My Interact started to not load programs and Level II, even after cleaning the head. Below the READ-WRITE buttons is a visible double pole, double throw switch. A strip of index card wet with head cleaning fluid cleaned the contacts (gently..) and then a dry strip. Now it works again.

from Wayne Glass Franklin, WI

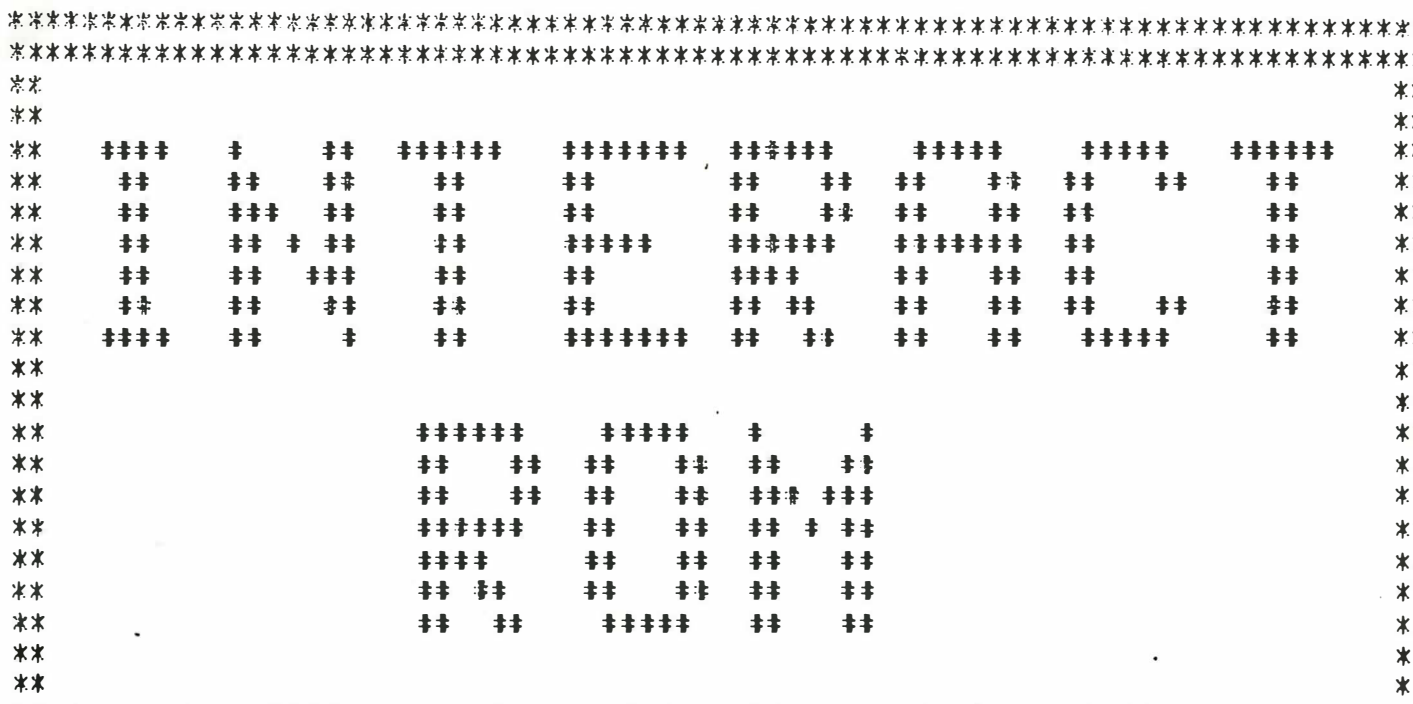
* * *

8080 DISASSEMBLER in BASIC - This BASIC program lists addresses, contents, corresponding ASCII character, and 8080 assembler language op codes and registers for any memory locations. For a data tape and complete instructions including a sample form for listing the output, returned by first class mail, send $5.00 to
David J. Schwab 10 Jay Lee Court Ann Arbor, MI 48104

UNO CARD GAME based on the card game. One person plays against the computer. This program is very challenging to your skill at strategy combined with luck. Runs in Basic

COLOR BAR PROGRAM This program is used to align your color or b&w TV. It produces fourteen different test patterns. They include a vertical horizontal line intersection, dots, cross hatch, corner boxes (for yoke alignment), vertical lines, horizontal lines, and color bars amoung others. This program is written in machine language for the highest speed in graphics possible. Just load the tape and you are ready - no language needed.
Send $5.00 for each individual program or send $8.00 for both the Color Bar and the Uno programs together to:

George A. Leggett 52895 Bunker Hill Blvd. New Baltimore, MI 48047



HERE IT IS! A ROM-BASED MONITOR FOR THE INTERACT! THIS ROM PLUGS INTO THE SPARE SOCKET INSIDE YOUR INTERACT AND PROVIDES FEATURES NOT AVAILABLE IN ANY CASSETTE-LOADED MONITOR. THIS MONITOR IS ALWAYS READY ON POWER-UP OR RESET-NO NEED TO LOAD FROM TAPE FOR EACH USEAGE. ONCE THIS ROM IS INSTALLED THE FOLLOWING COMMANDS ARE AVAILABLE TO THE USER-

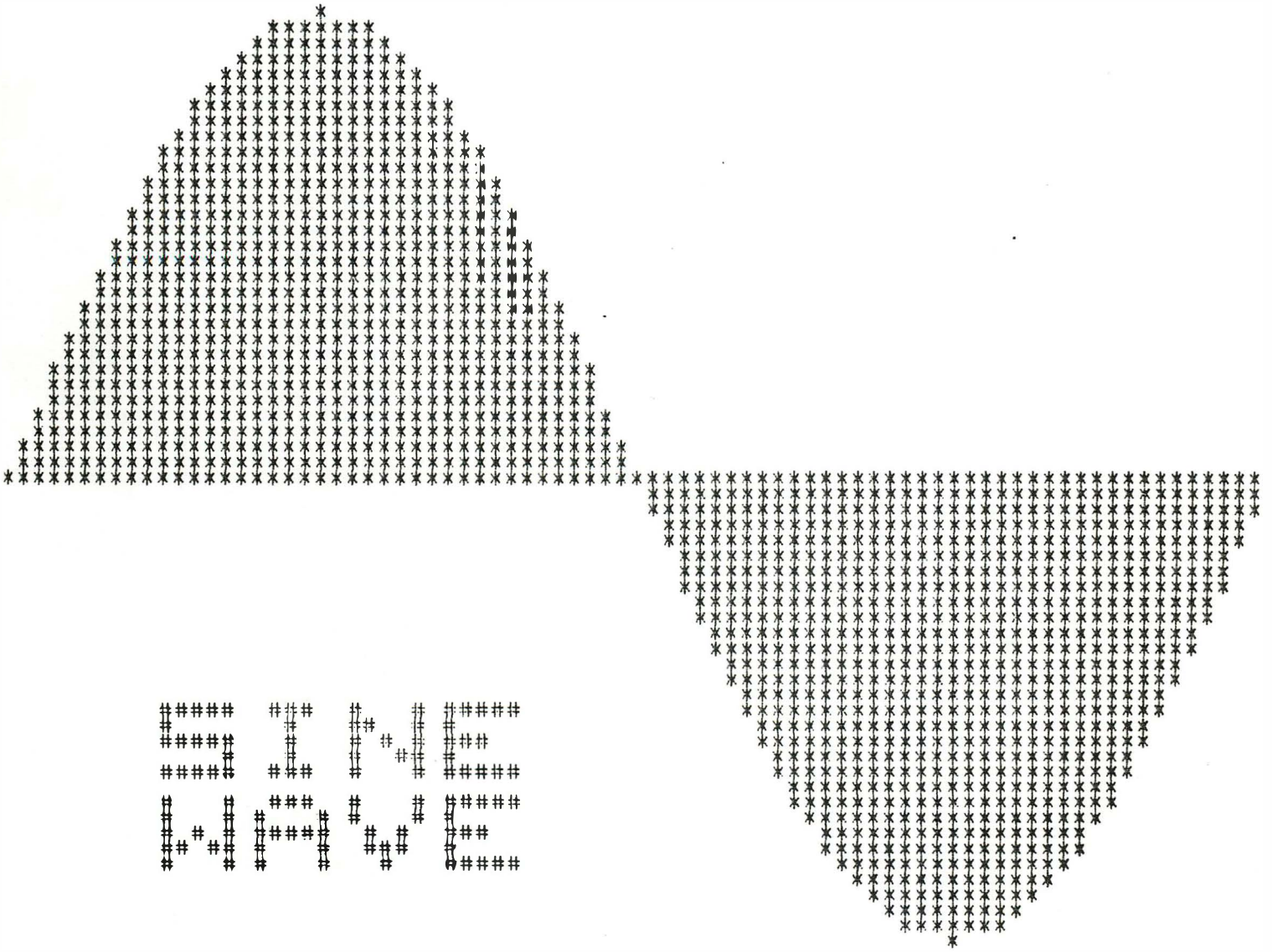
Inspect = inspect any memory location and change any ram data
Move = move any number of bytes foward or back in ram.
Dump = display blocks of memory data in HEX format.
Ascii = display blocks of memory data in HEX and ASCII.
Fill = fill any number of memory locations with user data
Search = search any memory block for 1 or 2 byte data words
Compare = compare two memory blocks and list all differences
Partial = set up a partial screen display for graphic work.
Go = execute user programs with full breakpoint control
Hex = adds two hex numbers of 1-4 digits each.
Rewind = turns on cassette unit to allow tape positioning.
Exit = exits rom monitor-returns control to INTERACT ROM
User = jumps to ram location 4C00H for additional user commands or to restart taped programs.
Load = loads cassette tape programs into ram memory.
Write = writes a block of memory locations to the cassette tape. Can be used to provide BACK-UP tapes of important programs or save mach. language programs.
Xreg = display and change any of the 8080s registers.

ORDER TODAY!! ROM MONITOR $IROM-1 at $34.95 each postpaid from W. HENDRICKSON 2313 W 181 st-Torrance CA. 90504.

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

MARCH - APRIL, 1981 VOL. II, NO. 2



THIS PICTURE WAS PRINTED BY AN EPSON MX-80

PRINT THE SCREEN

by Geoff Hall 1628 Trona Way San Jose, CA 95125

After creating a video picture have you wished you could make a permanent copy? Well, the RS232 interface + RS232 Basic + this short program makes your wish a reality.

PRINT THE SCREEN CONT.

The program PEEKs each 4 pixel bar of the screen, determines the color of each pixel and prints a corresponding character in a format representing the screen. A complication arises since the COMPRINT printer I own prints only 80 characters wide while the screen width is 112 pixels. So PRINT OPTION 1 which prints in the same orientation as the screen can only reproduce the left hand 80 pixels of each screen line. Alternatively, PRINT OPTION 2 prints the full picture by rotating it 90 degrees. In this way the 78 pixels picture height fits nicely into the printers 80 columns.

The program sections for the print options can be used as sub - routines within any RS232 Basic program and provide the capability of printing any desired picture displayed on the screen. So far I have only succeeded in printing pictures made by a RS232 Basic program, such as the demonstration listing which draws a sine wave. A preferable way to create pictures is by using COMPUTE-A-COLOR but I haven't yet discovered how to combine the tape recorded picture with the RS232 Basic program. I would appreciate hearing from anyone who knows how this may be done.

Now, if I only had high-resolution graphics for my Interact.....

(Ed. note - If you are not using a COMPRINT printer be sure to remove the control codes (CHR$) in lines 100, 280, and 420. If you have a Slagh port be sure to use PRINT not LPRINT.)

```
100 CLEAR500:POKE19215,25:LPRINT CHR$(30):CLS
110 PRINT"PRINT OPTIONS ARE":PRINT:PRINT"0=NO PRINT":PRINT"1=80 COLUMNS"
120 PRINT"2=112 COLUMNS":PRINT:INPUT"OPTION";A:CLS
130 FORX=0TO79:R=.07954*X:B=30*SIN(R):S=1:IFX>39THENS=-1
140 FORY=35TO(35+B)STEPS:PLOTX,Y,3:NEXTY,X:OUTPUT"SINE",10,20,1
150 OUTPUT"WAVE",10,14,1:ON A GOTO200,300
160 FORX=1TO9000:NEXT:GOTO100
200 REM 80 SCREEN COLUMNS X FULL HEIGHT
210 FORR=0TO77:T=16384+32*R
220 FORC=0TO19:D=PEEK(T+C)
230 FORX=1TO4:I=INT(D/4):A=D-4*I
240 IFA=0THENA$=" ":GOTO270
250 IFA=1THENA$=CHR$(35):GOTO270
260 A$="*":IFA=2THEN A$=CHR$(36)
270 D=I:LPRINT A$;
280 NEXTX:NEXTC:NEXTR:LPRINT CHR$(28):LPRINT CHR$(7):GOTO100
300 REM FULL SCREEN
310 FORC=27TO0STEP-1
320 FORX=1TO4:T=16384+C:LPRINT TAB(2);
330 FORR=0TO77:D=PEEK(T+32*R)
340 I=INT(D/64):P=D-64*I:D=P:IFX=1GOTO380
350 I=INT(D/16):P=D-16*I:D=P:IFX=2GOTO380
360 I=INT(D/4):P=D-4*I:IFX=3GOTO380
370 I=P
380 IFI=0THENA$=" ":GOTO410
390 IFI=1THENA$=CHR$(35):GOTO410
400 A$="*":IFI=2THENA$=CHR$(36)
410 LPRINT A$;
420 NEXTR:NEXTX:NEXTC:LPRINT CHR$(28):LPRINT CHR$(7):GOTO100
```

SHOOTING STARS

by Ken Stuempges 6261 N. Joyce Milwaukee, WI 53224

This game is based on one published in Interface Age magazine, April, 1977. It was authored by H. DeMonstoy. The version I wrote for the Interact has been substantially "amplified", however.

The game is played in a "universe" of stars(*) and black holes (0) that are arranged in a 3 by 3 matrix. Each position has its own number from one to nine. Position 1, 2, 3 are across the top, with 4, 5, 6 through the center and 7, 8, 9 across the bottom. The first universe has a star in the center (postion 5) and black holes all around it. The idea is to shoot stars only, never a black hole, and change the "universe" into eight stars surrounding one black hole.

Each star has its own "galaxy", and when a star is hit, every position in that galaxy will change, all stars will become black holes and all black holes will become stars. The first shot must be position 5, the only star in the universe. When this is done, position 5 becomes a black hole and position 2(above), position 4(left), position 6(right), and position 8(below) all become stars. So it goes, on and on. The best possible score is 11 shots, but watch out for the all black hole universe, because it is a loser, no stars left to shoot. It is possible to lose in 5 tries.

```
LIST
1 CLS
2 POKE 19215,25
6 GOSUB6010
8 COLOR 6,4,2,2
10 OUTPUT"WOULD YOU LIKE  INSTRU
CTIONS?       Y OR N",20,35,1
15 I$=INSTR$(1)
16 CLS
20 IF I$="Y" THEN GOSUB 891
30 COLOR 4,3,0,7
100 A=-1:B=-1:C=-1:D=-1:E=1:F=-1
:G=-1:H=-1:I=-1:J=0
140 PRINT:WINDOW 24
146 GOSUB150
147 GOTO250
149 IF J/3*3=J THEN PRINT"";
150 IF A=1 THEN OUTPUT"*",35,70,
1
151 IF A=-1 THEN OUTPUT"O",35,70
,6
155 IF B=1 THEN OUTPUT"*",60,70,
1
156 IF B=-1 THEN OUTPUT"O",60,70
,6
160 IF C=1 THEN OUTPUT"*",85,70,
1
161 IF C=-1 THEN OUTPUT"O",85,70
,6
165 IF D=1 THEN OUTPUT"*",35,50,
1
166 IF D=-1 THEN OUTPUT"O",35,50
,6
170 IF E=1 THEN OUTPUT"*",60,50,
1
171 IF E=-1 THEN OUTPUT"O",60,50
,6
175 IF F=1 THEN OUTPUT"*",85,50,
1
176 IF F=-1 THEN OUTPUT"O",85,50
,6
180 IF G=1 THEN OUTPUT"*",35,30,
1
181 IF G=-1 THEN OUTPUT"O",35,30
,6
185 IF H=1 THEN OUTPUT"*",60,30,
1
186 IF H=-1 THEN OUTPUT"O",60,30
,6
190 IF I=1 THEN OUTPUT"*",85,30,
1
191 IF I=-1 THEN OUTPUT"O",85,30
,6
```

SHOOTING STARS CONT.

```
200 RETURN
250 IF E=1 THEN GOSUB 300
260 IF A+B+C+D+F+G+H+I= 8 THEN 8
09
270 IF A+B+C+D+F+G+H+I=-8 THEN 8
20
300 POKE24558,85:PRINT"SHOOT":PO
KE24558,170:PRINT"(SHOT TOTAL ="
J")"
305 I$=INSTR$(1):FORK=1TO15:SOUN
D6,19850:NEXT:J=J+1:CLS
393 IF I$="1"THEN GOSUB 509
394 IF I$="2" THEN GOSUB 519
395 IF I$="3" THEN GOSUB 529
396 IF I$="4" THEN GOSUB 539
397 IF I$="5"THEN GOSUB 549
398 IF I$="6"THEN GOSUB 559
399 IF I$="7" THEN GOSUB 569
400 IF I$="8" THEN GOSUB 579
401 IF I$="9" THEN GOSUB 589
402 IF I$="0" THEN SOUND 7,4096:
GOTO 499
404 GOTO 140
499 WINDOW 77
500 PRINT"YOU GAVE UP AFTERONLY"
;J"SHOTS!!!"
505 GOTO 831
509 IF A=-1 THEN 800
510 A=-A:B=-B:D=-D:E=-E:GOTO 500
0
516 GOSUB 1010
517 RETURN
519 IF B=-1 THEN 800
520 A=-A:B=-B:C=-C:GOTO 5000
526 GOSUB 1010
527 RETURN
529 IF C=-1 THEN 800
530 B=-B:C=-C:E=-E:F=-F:GOTO 500
0
536 GOSUB 1010
537 RETURN
539 IF D=-1 THEN 800
540 A=-A:D=-D:G=-G:GOTO 5000
546 GOSUB 1010
547 RETURN
549 IF E=-1 THEN 800
550 B=-B:D=-D:E=-E:F=-F:H=-H:GOT
O 5000
556 GOSUB 1010
557 RETURN
559 IF F=-1 THEN 800
560 C=-C:F=-F:I=-I
566 GOSUB 1010
567 RETURN
569 IF G=-1 THEN GOTO 800
570 D=-D:E=-E:G=-G:H=-H:GOTO 500
0
576 GOSUB 1010
577 RETURN
579 IF H=-1 THEN 800
580 G=-G:H=-H:I=-I:GOTO 5000
586 GOSUB 1010
587 RETURN
589 IF I=-1 THEN 800
590 E=-E:F=-F:H=-H:I=-I:GOTO 500
0
596 GOSUB 1010
597 RETURN
599 RETURN
800 PRINT:SOUND 7,4096:COLOR 0,3
,3,3
801 PRINT"HEY! YOU CAN'T   SHOOT
 BLACK HOLES":GOSUB 1000:PRINT:P
RINT
802 PRINT"YOU CAN ONLY     SHOOT
 STARS!!":GOSUB8010:PRINT:PRINT
803 COLOR4,3,0,7:GOSUB 150:GOTO
300
809 PRINT
810 GOSUB1000:CLS:WINDOW 77:PRIN
T"YOU WON WITH     ONLY"J"SHOTS"
815 GOTO 831
820 GOSUB 1000:CLS:PRINT"YOU LOS
T!!":WINDOW 77
822 PRINT"WITH"J"SHOTS!":GOSUB 1
000
831 CLS:PRINT"DO YOU WANT TO   T
RY AGAIN"
832 PRINT"(Y-NO"
833 I$=INSTR$(1)
834 IF I$="Y" THEN CLS:WINDOW 24
:GOTO 100
835 IF I$="N" THEN CLS:WINDOW 77
:GOTO 889
836 PRINT"YOU DID'NT FOLLOWINSTR
UCTIONS!":GOTO831
889 CLS:PRINT"I HOPE THAT YOU  E
NJOYED THE GAME!"
890 END
891 WINDOW24:COLOR 0,5,2,6
892 PRINT"THERE ARE STARS":A=1:C
=1:E=1:G=1:I=1:GOSUB 150
893 GOSUB 1000
894 PRINT"AND BLACK HOLES":B=-1:
D=-1:F=-1:H=-1:GOSUB 150
895 GOSUB 1000
```

SHOOTING STARS CONT.

```
896 PRINT"IN THE UNIVERSE":GOSUB
 1000
897 PRINT"YOU SHOOT STARS":GOSUB
 8010
900 PRINT:PRINT"NOT BLACK HOLES"
:GOSUB 1000
901 PRINT:PRINT"BY TYPING
 THEIR NUMBER":GOSUB 1000
903 PRINT"TYPE A "0" AS A  SHOT
TO END GAME."
904 WINDOW 77:GOSUB 1000
910 CLS:PRINT"EACH STAR IS IN AG
ALAXY, WHEN YOU SHOOT A STAR,"
911 PRINT"EVERYTHING IN ITSGALAX
Y CHANGES.":PRINT:PRINT"ALL STAR
S BECOME"
912 PRINT"BLACK HOLES, AND ALL B
LACK HOLES  WILL BECOME STARS":G
OSUB1000
915 CLS:COLOR3,5,2,4:PRINT"HERE
ARE THE MAPSOF THE GALAXIES:"
920 PRINT:PRINT"THE NUMBER THAT
  YOU (SHOOT) WILLALTER THAT GAL
AXY"
921 GOSUB1000:COLOR4,3,0,7
925 GOSUB7000
930 FOR X=1 TO 4000:NEXT
931 CLS
932 PRINT"TO WIN, YOU MUST GET T
HIS PATTERN:"
948 GOSUB 8500
949 CLS:COLOR3,0,0,0:PRINT"WOULD
 YOU LIKE TO GO ON?"
950 PRINT"(Y-N)"
951 I$=INSTR$(1)
952 CLS:COLOR 4,3,0,7
953 IF I$="N" THEN 890
1000 FOR X=1 TO 1000:NEXT
1001 PRINT
1003 RETURN
1010 GOSUB6100
1030 SOUND 7,4096:COLOR 4,3,0,7
1040 RETURN
5000 GOSUB1010
5015 RETURN
6010 OUTPUT"*",70,75,2:X=10:Y=10
:Z=6
6012 COLOR 0,1,3,1
6015 OUTPUT"SHOOTING STARS",7,15
,5
6017 FORP=1TO300:NEXT
6020 COLOR0,1,3,1:PLOT X,Y,Z:SOU
ND6,18702:GOSUB6070
6030 Z=6
6040 X=X+3:Y=Y+3:IFX>72THENGOSUB
6080
6050 IF X>72 THEN RETURN
6060 Z=6:GOTO6020
6070 Z=4:PLOT X,Y,Z:SOUND 7,4096
:RETURN
6080 OUTPUT"0",70,75,3
6090 FORN=1TO4:COLOR7,7,7,7:SOUN
D1,362:COLOR3,4,7,0:NEXT
6100 FORN=1TO4:COLOR7,7,7,7:SOUN
D2,17008:COLOR7,0,1,2:NEXT
6120 FORN=1TO4:SOUND5,13200:COLO
R3,4,7,0:NEXT
6130 COLOR4,3,0,7:SOUND7,4096:CL
S:RETURN
7000 SOUND3,152
7005 CLS:B=75
7015 C=0
7020 IF C=3 GOTO 7045
7025 FOR X=1 TO 2:READ A
7030 OUTPUT"*",A,B,5:NEXT
7035 C=C+1
7040 GOTO 7020
7045 IF B=75THEN7140
7050 IFB=68THEN7150
7065 IFB=48THEN7140
7070 IFB=41THEN7140
7080 IFB=34THEN7160
7100 IFB=15THEN7140
7110 IFB=8THEN7170
7120 DATA19,19,48,65,92,92,10,19
,10,19,92,101,10,10,57,57,101,10
1,48,65
7125 DATA48,65,48,65,10,10,57,57
,101,101,10,19,10,19,92,101,19,1
9,
7130 DATA48,65,92,92
7140 B=B-7:GOTO7015
7150 B=48:GOTO7015
7160 B=15:GOTO7015
7170 FORD=1TO110
7175 OUTPUT"-",D,54,3
7180 OUTPUT"-",D,28,3
7185 OUTPUT"!",38,D,3
7190 OUTPUT"!",74,D,3
7195 D=D+2:NEXTD
7200 OUTPUT"1",10,75,6:OUTPUT"2"
,57,75,6:OUTPUT"3",101,75,6
7213 OUTPUT"4",10,41,6:OUTPUT"5"
,57,41,6:OUTPUT"6",101,41,6
7220 :OUTPUT"7",10,8,6:OUTPUT"8"
,57,8,6:OUTPUT"9",101,8,6
7225 SOUND7,4096
```

SHOOTING STARS CONT.

```
7230 RETURN
8010 FOR Z=1 TO 10:FOR X=1 TO 5
8020 OUTPUT"*",60,50,0:NEXT X
8030 FOR Y=1 TO 5
8040 OUTPUT"*",60,50,1:NEXT Y
8050 NEXT Z
8060 OUTPUT"*",60,50,0:RETURN
8500 A=1:B=1:C=1:D=1:E=-1:F=1:G=
1:H=1:I=1:GOSUB 150
8510 FOR X=1 TO 3000:NEXT:RETURN
Ok
```

LETTERS TO THE EDITOR

Dear Steve:

I thought your review and comparison of the two RS-232 ports was fair and accurate. For your information (and Mark's), we believe that the 32K expansion and BASIC will be able to support both ports equally well with the LLIST and LPRINT commands. The POKE addresses for setting BAUD rate, etc., however, may be different depending on which port a user has in his machine.

We should be announcing the 32K unit about the same time as your next issue goes to press to individuals who've requested to be placed on our "32K mailing list". We're not doing a complete separate mailing on the 32K as interest isn't overwhelming, particularly to owners who haven't gotten into programming as yet. It will be covered as well in our Spring Catalog which goes in the mail the end of March.

Best regards,

Dave

David L. Ross,
President

MICRO VIDEO™

* * *

Microsoft Level II Basic Tape (Version 4.7). Includes Command Cards and Interact Level II documentation. $15.00 .
Ron Massie - evenings after 6:30 PM (313) 791-0608.

ROADRACER (Steer through random course using left joystick)
INCOME TAX RECORDING (Enter income & deductions using data statements)
HORSERACE (Up to 8 players),Send $5 ea. or all three for $12 to
Walter Favorite 132 Peach St. Pk. Forest, Il 60466 (all on cassette)

PRODUCT REVIEW

HI-LO Monitor by Harry Holloway

Mr, Holloway does himself an injustice in calling this excellent piece of software a "Monitor". In the context of a micro-computer with only a TV screen and a cassette tape as peripherals, it comes close to being an Operating System.

It has all the components of more conventional monitors (Fills, Dumps, Set memory, etc.) and in addition has hexadecimal arithmetic, for computing addresses; it has an assembler, a disassembler, an ASCII dump mode, and a family of tape control instructions which will gladden the heart of anyone who wishes to really understand the tape system, and use it efficiently.

The package comes with a 25 page instruction manual which is complete and well written. The manual is tutorial in the tape section. The description of the logical structure of an Interact tape is Superb, and is worth the cost of the entire package.

Reviewed by W. M. Carey Medford Lakes, NJ

ROM MONITOR IROM-1 by W. Hendrickson

In the past month I have just added three additions to my Interact computer. (1) RS232 port (Slagh System), (2) a Teletype printer, and (3) ROM Monitor (by W. Hendrickson). I will tell about the third item first.

Installing the ROM is fairly simple and easy due to the socket already in the Interact. Once installed you are ready to go. Go, you say, in what? In Basic? No, but something just as good, if not better. 8080 machine language! I know this sounds crazy but in my opinion this is the only way to go for more memory, faster graphics, and unlimited control. When this ROM is installed you have 13440 bytes of memory (the other 2944 bytes are for screen and stacks) and another 2080 bytes of screen can be used by partitioning the screen for a total of 15520 bytes!

The ROM has 16 commands which have met all of my 8080 programming requirements which include the following: Inspect, Display, Ascii, Exit, User, Hex, Partial, Load, and Write. The only criticism I have about the ROM is that there is no backspace used in it. If you do make a mistake you must start again. This took a while to get used to but presents little problem now. To me this ROM is a great asset for any Interact owner interested in programming in machine language. Unlike other monitor programs I've tried including one that I wrote, all have one thing in common, they take memory. The ROM takes none (except for stacks).

The manual was also well written. A great help to me was a list of subroutines within his ROM. One final note on the ROM that should be mentioned is if you have any new Micro Video games you may be in trouble. Why? Some new programs search for a monitor in the second socket and dump. To prevent this you must wire a switch to the CE (chip enable) or power pins to disable the ROM. When this is done then the programs may be loaded as normal. The reason behind this is to protect their software [illegible] is safe.

PRODUCT REVIEW CONT.

As for my RS232 port I was extremely pleased with the personal service I received from Mark Slagh and more than happy with the results. Thank you, Mark. Finally, I'll tell you about the teletype. I bought the ASR-33 at Computer Mart (Clawson, MI) for $150 and am very pleased with it. I chose the Teletype for two reasons: (1) it was very inexpensive and (2) it can use either roll paper 8½ in. wide or standard typing paper with 64 characters per line. The only trouble with these units over new printers is that they are slow (110 baud rate is approximately 10 characters a second) but it beats manual typing.

I hope the above information is a help to you. Remember Interact is not dying. It is just that it is a newborn and you must be patient while it grows.

Reviewed by George Leggett New Baltimore, MI

KALEIDOSCOPE

by Ruth Ann Halpern 1 Tamarack Rd. Natick, MA 01760

After fiddling with some graph paper for a while, I came up with these two programs. They certainly aren't practical but fill up the screen with pretty patterns. Both run in Graphics Basic and could easily be changed to Fastline by changing all plots to BOX. The first program is an eight way kaleidoscope which images in 8 sectors. The second is a sixteen way kaleidoscope which plots in 16 sections.

The time of display for each program can readily be shortened by changing the outer loop (FOR D = 1TO 100) in both programs. Also colors can be changed but black and white make the best background since the pattern is square and the left and right 18 pixels are left blank on the screen. One could also play with the maximum values of H and W.

```
LIST
10 REM EIGHT WAY KALEIDOSCOPE
20 CLS
30 PRINT "TO GENERATE A    NEW K
ALEIDOSCOPE AFTER THE FIRST  TYP
E ANY KEY"
40 PRINT:PRINT "TO EXIT PROGRAM
 TYPE E"
50 FOR T=1 TO 1000:NEXT
80 CLS
85 COLOR 0,1,4,3
90 FOR D=1 TO 1000
100 CO=INT(4*RND(1)):REM SET COL
OR
110 H=INT(10*RND(1))+1:REM SET H
EIGHT OF BLOCK
120 W=INT(10*RND(1))+1:REM SET W
IDTH
130 CX=56:REM CENTER OF X AXIS
140 CY=38:REM CENTER OF Y AXIS
150 A=INT((38-W)*RND(1))+1:REM S
ET X
155 IF A+CY+W>76 OR CY-A-W<1 GOT
O 150
160 B=INT((A-W)*RND(1))+1:REM SE
T Y
170 IF B<0 GOTO 160
172 IFB+CY+H>76 OR CY-B-H<1 GOTO
 160
180 PLOTA+CX,B+CY,CO,W,H
190 PLOTB+CX,A+CY,CO,H,W
200 PLOTCX-B-H,A+CY,CO,H,W
210 PLOTCX-A-W,B+CY,CO,W,H
220 PLOTCX-A-W,CY-B-H,CO,W,H
230 PLOTCX-B-H,CY-A-W,CO,H,W
240 PLOTB+CX,CY-A-W,CO,H,W
250 PLOTA+CX,CY-B-H,CO,W,H
260 NEXT
270 A$=INSTR$(1)
280 IF A$="E" GOTO 300
290 GOTO 80
300 END
Ok
```

KALEIDOSCOPE CONT.

```
LIST
10 REM 16 WAY KALEIDOSCOPE
20 CLS
30 PRINT "TO GENERATE A    NEW K
ALEIDOSCOPE AFTER THE FIRST  TYP
E ANY KEY"
40 PRINT:PRINT "TO EXIT PROGRAM
 TYPE E"
45 FOR T=1 TO 1000:NEXT
50 CLS
60 COLOR 0,7,6,4
70 FOR D=1 TO 1000
80 CO=INT(4*RND(1)):REM SET COLO
RS
90 H=INT(6*RND(1))+1:REM SET HEI
GHT OF BLOCKS
100 W=INT(6*RND(1))+1:REM SET WI
DTH
110 CX=56: CY=38: REM CENTER OF
SCREEN
120 RX=75: LX=37: REM LEFT AND R
IGHT CENTERS OF X
130 UY=57:LY=19: REM UPPER AND L
OWER CENTERS OF Y
140 A=INT((20-W)*RND(1)):REM SET
 X
145 IF UY+A+W>76 OR LY-A-W<1 GOT
O 140
150 B=INT(39 *RND(1)):REM SET Y
155 IF CY-B-H<1 GOTO 150
170 PLOT RX+A,CY+B,CO,W,H
180 PLOT CX+B,UY+A,CO,H,W
190 PLOT RX-A-W,CY+B,CO,W,H
200 PLOT CX+B,UY-A-W,CO,H,W
210 PLOT LX+A,CY+B,CO,W,H
220 PLOT CX-B-H,UY+A,CO,H,W
230 PLOT LX-A-W,CY+B,CO,W,H
240 PLOT CX-B-H,UY-A-W,CO,H,W
250 PLOT LX+A,CY-B-H,CO,W,H
260 PLOT CX-B-H,LY+A,CO,H,W
270 PLOT LX-A-W,CY-B-H,CO,W,H
280 PLOT CX-B-H,LY-A-W,CO,H,W
290 PLOT RX+A,CY-B-H,CO,W,H
300 PLOT CX+B,LY+A,CO,H,W
310 PLOT RX-A-W,CY-B-H,CO,W,H
320 PLOT CX+B,LY-A-W,CO,H,W
325 NEXT
330 A$=INSTR$(1)
340 IF A$="E" GOTO 360
350 GOTO 50
360 END
Ok
```

STARPLOT

by Michael Rose 10 N. Lippencott Fox Lake, IL 60020

This program plots stars of any reasonable number of points and almost any possible complexity (order). If P = no. of points and O = order (number of intersections of a star segment), then you will get a complete star if $\emptyset < O < P/2$ and O and P are relatively prime integers. My favorite are stars (8,3) and (9,4). If O = 1 then the star is a simple polygon. To plot a new star, depress the fire button on the left joystick.

```
LIST
1 REM STAR PLOT
2 REM INTERACT LEVEL II
3 REM BY MICHAEL ROSE
5 CLS
10 INPUT"NO. OF POINTS";P
15 INPUT"ORDER OF STAR";O
20 Q=60:W=30
25 CLS
30 FORN=1TOP
40 A=6.283185*O*N/P
45 H=30*COS(A)+30
50 V=30*SIN(A)+30
55 R=ABS(H-Q)
60 M=SGN(H-Q)
65 S=ABS(V-W)
70 B=SGN(V-W)
75 IFS>RGOTO200
80 FORT=0TOR
85 U=Q+M*T
90 I=W+(V-W)*T/R
95 PLOTU+26,I+8,1
100 NEXTT
105 Q=H:W=V
110 NEXTN
```

STARPLOT CONT.

```
115 IFFIRE(0)=1GOTO115
120 GOTO5
200 FORT=0TOS
205 U=Q+(H-Q)*T/S
210 I=W+B*T
215 PLOTU+26,I+8,1
220 NEXTT
230 GOTO105
250 END
Ok
```

A NOTE ON THE TAPE RECORDING SCHEME

by William M. Carey 58 Algonquin Trail Medford Lakes, NJ 08055

Many references to the Interact system mention that the erase head is not connected, and / or warn against writing over a previously recorded tape. Such warnings, implied or explicit, puzzle me.

The Interact tape system employs a digital recording technique. In such a recording scheme, writing is performed by saturating the tape in one direction or the other, and the distance between flux changes determines whether a one or a zero has been recorded. Such a scheme is inherently self-erasing; any previous data on the tape is completely destroyed by the new data.

Such is the theory and it works pretty well. I repeatedly record over old data and have never had any trouble from that source. Conceivably, if data is recorded on a tape at time A, and an attempt to record over that data is made at time B, and the tape head has been realigned in the interim, problems could result. I have never seen any such, and those problems would be common to all types of recording. The only solution in the latter case would be to use a bulk eraser on the tape. The bulk eraser does an excellent job, provided you wish to erase the entire tape. It is useless for partial erasures.

* * *

FLAGS

by Al Dietrich P.O. Box 306 Cedarburg, WI 53012

This program consists of two graphics programs. The first is a flag of the state of Texas and the second is a flag of the state of New Mexico. Both of these flags make extensive use of the machine language calls that were listed in the first few issues of Interaction

```
LIST
10 COLOR3,1,2,0
20 REM PROGRAM BY AL DIETRICH 10
/1/80 MADISON, WIS.
40 POKE19215,25:POKE19473,0:POKE
19474,93
50 GOSUB2000
60 OUTPUT"PRESS ANY KEY TO CONTI
NUE",6,20,3
70 A$=INSTR$(1)
80 GOSUB 3000
100 OUTPUT"   FLAG OF NEW    MEX
ICO, LAND OF    ENCHANTMENT",6,2
0,3
110 A$=INSTR$(1)
1999 END
2000 REM DRAW TEXAS FLAG
2010 COLOR0,1,4,7:CLS
2020 GOSUB6000
2030 DATA1,6,93,195,162,5:H=2381
8
2040 GOSUB6050
2050 DATA45,30,2,5,10:B=USR(0)
2060 REM WHITE PART
2070 GOSUB6050
2080 DATA22,54,3,5,40:B=USR(0)
2090 REM RED PART
2100 GOSUB6050
2110 DATA23,54,1,27,40:B=USR(0)
2120 REM DRAW STAR,FIRST DRAW BL
OCK
2130 GOSUB6050
2140 DATA10,9,3,24,20:B=USR(0)
2150 REM DRAW TOP ARM
2160 GOSUB6000
2170 DATA1,6,93,195,151,4
2180 H=23820:GOSUB6050
2190 DATA16,93,10,8,3,14,20
2200 H2=23833:GOSUB6100:DATA8,8,
28,28,28,62,62,62,127,127:B=USR(
0)
2210 REM LEFT ARM
2220 GOSUB6050
2230 DATA16,93,6,8,3,24,12
2240 H2=23829:GOSUB6100
2250 DATA255,127,63,15,3,1:B=USR
(0)
2260 REM RIGHT ARM
2270 GOSUB6050
2280 DATA16,93,6,8,3,24,29
2290 GOSUB6100:DATA255,254,252,2
40,192,128:B=USR(0)
2300 REM LEFT LOWER ARM
2310 GOSUB6050
2320 DATA16,93,10,8,3,31,15
2330 H2=23833:GOSUB6100
2340 DATA15,15,31,31,31,60,56,11
2,96,128:B=USR(0)
2350 REM RITE LOWER ARM
2360 GOSUB6050
2370 DATA16,93,10,8,3,31,26
2380 GOSUB6100:DATA240,240,248,2
48,120,60,28,14,6,1:B=USR(0)
2400 RETURN
2999 END
3000 REM N.M. FLAG
3010 COLOR4,3,1,0
3011 POKE4096,76:POKE6144,3
3015 CLS
3017 GOSUB6000:DATA1,6,93,195,16
2,5
3020 H=23818:GOSUB6050
3022 DATA45,85,1,5,10:B=USR(0)
3025 GOSUB6000:DATA1,6,93,195,15
1,4
3030 H=23820:GOSUB6050:DATA16,93
,8,8,2,18,43
3040 H2=23831:GOSUB6100:DATA7,15
,24,48,96,96,192,192
3050 B=USR(0)
3060 REM LOWER L CIRCLE
3070 GOSUB6050
3080 DATA16,93,8,8,2,26,43
3090 GOSUB6100:DATA192,192,96,96
,48,24,15,7:B=USR(0)
3100 REMUPPER R CIRCLE
3110 GOSUB6050
3120 DATA16,93,8,8,2,26,51
```

FLAGS CONT.

```
3130 GOSUB6100:DATA3,3,6,6,12,24
,240,224:B=USR(0)
3140 REM UPPER R CIRCLE
3150 GOSUB6050
3160 DATA16,93,8,8,2,18,51
3170 GOSUB6100:DATA224,240,24,12
,6,6,3,3:B=USR(0)
3180 REM DRAW ARMS
3190 GOSUB6000
3200 DATA1,6,93,195,162,5
3210 POKE23814,9:POKE23815,1:POK
E23816,2
3220 L=23817:H=23818
3230 FORI=1TO8
3240 READX:READY
3250 POKEL,X:POKEH,Y:B=USR(0):NE
XTI
3260 DATA10,49,10,52,34,49,34,52
,32,46,32,55,12,46,12,55
3270 REM HORIZONTAL ARMS
3280 POKE23814,1:POKE23815,9
3290 FORI=1TO8
3300 READX:READY
3310 POKEL,X:POKEH,Y:B=USR(0):NE
XTI
3320 DATA27,35,24,35,27,59,24,59
,21,57,30,57,21,37,30,37
3330 RETURN
3999 END
6000 REM INITIALIZE BLOCK
6010 FORX=23808TO23813
6020 READY:POKEX,Y:NEXTX
6030 RETURN
6050 REM POSITION BLOCK
6060 FORX=23814TOH
6070 READY:POKEX,Y:NEXTX
6080 RETURN
6100 REM INITIALIZE BIT PATTERN
6110 FORX=23824TOH2
6120 READY:POKEX,Y:NEXTX
6130 RETURN
Ok
```

POLAR FUNCTION PLOTS

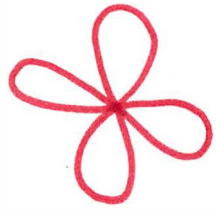
by F. Stauber suite 112 10 Parkway Forest Dr. Willowdale, Ontario

This short program plots polar coordinate functions. By changing the function in line 5000, an infinite number of designs can be plotted.

Try R=SIN(2*COS(2*T)) R=2*COS(SIN(50*T)) R=2*SIN(4*T)

For more information on plotting polar functions, read "The Intricate Graphs of the Polar Functions" in the June, 1980 issue of Creative Computing magazine.

```
5 COLOR0,1,2,3
10 CLS
1000 L=360
1020 M=0
1040 FORA=6TOLSTEP6
1050 GOSUB4990
1060 X=R*COS(T):Y=R*SIN(T)
1070 IFABS(X)>MTHENM=ABS(X)
1080 IFABS(Y)>MTHENM=ABS(Y)
1090 NEXT
1120 M=(M+M/32)/32
1130 FORA=1TOL
1140 GOSUB4990
1150 X=R*COS(T)/M+32
1160 Y=R*SIN(T)/M+32
1170 X=X+20:Y=Y+7
1180 PLOTX,Y,3
1190 NEXT
1200 GOTO1200
4990 T=A*0.017453
5000 R=2*COS(2*T)
5100 RETURN
```



CALIFORNIA USERS' GROUP

A group of Los Angeles area Interact owners are interested in starting a users group. If you are interested, contact
Bob Rupkey 661-19th Street Manhattan Beach, CA 90266
(213) 536-2606

PLANETS

by Harry Holloway P.O. Box 2263 Ann Arbor, MI 48106

This program shows the positions of the sun and the five naked-eye planets in the sky. The coordinate system is ecliptic. After input of a date (year, numeric month) the user is given a choice of wide or narrow fields.

Wide Field - This shows a 360 degree sweep around the ecliptic with 15 degrees on either side. The Y-axis is two fold expanded. The sun is marked by an * and the planets are shown as white pixels. The planets are distinguished by bars above the plot. The shortest bar marks Mercury and the longest marks Saturn.

Narrow Field - The user inputs the ecliptic longitude of the center of the field and the program plots 60 degrees along the ecliptic and 15 degrees on either side. The user also has the option of plotting the brighter stars in the field.

Controls - Left joystick - forward to advance date
back to retard date
left for new option
Left pot - adjust increment for date
Left fire - plot planetary positions in blue
option best used without stars.

```
LIST
0 REM"PLAN".ELEMENTS 1964 (BAKER
)
10 PI=4*ATN(1):K=180/PI:N1=8:N2=
86:K1=4/15:K2=1.6
20 DIMMU(5),M0(5),S1(5),C1(5),S2
(4),X(5),Y(5),Z(5),C(4),C5(4,4)
30 DIMMD(12),NM$(12):DEFFNA(X)=(
X>13)*(X<111)
50 CLS:COLOR0,7,3,4:INPUT"YEAR,M
ONTH";Y,M:Y=INT(Y):M=INT(M)
60 INPUT"W(IDE) OR N(ARROW) FIEL
D";G$:W=(G$="W"):CLS:IFW=-1GOTO8
0
70 INPUT"LONG OF CENTER";L0:L0=I
NT(L0):INPUT"STARS";H$:H$=LEFT$(
H$,1):CLS
80 CL=46-16*W:BL=22-32*W
90 FORJ=14TO110:PLOTJ,70,3:PLOTJ
,BL,3:NEXT
100 FORJ=BL+1TO69:PLOT14,J,3:PLO
T110,J,3:NEXT
110 FORJ=14TO104STEP6:FORJ1=1TO3
:PLOTJ+J1,CL,3:NEXTJ1,J
120 IFW=-1GOTO160
130 FORJ=30TO94STEP16:FORJ1=23TO
26:PLOTJ,J1,3:NEXTJ1,J
140 FORJ=30TO67STEP8:FORJ1=15TO1
7:PLOTJ1,J,3:NEXTJ1,J
150 OUTPUTL0+30,0,20,3:OUTPUTL0,
48,20,3:GOTO170
160 FORJ=38TO86STEP24:FORJ1=55TO
57:PLOTJ,J1,3:NEXTJ1,J
165 OUTPUT360,0,52,3:OUTPUT180,4
8,52,3:OUTPUT"0",108,52,3
170 RESTORE:FORJ=0TO5:READM0(J),
MU(J),EC(J),C1(J),S1(J):IFJ=5GOT
O190
180 READI(J),C2(J),S2(J)
```

PLANETS CONT.

```
190 NEXT:FORJ=0TO12:READMD(J),NM
$(J):NEXT:IFW+(H$="N")<>0GOTO240
200 FORJ=1TON1+N2:READJ1:XS=INT(
J1/100):YS=J1-100*XS
210 XS=XS-K2*(L0-30):XS=110.5-XS
+576*INT(XS/576):IFFNA(XS)<>1GOT
O230
220 PLOTXS,YS,2-(J>N1)
230 NEXT
240 IP=0:DP=1:MP=M:YP=Y:OUTPUT",
INT=",66,6,3
245 OUTPUTDP,0,6,3:OUTPUTNM$(M),
20,6,3:OUTPUTYP,36,6,3
250 GOSUB800:DY=1+MD(M-1)-LY*(M>
2)
260 D=365*(Y-1964)+INT((Y-1961)/
4)-INT((Y-1901)/100)+INT((Y-1601
)/400)
265 D=D+DY-1
270 FORJ=0TO5:E=M0(J)+D*MU(J):E=
E+EC(J)*SIN(E)
280 C=COS(E)-EC(J):S=(1-EC(J)*EC
(J)/2)*SIN(E)
290 X(J)=C*C1(J)-S*S1(J):Y(J)=C*
S1(J)+S*C1(J):IFJ=5GOTO310
300 Z(J)=I(J)*(C*S2(J)+S*C2(J))
310 NEXT:FORJ=0TO5:DX=X(J)-X(5):
YD=Y(J)-Y(5)
320 IFJ=5THENDX=-X(5):YD=-Y(5)
330 X(J)=K*ATN(YD/DX)-180*(DX<0)
+360*(DX>0)*(YD<0)
340 X(J)=62.5-48*W+K2*L0*(1+W)+X
(J)*(K1*W-K2*(1+W))
350 Z(J)=K*Z(J)/SQR(DX*DX+YD*YD)
360 Y(J)=46.5-16*W+Z(J)*(K2*(1+W
)-2*K1*W):NEXT
370 FORJ=0TO4:IFFNA(X(J))<>1GOTO
390
380 C(J)=POINT(X(J),Y(J))
390 NEXT:IFFNA(X(5))<>1GOTO410
400 FORJ=-2TO2:FORJ1=-2TO2:C5(J+
2,J1+2)=POINT(X(5)+J,Y(5)+J1):NE
XTJ1,J
410 FORJ=0TO4:IFFNA(X(J))<>1GOTO
430
420 PLOTX(J),Y(J),1:FORJ1=1TOJ+1
:PLOTX(J),70+J1,3:NEXT
430 NEXT:IFFNA(X(5))<>1GOTO450
440 OUTPUT"*",X(5)-2,Y(5)+2,1
450 IFJOY(0)=1GOTO50
460 IN=1+INT(POT(0)/5):IFIN=IPGO
TO480
470 OUTPUTIP,90,6,0:OUTPUTIN,90,
6,3:IP=IN
480 IFJOY(0)<>8GOTO500
490 IN=-IN:GOTO510
500 IFJOY(0)<>4GOTO450
510 SW=FIRE(0)
520 D=D+IN:DY=DY+IN:J=(DY<1)-(DY
>365+LY):IFJ=0GOTO560
530 Y=Y+J:IFJ=1THENDY=DY-365-LY
540 GOSUB800:IFJ=-1THENDY=DY+365
+LY
550 OUTPUTYP,36,6,0:OUTPUTY,36,6
,3:YP=Y
560 FORJ=0TO12:IFDY>MD(J)-LY*(J>
1)THENM=J+1
570 NEXT:IFM=MPGOTO590
580 OUTPUTNM$(MP),20,6,0:OUTPUTN
M$(M),20,6,3:MP=M
590 DM=DY-MD(M-1)+LY*(M>2):OUTPU
TDP,0,6,0:OUTPUTDM,0,6,3:DP=DM
600 FORJ=0TO4:IFSW=0THENC(J)=3
610 IFFNA(X(J))<>1GOTO630
620 PLOTX(J),Y(J),C(J):FORJ1=1TO
J+1:PLOTX(J),70+J1,0:NEXT
630 NEXT:IFFNA(X(5))<>1GOTO270
640 FORJ=-2TO2:FORJ1=-2TO2:PLOTX
(5)+J,Y(5)+J1,C5(J+2,J1+2):NEXTJ
1,J
690 GOTO270
800 LY=(Y-100*INT(Y/100)=0)-(Y-4
*INT(Y/4)=0)
810 LY=LY-(Y-400*INT(Y/400)=0):R
ETURN
900 DATA.046,71423E-6,.2056,.087
8,.377,.122,.3386,.1876
905 DATA3.896,279624E-7,.0068,-.
4751,.5453,.059,.4179,.5904
910 DATA5.739,914611E-8,.0934,1.
3854,-.6344,.032,.423,-1.4638
915 DATA.139,14504E-7,.0483,5.06
12,1.204,.023,.2987,-5.1938
920 DATA4.201,580875E-9,.0537,.2
217,9.5702,.043,8.6973,-3.9994
925 DATA6.230,172028E-7,.0168,-.
2041,.9788
930 DATA0," ",31,JAN,59,FEB,90,M
AR,120,APR,151,MAY,181,JUN
935 DATA212,JUL,243,AUG,273,SEP,
304,OCT,334,NOV,365,DEC
950 DATA11037,13055,15736,17462,
17956,23847,32443,39840
952 DATA257,2149,2648,4155,4243,
4532,5360,5862,7563,5836,6127,64
38,6925
954 DATA7033,9452,9352,9453,1023
6,10640,10736,10841,13343,12062,
14264
```

PLANETS CONT.

```
956 DATA15048,15849,16030,19729,
20062,20346,21138,22361,22466,23
554,23465
958 DATA23760,25669,25961,27366,
27754,28148,28260,30251,30560,32
260,35847
960 DATA36560,36834,37453,38644,
38638,38748,38732,39440,40037,40
628,39765
962 DATA40958,41644,43235,43836,
43829,44043,44540,45041,45235,45
438,45149
964 DATA45649,48557,48553,49945,
50535,51242,51542,49659,51560,53
164,53151
966 DATA54060,53937,53933,54446,
56058,56252,56660,56952,57058
Ok
```

COLOR SEARCH

by Ken Stuempges 6261 N. Joyce Milwaukee, WI 53224

I was inspired to write Color Search after using the Sound Search routine in the Basic Examples Booklet, published by Micro Video. It is a handy little program to display all the possible color combinations for both output and print statements. I use it to come up with colorful combinations for Basic games.

The program uses the joystick and fire buttons to alter screen color and the corresponding colors for output and print statements. Variables A, B, C, and D relate to those used with the COLOR statement. Variable E relates to the output color. Every possible combination is obtainable. The program initializes with color 3,0,0,0 and output color 1. This is for effect only. The variables will all step from 0 to 7 and then start over depending on the number of times the joystick is moved. Holding the joystick up and simultaneously depressing the fire button will reset the variables.

```
LIST
10 A=3:B=0:C=0:D=0:E=1
20 COLOR A,B,C,D:CLS
30 OUTPUT"COLOR SEARCH",20,74,E
32 OUTPUT" ____________",12,71,
E
35 OUTPUT"OUTPUT X, Y, E",12,65,
E
40 OUTPUT"COLOR A, B, C, D",8,57
,E:WINDOW50
50 PRINT"      A ="A:PRINT"
 B ="B:PRINT"      C ="C
60 PRINT"      D ="D:PRINT"
 E ="E
70 IF JOY(0)=4 THEN PRINT CHR$(7
):GOTO 130
80 IF JOY(0)=2 THEN PRINT CHR$(7
):GOTO 140
90 IF JOY(0)=8 THEN PRINT CHR$(7
):GOTO 150
100 IF JOY(0)=1 THEN PRINT CHR$(
7):GOTO 160
110 IF FIRE(0)=0 THEN PRINT CHR$
(7):GOTO 170
120 GOTO 70
130 A=A+1:GOTO 200
140 B=B+1:GOTO 210
150 C=C+1:GOTO 220
```

COLOR SEARCH CONT.

```
160 D=D+1:GOTO 230
170 E=E+1:GOTO 240
200 IF A=8 THEN A=0:GOTO 20
210 IF B=8 THEN B=0:GOTO 20
220 IF C=8 THEN C=0:GOTO 20
230 IF D=8 THEN D=0:GOTO 20
240 FOR X=1 TO 75:NEXT
245 IF FIRE(0)=0 AND JOY(0)=4 TH
EN 10
250 IF E=8 THEN E=0
255 GOTO 20
Ok
```

WUMPUS

by Kevin TenBrook 12813 Westpark Houston, TX 77082

This is an improved version of Wumpus. There are different caves to try. 1. Dodecahedron - the standard cave and tunnels 2. String of beads - circular tunnels with side passages 3. Dendrite (with degengenegracies) - looks like a root. Degeneracies include rooms where tunnels lead back to room you are in, and rooms where two tunnels lead to the same room. Easy to get trapped in this cave.

Other modifications : 1. If a bat bothers you, shoot it. 2. If you shoot an arrow and miss, you can go and get it back.

(For more information on Wumpus see More Basic Computer Games published by Creative Computing Press.)

```
LIST
1 REM ***WUMPUS REMAKE - BY KEVI
N TENBROOK, 1980 **
5 COLOR7,4,4,4:CLS
10 DIMS(19,2),L(5),AL(4),P(4)
20 DEFFNA(X)=INT(RND(1)*19.5):DE
FFNB(X)=INT(RND(1)*2.5)
30 DEFFNC(X)=INT(RND(1)*3.5)
40 PRINT"WELCOME TO THE   WUMPUS
 HUNT. I AMYOUR GUIDE. IN   FRON
T OF YOU IS"
50 PRINT"A MOUNTAIN. IN   THE MO
UNTAIN ARE 3 CAVES. IN FRONTOF E
ACH CAVE IS";
55 PRINT" ASIGN. THE SIGNS  SAY:
":PRINT:FORI=0TO200:NEXTI
56 GOSUB1000
60 PRINT"CAVE NO.1":PRINT"DODECA
HEDRON":PRINT"WARNING! WUMPUS  I
NSIDE!":PRINT
65 GOSUB1000
70 PRINT"CAVE NO.2":PRINT"STRING
 OF BEADS  WARNING! ANGRY   WUMP
US INSIDE!":PRINT
75 GOSUB1000
80 PRINT"CAVE NO.3":PRINT"DENDRI
TE":PRINT"WARNING! WUMPUS  LOST
INSIDE.":PRINT
90 INPUT"WHICH CAVE DO YOUWANT";
N:PRINT
100 N=INT(N):IFN<1ORN>3THEN90
110 ONNGOSUB700,735,765
120 FORI=0TO19:FORJ=0TO2:READS(I
,J):NEXTJ:NEXTI
130 FORI=0TO4:AL(I)=20:NEXTI
140 FORI=0TO5:L(I)=FNA(X):NEXTI
150 FORI=0TO5:FORJ=ITO5:IFI=JTHE
N160
155 IFL(I)=L(J)THEN140
160 NEXTJ:NEXTI
170 A=5:L0=L(0):PRINT"OK-LET'S H
UNT THEWUMPUS!":PRINT
180 FORI=1TO5:FORJ=0TO2:IFS(L(0)
,J)<>L(I)THEN230
190 ONIGOTO200,210,210,220,220
200 PRINT"I SMELL A WUMPUS!":PRI
NT:GOTO230
210 PRINT"I FEEL A DRAFT...PIT N
EAR.":PRINT:GOTO230
```

WUMPUS CONT.

```
220 PRINT"BATS NEARBY...   CAREF
UL!":PRINT
230 NEXTJ:NEXTI
240 FORI=0TO4:IFAL(I)<>L0THEN250
245 A=A+1:AL(I)=20:PRINT"LOOK HE
RE!        I FOUND AN ARROW.":PRI
NT
250 NEXTI
260 PRINT"YOU ARE IN ROOM:";SPC(
8);L(0)+1:PRINT
270 PRINT"TUNNELS LEAD TO:";S(L0
,0)+1;S(L0,1)+1;S(L0,2)+1:PRINT
280 INPUT"MOVE(M),SHOOT(S),OR QU
IT(Q)";A$:IFA$="Q"THENEND
290 IFA$="M"THEN320
300 IFA$="S"THEN460
310 PRINT"WRONG DUMMY":PRINT:GOT
O280
320 INPUT"WHERE TO";L0:PRINT:L0=
INT(L0-1)
330 IFL0>19ORL0<0THENPRINT"WRONG
 DUMMY":GOTO320
340 FORI=0TO2:IFS(L(0),I)=L0ORL(
0)=L0THEN370
350 NEXTI
360 PRINT"OUCH! BUMPED A   WALL.
 NO TUNNEL  THAT WAY.":PRINT:GOT
O280
370 L(0)=L0:IFL0=L(1)THENPRINT"O
OPS....BUMPED A WUMPUS.":PRINT:G
OSUB420
380 IFL0=L(2)ORL0=L(3)THENPRINT"
YYYYYIIIIIEEEE...FELL IN A PIT."
:GOTO680
390 IFL0=L(4)ORL0=L(5)THENPRINT"
SNATCH BY A BAT! YOU'RE LOST.":P
RINT:GOTO410
400 GOTO180
410 L0=FNA(X):GOTO370
420 IFFNC(X)=3THEN440
430 L(1)=S(L(1),FNB(X))
440 IFL0=L(1)THENPRINT"TSK TSK T
SK...   WUMPUS WUMPED YOU ":GOTO
680
450 RETURN
460 INPUT"NUMBER OF,ROOMS- (1-5)
";N:N=INT(N-1)
470 IFN<00RN>4THENPRINT"WRONG DU
MMY.":PRINT:GOTO460
480 FORI=0TON:INPUT"ROOM =";P(I)
:P(I)=INT(P(I)-1)
490 IFP(I)<00RP(I)>19THENPRINT"E
RROR START FROM SCRATCH.":PRINT:
GOTO480
500 NEXTI
510 A=A-1:A9=L(0)
520 FORI=0TON:FORJ=0TO2:IFS(A9,J
)=P(I)THEN550
530 NEXTJ
540 A9=S(A9,FNB(X)):GOTO560
550 A9=P(I)
560 AL(A)=A9:IFA9<>L(1)THEN580
570 PRINT"HA! YOU GOT THE  WUMPU
S! HE WAS INROOM";L(1)+1:PRINT:G
OTO630
580 IFA9<>L(0)THEN600
590 PRINT"YEOW! ARROW IN   YOUR
ARSE!":GOTO680
600 FORJ=4TO5
610 IFL(J)<>A9THEN630
620 L(J)=20:AL(A)=20:PRINT"SKEWE
RED A BAT!":GOTO660
630 NEXTJ
640 NEXTI
650 PRINT"ARROW MISSED."
660 IFA=0THENPRINT"OUT OF ARROWS
.":GOTO680
670 GOSUB420:GOTO180
680 INPUT"PLAY AGAIN";A$:IFA$="Y
"THEN90
690 END
700 RESTORE710
710 DATA1,4,7,0,2,9,1,3,11,2,4,1
3,0,3,5,4,6,14,5,7,16,0,6,8,7,9,
17
720 DATA1,8,10,9,11,18,2,10,12,1
1,13,19,3,12,14,5,13,15,14,16,19
,6
730 DATA15,17,8,16,18,10,17,19,1
2,15,18:RETURN
735 RESTORE740
740 DATA1,2,19,0,2,3,0,1,3,1,2,4
,3,5,6,4,6,7,4,5,7,5,6,8,7,9,10,
8,10
750 DATA11,8,9,11,9,10,12,11,13,
14,12,14,15,12,13,15,13,14,16,15
,17
760 DATA18,16,18,19,16,17,19,0,1
7,18:RETURN
765 RESTORE770
770 DATA0,0,4,1,1,4,2,2,5,3,3,5,
0,1,6,2,3,6,4,5,9,7,8,8,7,7,9,6,
8
780 DATA10,9,12,13,11,12,12,10,1
1,11,10,14,15,13,16,17,13,18,19,
14
790 DATA16,16,14,17,17,15,18,17,
15,19,19:RETURN
1000 FORM=0TO1000:NEXTM
1010 RETURN
Ok
```

ADVERTISING RATES

Due to the increasing printing costs and the new increase in postal rates, I must revise the advertising rates. One issue costs are :

$40 per full page approx. 10 in
$20 per half page approx. 5 in.
$10 per quarter page approx 2½ in.
$1.00 per line over the first 5 free lines to subscribers

Non subscribers add 25% to these prices. Discounts available on consecutively running ads.

MAY - JUNE ISSUE DEADLINE FOR ADVERTISING MAY 2

C.A.C.H.E.

The Chicago Area Computer Hobbyist Exchange has an Interact Special Interest Group. CACHE meetings are the third sunday of every month at the DeVry Institute 3300 N. Campbell in Chicago.

For more information on the Interact SIG contact Jean Barber at (312) 945-4171. General CACHE meeting information is available on their recorded Hotline (312) 849-1132.

★★

INTERACTION ON TAPE

INTERACTION PROGRAMS are available on data cassettes at $5.00 per issue. Please specify issue no. and year. If you notify me that you will be ordering every issue this year, I will have a cassette already recorded and ready to mail when your order and check arrives.
from George Leggett 52895 Bunker Hill New Baltimore, MI 48097

EDU-BASIC PRINTER OVERLAY - For use with the Slagh U80 port. Allows you to output and list programs in Edu-Basic to a printer. Also allows you to manipulate memory and call machine language subroutines.

QUEST in Edu-Basic - An 8K adventure program. You must retrieve a treasure from an underground maze. Descriptions are given of each room and you have 6 directions to try to proceed. A pirate lurks in the maze and may steal the treasure back!

8080 DISASSEMBLER in Level II Basic - Lists addresses, contents, ASCII characters, and 8080 op codes and registers for any memory location. With complete instructions and sample output listing form

$5.00 for each program on data casstte from :
DAVID J. SCHWAB 10 Jay Lee Ct. Ann Arbor, MI 48104

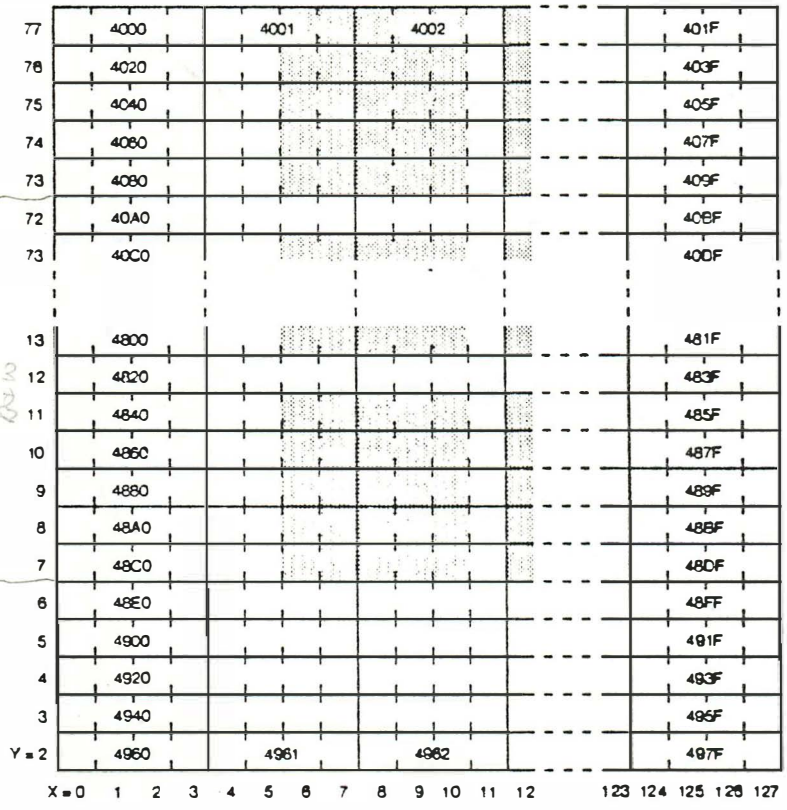
INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

May — June, 1981 VOL. II, no. 3

Interact Display Format

By: Richard C. Pasco
235 College Ave.
Mountain View CA 94040
(415) 968-9867



Notes:

Numbers in boxes are hexadecimal byte addresses.

The low order (least significant) two bits of each byte correspond to the leftmost pixel.

Addresses 4980 through 49FF do not appear on the TV.

But most tapes load them with the picture, and they aren't used for anything else.

Numbers in margins are decimal coordinates, as understood by PLOT and OUTPUT.

Columns right of 112 do not appear on the TV, but they are shown above.

Only those 76 rows which appear on the TV are shown above.

Each row occupies three video raster scan lines, or 228 for the visible picture.
There are 34 more blank scan lines during vertical retrace, for a total of 262.

With OUTPUT, X and Y refer to the upper-left pixel of the first character.

Shaded areas are occupied by characters from PRINT statements.

PRINT column N corresponds to OUTPUT X = 6*N, for N from 1 to 17.
PRINT line N from bottom corresponds to OUTPUT Y = 5 + 6*N, for N from 1 to 12.
The topmost (12th) line of characters may lie off the screen of some TVs.

PLOT and OUTPUT do not work for Y = 77.

CANNON

by Glenn Darling 1111 Barnes Rd. Mason, MI 48854

This game is styled after the old Basic game known as Elevate. The object of Cannon is to shoot an approaching ship from your position on shore. The left joystick is used to change the cannon elevation in either 5 or ½ degree increments. It should be fairly easy to sink the ship in the 8 shots allowed. The 20 million in line 325 can be halved or doubled to make it twice as hard or easy to sink the boat. Line 595 can also be changed to make the ship move faster or slower. If you have a very small TV, you might want to change lines 200, 220, 315, 545, and575 to assure that all the text will appear on the screen.

```
LIST

2 CX=12:CY=20:PR=0:CLS
4 A=0:COLOR7,1,2,0
10 PRINTCHR$(8):GOSUB800
20 PRINT"DO YOU WANT TO   KNOW HOW TO PLAY?"
25 PRINT:PRINT:A$=INSTR$(1)
30 IFA$<>"Y"THEN80
35 CLS:COLOR4,1,0,7
40 PRINT"YOU ARE A PRIVATEON COASTLINE DUTYDURING THE WAR   OF 1812."
45 L=1000:GOSUB770:CLS
50 PRINT"YOU SPOT AN ENEMYBATTLESHIP IN THEDISTANCE."
53 GOSUB770:CLS
60 PRINT"YOU MUST SINK HIMBEFORE HE GETS   CLOSE ENOUGH TO  DESTROY YOUR"
62 PRINT"POSITION"
65 GOSUB770:CLS
70 PRINT"USE THE JOYSTICK TO SELECT THE    CANNON ELEVATION. "
72 PRINT"PRESS HIT TO FIRE":PRINT:PRINT"GOOD LUCK!!"
75 GOSUB770:CLS
80 PRINT"DO YOU NEED AN   EXTRA HAND TO    HELP WITH THE    DISTANCES?"
82 P$=INSTR$(1):CLS
100 COLOR0,1,2,4
130 R=INT(50000-(10000*RND(1)))
132 CLS
135 OUTPUT"TARGET RANGE:",6,70,2
137 OUTPUTR,6,62,1:OUTPUT" YARDS",50,62,2
140 IFR>10000THEN158
145 H=1:R=12000
158 PRINT:WINDOW11:OUTPUTCHR$(1),CX,CY,2
162 OUTPUTCHR$(5),4,CY,1
165 OUTPUTCHR$(7),(R/500),CY,3
168 IFH=1THEN700
170 GOSUB200
180 GOSUB300
185 GOSUB400
187 IFPR>85GOTO750
190 GOSUB500
193 WINDOW77
195 IFS=0THEN900
196 GOTO132
200 OUTPUT"ELEVATION:",6,6,2:PN=PR
205 IFJOY(0)=4THENPR=PR+.5
207 IFJOY(0)=1THENPR=PR+5
210 IFJOY(0)=8THENPR=PR-.5
```

Cannon, cont.

```
212 IFJOY(0)=2THENPR=PR-5
215 IFPR<0THENPR=0
216 IFPR>100THENPR=100
220 OUTPUTPN,65,6,0
225 OUTPUTPR,65,6,2
250 IF PN<>PR THEN 200
255 IF FIRE(0)=0 THEN RETURN
260 GOTO200
300 PRINT:PRINT
305 T1=(PR*2)/57.2958
310 J=INT(50000*SIN(T1))
312 IFP$="N"THEN325
315 OUTPUT"DISTANCE: ",6,6,2:OUTPUTJ,55,6,1
325 N=R-J:X=20000000/R
330 IFN<0THENS=2
335 IFN>0THENS=1
337 IFABS(N)>XTHEN340
338 S=0:J=R-4000
340 IFS=1THENJ=J-7000
345 IFS=2THENJ=J+3500
350 FORI=1TO10
351 IFJ<=0THENJ=1
352 K=J*(I/10)
355 XP(I)=INT(K/500)
357 IFPR<45THEN363
360 YP(I)=INT(40*(SIN((K/J)*3.14159)))
362 GOTO365
363 YP(I)=INT(PR*(SIN((K/J)*3.14159)))
365 NEXTI
399 RETURN
400 OUTPUTCHR$(4),4,CY,0
401 OUTPUTCHR$(6),4,CY,1
402 GOSUB770
403 SOUND3,1000
404 OUTPUTCHR$(4),CX,CY,0
406 OUTPUTCHR$(2),CX,CY,2
408 OUTPUTCHR$(4),CX,CY,0
410 OUTPUTCHR$(3),CX,CY,2
66
412 OUTPUTCHR$(4),CX,CY,0
414 OUTPUTCHR$(2),CX,CY,2
416 OUTPUTCHR$(1),CX,CY,2
418 SOUND7,4096:SOUND3,264
420 FORI=1TO10:L=I-2
423 IFL<1THENL=1
425 PLOTXP(I)+CX,YP(I)+CY,1
427 PLOTXP(L)+CX,YP(L)+CY,0
430 NEXTI
432 PLOTXP(9)+CX,YP(9)+CY,0
433 PLOTXP(10)+CX,YP(10)+CY,0
435 SOUND7,4096
440 OUTPUTCHR$(4),4,CY,0
445 OUTPUTCHR$(5),4,CY,1
499 RETURN
500 ON(S+1)GOTO505,540,570
505 COLOR3,4,5,6
507 OUTPUT"SUNK!!",35,40,1
```

cannon, cont.

```
508 PRINT:PRINT:GOSUB800
510 COLOR0,1,2,4:T=R/500:SOUND3,244
514 OUTPUTCHR$(9),(R/550),CY,1
515 OUTPUTCHR$(9),(R/450),CY+6,1
517 OUTPUTCHR$(4),(R/500),CY,0
519 OUTPUTCHR$(8),(R/500),CY,2
520 GOSUB770
521 OUTPUTCHR$(4),T,CY,0
522 OUTPUTCHR$(8),T,CY-4,2
523 L=200:GOSUB770
524 OUTPUTCHR$(4),T,CY-4,0
526 OUTPUTCHR$(8),T,CY-8,2
527 GOSUB770
528 OUTPUTCHR$(4),T,CY-8,0
530 OUTPUTCHR$(8),T,CY-12,2
531 GOSUB770
532 OUTPUTCHR$(4),T,CY-12,0
535 SOUND7,4096:GOTO580
540 PRINT:PRINT
545 OUTPUT"SHORT",50,6,3
549 GOTO580
570 PRINT:PRINT
575 OUTPUT"LONG",50,6,3
580 A=A+1
585 IFA>7ANDS<>0THEN600
590 L=1000:GOSUB770
595 R=R-INT(RND(1)*8500)
597 L=800:GOSUB770:RETURN
600 CLS:WINDOW77
610 PRINT"YOUR'E ALL OUT OFSHOTS,MATE. YOU  ARE AT THE MERCY OF THE SHIP."
615 PRINT:PRINT:I=INT(RND(1)*4+1)
618 ONIGOTO620,630,630,640
620 PRINT"THE SHIP SAILS   CLOSER TO SHOOT  YOU AND SINKS"
622 PRINT"ON A REEF.       LUCKY TODAY,     AREN'T YOU?"
625 L=999:GOSUB770:GOTO900
630 R=12000:H=1:GOTO162
640 PRINT"THE SHIP'S CREW  TAKES A VOTE AND DECIDES TO SPARE YOU."
645 L=999:GOSUB770:GOTO900
700 SOUND3,58:H=0
710 OUTPUTCHR$(9),CX,CY,1
715 OUTPUTCHR$(9),CX-4,CY,1
740 OUTPUT"SHIP GETS YOU!!",6,6,1
745 L=1000:GOSUB770
749 SOUND7,4096:GOTO900
750 CLS:WINDOW77
755 PRINT"YOU SHOT STRAIGHTUP,YOU DUMMY!!":L=900:GOSUB770:GOTO900
770 FORX=1TOL:NEXTX:RETURN
800 POKE19215,25:RESTORE
808 POKE24545,0:POKE24546,94
810 FORX=24064TO24083
812 READY:POKEX,Y:NEXTX
815 DATA9,9,0,128,1,0,2,0,4,0,28,0,34,0,98,0,162,0,28,0
820 FORX=24084TO24101
822 READY:POKEX,Y:NEXTX
825 DATA0,0,1,0,2,0,4,0,28,0,34,0,98,0,162,0,28,0
830 FORX=24102TO24119
832 READY:POKEX,Y:NEXTX
```

Cannon, cont.

```
835 DATA0,0,0,0,0,0,12,0,60,0,68,0,196,0,196,0,56,0
840 FORX=24120TO24137
842 READY:POKEX,Y:NEXTX
845 DATA255,128,255,128,255,128,255,128,255,128,255,128,255,128,255,128
846 DATA255,128
850 FORX=24138TO24155
852 READY:POKEX,Y:NEXTX
855 DATA12,0,12,0,30,0,45,0,76,128,12,0,18,0,18,0,18,0
860 FORX=24156TO24172
862 READY:POKEX,Y:NEXTX
865 DATA30,0,45,0,63,0,12,0,12,0,12,0,18,0,18,0,18,0
870 FORX=24173TO24190
872 READY:POKEX,Y:NEXTX
875 DATA4,0,4,0,6,0,6,0,71,0,47,128,255,128,63,128,31,0
880 FORX=24191TO24208
882 READY:POKEX,Y:NEXTX
885 DATA2,0,7,128,15,128,31,128,63,128,3,128,3,0,2,0,2,0
890 FORX=24209TO24226
892 READY:POKEX,Y:NEXTX
895 DATA0,0,73,0,0,0,40,128,130,0,68,0,0,0,0,0,0,0
899 RETURN
900 CLS:WINDOW77
950 PRINT"DO YOU VOLUNTEER FOR ANOTHER ROUNDOF COASTAL DUTY?"
955 A$=INSTR$(1):A=0:CLS
970 IFA$="Y"THEN80
998 WINDOW77
999 END
Ok
```

BASIC MAILING LIST

by Stephen Cook

Because the Interact Basics do not save string arrays to tape, this program is a Basic solution to printing mailing lists. By programming information in data statements and then appending this program, lines 1000-2000. with Ezedit about 50 - 60 names can be stored in one program. Precede each address information with the number of data items to follow (see lines 1,5. 11). A zero must be the last data statement otherwise an OD error message will result. Remember a comma in a data statement will indicate a new data item follows. For example line 8 has 1 data statement while line 9 has 2 data items. Quotes are not needed unless you want to indicate a comma within a data item.

This program was written for MicroVideo's RS-232 Basic and an Epson MX-80 printer. For the Slagh port and other printers change the LPRINTs to PRINTs and modify lines 1020 and 1080. As the program stands it was designed to print continuous labels using line 1180 to advance to the top of the next label. By changing line 1150 to LPRINT A$; adding a LPRINT SPC(1); to line 1140 and changing the top of form in line 1180 to a carraige return. the name and address will print on a single line.

Mailing list, cont.

```
1 DATA3
2 DATASAMPLE
3 DATADATA
4 DATAENTRY
5 DATA6
6 DATASTEPHEN COOK
7 DATAINTERACTION NEWSLETTER
8 DATA15356 PREVOST
9 DATADETROIT, MI  48227
10 DATA313-272-7594
11 DATA0
1000 REM NAME LIST PROGRAM
1010 CLEAR:CLS
1020 POKE25097,10:REM AUTO LF FOR RS232 BASIC ONLY
1030 PRINT"   INTERACTION":PRINT"   NEWSLETTER":PRINT"   SUBSCRIBERS"
1040 PRINT"   LIST (A-B)":PRINT"   OK (Y-N)?"
1050 Q$=INSTR$(1)
1060 IFQ$="N"THEN1920
1070 INPUT"FORM LENGTH      (IN LINES)";FL
1080 LPRINTCHR$(27)"C"CHR$(FL);:REM FORM LENGTH CONTROL CODE FOR EPSON
1090 FORN=1TO100
1100 READN1
1110 IFN1=0THEN1910
1120 FORL=1TON1
1130 READA$
1140 REM TABS OR SPACES
1150 LPRINTA$:REM USE ; FOR ONE LINE PRINTING
1160 NEXTL
1170 REM FORM FEED OR CR FOR SINGLE LINE
1180 LPRINTCHR$(12);:REM OR CHR$(13)
1190 NEXTN
1900 REM
1910 PRINT"END OF LIST!"
1920 LPRINTCHR$(7);:PRINTCHR$(7)
1930 PRINT"HIT SPACE BAR TO CLOAD NEXT LIST"
1940 Q$=INSTR$(1)
1950 IFQ$<>" "THENEND
2000 CLOAD
```

* * *

++

--ROM MONITOR OWNERS-- Now available, ROM MONITOR APPLICATION NOTE #2! This note is dedicated to the Interact tape system and details: a) How the Interact Tape system works; b) Tape formats; c) Using the ROM MONITOR to write from tape list anywhere in memory!; d) Back-up copies using the ROM MONITOR; d) Loading tapes to user defined locations; e) Displaying tape headers; and more. order now- RMAPP#2 at $3.00 ea (to cover printing, postage & handling); from Walt Hendrickson, 2313 W 181st street, Torrance, CA. 90504.

++

PRODUCT REVIEWS

Product Review: Slagh System Services U80M Port Kit

by Richard Pasco
235 College Ave.
Mountain View, Calif. 94040

The Slagh System Services U80M is an RS-232 interface card for the Interact. It is a PC card about 3" x 5" containing a Motorola 6850 ACIA, an Intel 8212 bus interface, and a few TTL and analog parts. There is one 40-pin wirewrap socket on the card; installation is accomplished by unplugging the 8080 CPU from the Interact, plugging the wirewrap pins into the vacated socket, and reinstalling the 8080 into the wirewrap socket. Power, clock, and bus connections, as well as mechanical support, are derived from this connection. The only other connection is an (optional, see below) 7-conductor cable which is soldered to the board and terminates in a DB25 plug just outside the Interact.

When mine arrived, its packing was inadequate to withstand a cross-country journey: the PC board, small components in coin envelopes, IC's in carriers, and IC sockets stuck into a foam block had all been dumped loosely into a polybubble mailing envelope. During shipping, some of the sockets had fallen out of the foam and their pins were broken off by subsequent handling. This could be avoided by putting tape or rubber bands around the sockets in the foam, and putting everything with appropriate padding into a rigid box. This could be easily afforded with the 5% asked for shipping.

Assembly went quickly thanks to very complete instructions. The provision of IC sockets was an unexpected plus. The glass-epoxy board with plated-through holes is of high quality.

The advertisement does not make clear that the $9.00 cable is in most instances required. I hadn't ordered one with my U80M, thinking that the cable was a long two-headed DB-25 cable for connection from the Interact to the the printer or modem. It isn't! It is *the* U80's only DB-25 connector, and you'll need it if you want a DB-25 connector at the back of your Interact. The U80M itself ends at the solder holes on its PC board. Fortunately, Mark Slagh told me of this when I phoned him regarding my order.

Also on the subject of the cable, the kit assembly instructions tell you to solder the cable to the U80 PC board. If you do this the large DB-25 connector won't fit through the power-cord hole on the lower housing. Two solutions: either route the cable through the hole before soldering it to the PC board, or melt with your soldering iron a notch for the cable in the upper housing. I chose the latter approach, which allows for easier dissassembly.

Installation was also easier than expected. Several articles in *Interaction* as well as the U80 installation instructions stress the delicacy of the operation, so I prepared for the worst. Assisted by more caution than necessary, it was easy to avoid the known pitfalls of broken keyboard wires, bent IC pins, etc.

I was quite pleasantly surprised when the interface worked the first time I tried it (with BASIC and Slagh's P8011 overlay). I spent the rest of the afternoon listing all my programs and modifying them to take advantage of printed output.

The handshake logic between the printer and the BASIC overlay is a nice feature. If your printer has a buffer, a fast data rate is selected, and the printer's "ready" line is connected to the port's "clear to send" line, then BASIC may LIST or PRINT quickly into printer's buffer and get on with other things, while the printer prints the received data at its slower speed. If the handshake logic reports that the printer's buffer is full, the P8011 holds up the program until the printer is ready again.

I'm looking forward to receiving my P8012 BASIC overlay when it is released. Mark Slagh promises nice features not found in the P8011, such as ability to take input data from the port (thus making it easy to use a terminal with a good keyboard, or to write a terminal emulator in BASIC), and/or to send NULLs after CRs, to allow the port to be used with printers unable to use the handshake logic described above.

Product reviews, cont.

Perhaps the nicest feature of the U80 is the excellent support I have received from Mark Slagh. He's pleasant and informative over the phone, returning phone calls and sharing willingly of his expertise and limited time.

INTERACT OPERATING SYSTEM by Richard Ferris

With the addition of this 2K ROM operating system, the Interact finally grows up. I don't think any serious Interact user should be without this I.O.S. (Interact Operating System).

The I.O.S. is a 2716 EPROM that plugs into the spare socket in your machine. It comes with a user's guide that completely explains how to use the over 20 commands the I.O.S. has. I also got the source listing Mr. Ferris advertised, and it proved to be as complete as I could of hoped It's over 90 pages long! With the listing is included a free, completely, annotated listing of the original ROM. This listing alone is worth the price of the entire package.

I know there is another ROM monitor available, so in order that readers who are considering buying a ROM monitor can make an informed choice, I'd like to give a somewhat lengthy description of the I.O.S. First the I.O.S.'s good points:

1) It has an extensive library of commands, 23 in all, plus 3 user defined command jumps. The commands available at a keystroke are: A- ASCII dump, B- go back to ROM 1, C- clear screen, D- display memory, E- memory exercisor routine, F- fill memory, G- go, with or without multiple user set breakpoints, H- hex sums and differences, I,J,K- user defined command jumps, L- load tape, M- move memory, N- convert decimal to hex, O- convert hex to decimal, P- partial screen, Q- quit, R- rewind, S- substitute memory, T- disassembler U- uncover address references, V- compute checksum for a range of memory, W- write tape (with options), X- examine registers, Y- turn interrupt control code on or off, Z- jump back to 4C00 hex (very handy command for Basic programmers).

2) On power up, you can clean out memory for a fresh start, or go back to ROM 1 for a convential start.

3) There is an interrupt control routine which allows a panic breakpoint function when a control P is pressed or a quit when control Q is pressed.

4) The scrolling speed of the screen has been increased by a factor of 6. It is amazing how slow the old way seems in comparison.

5) All command routines are set up as independent subroutines and can be accessed directly from your programs. (this however requires that you have the listing so know where his routines are located, which registers they destroy, etc.).

6) The I.O.S. has been written such that output to port routines may be easily appended in RAM.

Now for a list of the I.O.S.'s bad points:

1) Disassembler requires a separate tape loaded table to be loaded somewhere in RAM (anywhere you want). This tape is included with the I.O.S.

PRODUCT REVIEWS, cont.

2) If I expand my Interact in the future to 32K the Exerciser routine ans clear memory routine will not function correctly. However, if you specify you want an I.O.S. for a 32K machine, the same two routines won't work quite right in a 16K machine.

3) MicroVideo, as you may know has installed bombs in some of their newer programs, which blow when they find something where the second ROM is. This is an inconvenience which is overcome by modifying the program using the U command to find references in the program to the ROM 2 area. (This problem I can't blame on the I.O.S. but rather on MicroVideo for pulling such a lousy stunt.)

Well, I hope this information is helpful to anyone who is considering buying a ROM operating system. For my part, I am thoroughly satisfied with my purchase, and heartily recommend it to others.

Reviewed by Kevin TenBrook Houston, TX

UPPER AND lower CASE FOR THE INTERACT

by Richard Pasco 235 College Ave. Mountain View, CA 94040

While it is widely believed that the Interact keyboard is upper-case only, in fact the ROM routines which scan the keyboard and encode it into ASCII provide full upper and lower case codes. BASIC, however, translates everything into upper case. This translation can be disabled with POKE 24652, 201 and reinstated, if desired, with POKE 24652, 254.

(Ed note - lower case can be enabled in MicroVideo's RS-232 BASIC by POKEing location 24651 as above.)

★★

BASIC CURSOR

```
********************  CURSOR PROGRAM  ***************************
*                                                                 *
*  BY  HENRY P. COTTEN                          10 APRIL 1981     *
*      2465 FOUNTAINEBLEAU DRIVE                                  *
*      DORAVILLE, GEORGIA  30360                                  *
*                                                                 *
*        THIS PROGRAM WILL PUT A CURSOR ON THE SCREEN             *
*              FOR INTERACT 8K GRAPHICS BASIC                     *
*                   also for Level II Basic                       *
*******************************************************************
```

LOC.	COMMAND			DESCRIPTION
5DCE: C2 2C 60	JNZ		602C	JUMP IF KEYBOARD INPUT
5DD1: E5	PUSH	H		
5DD2: D5	PUSH	D		
5DD3: C5	PUSH	B		
5DD4: 2A 06 4C	LHLD		4C06	LOCATION OF CURSOR REGISTER
5DD7: EB	XCHG			PUT CURSOR X,Y INTO D
5DD8: D5	PUSH	D		SAVE REGISTER FOR LATER
5DD9: 0E 01	MVI	C	01	CURSOR TYPE (01=FULL BLOCK)
				(5F=UNDERLINE)
5DDB: CD 5B 05	CALL		055B	DISPLAY CURSOR ON SCREEN
5DDE: 0E 00	MVI	C	00	
5DE0: CD 2E 06	CALL		062E	CHANGE CHARACTER COLOR TO BACKGROUND
5DE3: 01 54 01	LXI	B	0154	ON TIME CONSTANT
5DE6: CD F6 07	CALL		07F6	DELAY
5DE9: D1	POP	D		RECALL CURSOR LOCATION
5DEA: 0E 01	MVI	C	01	CURSOR TYPE (SEE 5DD9 ABOVE)
5DEC: CD 5B 05	CALL		055B	DISPLAY CURSOR IN BACKGROUND COLOR
5DEF: 0E 03	MVI	C	03	
5DF1: CD 2E 06	CALL		062E	SET CHARACTER COLOR BACK TO NORMAL
5DF4: 01 00 01	LXI	B	0100	OFF TIME CONSTANT
5DF7: CD F6 07	CALL		07F6	DELAY
5DFA: C1	POP	B		
5DFB: D1	POP	D		
5DFC: E1	POP	H		
5DFD: C3 25 60	JMP		6025	JUMP BACK TO KEYBOARD LOOP

```
*****************  ADDITIONAL CODE REQUIRED  **********************
*                                                                 *
* THIS IS IN THE 8K BASIC PROGRAM -----  DO NOT POKE IN VALUES    *
* BECAUSE OF DESTROYING THE LOOP BEFORE YOU ARE FINISHED !        *
*                                                                 *
* EITHER USE A MONITOR PROGRAM THAT CAN WRITE TO A LOCATION       *
* OR FOLLOW THE SHORT PROGRAM THAT FOLLOWS                        *
*                                                                 *
*******************************************************************
```

LOC.	COMMAND		DESCRIPTION
6029: C3 CE 5D	JMP	5DCE	JUMP TO CURSOR SUBROUTINE
602C	.....		(CONTINUATION OF 8K BASIC PROGRAM)

Cursor, cont.

```
************  IF YOU DO NOT HAVE A MONITOR PROGRAM  ************
*                                                               *
*  POKE THIS PROGRAM IN --- THEN USE THE "USR" FUNCTION TO      *
*  RUN THE PROGRAM.  THIS WILL GET THE CODE IN WITHOUT HAVING   *
*  TO USE THE KEYBOARD.                                         *
*                                                               *
*****************************************************************
```

```
LOC.            COMMAND         DESCRIPTION

5D00:3E C3      MVI  A  C3      CODE FOR 'JMP'
5D02:32 29 60   STA     6029
5D05:21 CE 5D   LXI  H  5DCE
5D08:22 2A 60   SHLD    602A    STORE LOCATION FOR JMP
5D0B:C9         RET
```

```
***** NOW THAT EVERYTHING IS ENTERED - BUT IT DOES NOT WORK *****
*                                                               *
*   TO MAKE THE CURSOR START:  PRESS THE BACKSPACE KEY          *
*   (NOTE:  THIS WILL HAVE TO BE DONE EACH TIME AFTER A         *
*   RESET.)                                                     *
*                                                               *
*****************************************************************
```

```
*****************  ADDITIONAL PROGRAM NOTES  ********************
*                                                               *
*  THE ABOVE PROGRAM WAS WRITTEN SPECIFICALLY FOR THE 8K BASIC  *
*  FOUND IN THE INTERACT... BUT...                              *
*  THIS CURSOR PROGRAM CAN BE USED IN OTHER PROGRAMS, SUCH AS   *
*  A MONITOR, ETC.                                              *
*                                                               *
*  THREE THINGS ARE REQUIRED:                                   *
*    1)  KNOW WHERE THE CURSOR REGISTER IS LOCATED              *
*    2)  KNOW WHERE THE KEYBOARD CALL IS LOCATED                *
*    3)  USER DEFINED CHARACTER TABLE MUST BE USED              *
*                                                               *
*   THE USER DEFINED CHARACTER TABLE IN (3) IS A 7 BYTE TABLE   *
*   CONSISTING OF:                                              *
*    BYTE 1   05H - CHARACTER HEIGHT                            *
*         2   05H - CHARACTER WIDTH                             *
*         3   FFH                                               *
*         4   FFH                                               *
*         5   FFH                                               *
*         6   FFH                                               *
*         7   FFH                                               *
*  THE ADDRESS OF BYTE 1 OF THIS TABLE IS STORED IN THE ROM     *
*   STACK LOCATIONS 5FE1H-5FE2H IN LO-HI FORMAT.                *
*   IF THE PROGRAM HAS A BACKSPACE THAT ERASES THE KEYBOARD     *
*   CHARACTER, STEP (3) WILL NOT BE REQUIRED.                   *
*                                                               *
*  USUALLY, IF THE PROGRAM CONTAINS A CALL TO SUBROUTINE 07E0H  *
*  THEN THIS PROGRAM WILL WORK -- BUT THE FIRST AND LAST LINES  *
*  WILL HAVE TO BE CHANGED --- AND AN ADDITIONAL LINE WILL HAVE *
*  TO BE ADDED TO THE BEGINNING.                                *
```

Cursor, cont.

```
*                                                                    *
*  FOR EXAMPLE:                                                      *
*  ASSUME A PROGRAM HAS A CALL TO 07E0H AT LINE 4801H ----           *
*                                                                    *
*  CHANGE LINE 4801 TO READ:                                         *
*    4801   CALL 5DCD                                                *
*                                                                    *
*  MODIFY THE SUBROUTINE BY ADDING AND CHANGING AS FOLLOWS:          *
*    5DCD:CD E7 07   CALL 07E7    CALLS KEYBOARD CONTINOUSLY         *
*    5DCE:C0         RNZ          RETURN IF KEYBOARD IS PUSHED       *
*    ..........                   (CONTINUATION OF CURSOR            *
*    ..........                    PROGRAM ........)                 *
*    5DFD:C3 CD 5D   JMP  5DCD    JUMP BACK TO BEGINNING OF LOOP     *
*                                                                    *
**********************************************************************

**********************  ONE LAST NOTE  ********************************
*                                                                    *
*  THE TIMING LOOPS AT 5DE3H-5DE6H AND 5DF4H-5DF7H ARE NOT           *
*  ABSOLUTELY NECESSARY.  THE SYNC BETWEEN THE T.V. RASTER AND       *
*  THE DISPLAY OF THE CURSOR IS NOT EXACTLY THE SAME.  AS A          *
*  RESULT, THE CURSOR WILL EXHIBIT CHANGES IN COLOR DEPENDING        *
*  ON THE ON/OFF TIME CONSTANTS @ 5DE4H-5DE5H AND @ 5DF5H-5DF6H.     *
*  BY CHANGING THE VALUES, INTERESTING FLASHING PATTERNS CAN BE      *
*  OBTAINED---BUT BE CAREFULL --- TOO LONG A TIME CONSTANT WILL      *
*  SERIOUSLY SLOW THE LOOP CAUSING SOME KEYBOARD INPUT DATA TO       *
*  BE LOST.                                                          *
**********************************************************************
```

SCROLLING SIDEWAYS

by Terry Wormington 340 Cinnamon Dr. Satellite Beach, FL 32937

For those who missed the MicroVideo Interact newsletter, there was a sideways scroll technique given. It involved the following two instructions: POKE 2488, 1 (normally 32) and POKE 24864, 1 (normally 6).

With every successive print instruction, the entire screen shifts to the left one pixel. However, the message wraps around upon itself. I haven't done this yet, but it appears that outputting a CHR$(1) at an "off screen" location for that line would kill the wrap around. ("off screen" is beyond X=112). What the MicroVideo newsletter did not do was tell us what we were doeing with the POKEs above. With a little playing it becomes obvious:

LOCATION 24888 This is the value to move each video word by, 32 words being the number of locations to move a line one screen pixel. If a value not a multiple of 32 is entered here, the screen will scroll partially sideways if the value is less than 32 and will move partially up and sideways, that is diagonally, when the value is greater than 32.

SCROLLING SIDEWAYS cont.

LOCATION 24864 Each video memory is normally moved by 32 locations . This corresponds to one screen pixel. To clear the bottom line, this must be done 6 times. Thus this location contains the count value for the number of times to move the text during a scroll.

Some interesting effects can be obtained by modifying these values.

Double space scroll	: POKE 24888, 64 : POKE 24864, 6
Double speed single space scroll	: POKE 24888, 64 : POKE 24864, 3
Triple speed single space scroll	: POKE 24888, 96 : POKE 24864, 2

I often use the triple speed It is difficult to read listings at this speed - it's more like a rapid "skim" . A few words per line is about all one can recognize at this speed. However, it is handy for filling the screen in a hurry when an input is required from the user, the INPUT command (or an A$=INSTR$(1) "to continue") is used to hold the program once the screen is filled. It is possible to POKE 192 at 24888 and 1 at 24864 to move a full line at once but this causes some machine code to creep into the screen memory. It doesn't harm anything but merely makes reading the screen harder to read.

JOTTO

by Ruth Ann Halpern 1 Tamarack RD. Natick, MA 01760

Jotto is usually played by two people with paper and pencil. Here you let the computer chose a word and you guess it. You'll still need pencil and paper to keep track of letters as you eliminate them. Directions for play are in the program. The list of words are read into "packed" arrays from a data tape. Each data tape contains 100 possible 5 letter words for the computer to chose from. They are read-in in "packed" form to conserve memory and loading time.

The second listing is the program used to prepare the data tape (transform a string array into a numerical array). To produce other data tapes just redo all the data statements, each data statement consists of ten 5 letter words, no commas, no spaces. Note this program contains no prompt so be sure to advance tape or put in fresh tape and press read and write before typing RUN.

```
5 CLS
10 CLEAR(750)
20 DIMW$(9),W(49)
30 PRINT"DO YOU NEED       INSTRUCTIONS Y/N":A$=INSTR$(1)
40 IFA$="Y"THENGOSUB500
45 CLS
50 PRINT"INSERT WORD TAPE PRESS READ THEN  ANY LETTER":A$=INSTR$(1)
60 FORI=0TO9
70 CLOAD*W
80 FORJ=0TO49
100 W$(I)=W$(I)+CHR$(W(J))
110 NEXT:NEXT
120 T=0:PT=0
130 T$=W$(INT(RND(1)*10))
140 I=(INT(RND(1)*10)+1)*5-4
```

Jotto, cont.

```
150 TW$=MID$(T$,I,5)
155 CLS
160 PRINT"YOUR 5 LETTER    GUESS PLEASE":PRINT
165 INPUTG$
170 C$=TW$
180 P=0
190 IFG$="I QUIT"GOTO399
220 IFLEN(G$)<>5THENPRINT"INCORRECT INPUT":GOSUB750:GOTO155
230 T=T+1
240 IFG$=TW$THENPT=PT+10:GOTO320
250 FORI=1TO5
260 FORJ=1TO5
270 IFMID$(G$,I,1)=MID$(C$,J,1)THENGOSUB960
280 NEXTJ
290 NEXTI
294 CLS
295 IFP=5THENP=10
300 PT=PT+P
310 PRINTP;"POINTS-TURN";T:GOSUB750:GOTO155
320 M=INT(RND(1)*5)+1
330 ONMGOSUB800,810,820,830,840
340 IFT<14THENB=105-T*5
350 IFT>13ANDT<17THENB=40
360 IFT>16ANDT<21THENB=35
370 IFT>20THENB=0
380 PRINT:PRINT"YOU SCORED";PT;" AND A BONUS OF";B;"TOTAL POINTS";PT+B
390 GOSUB750:GOTO430
399 CLS
400 M=INT(5*RND(1))+1
410 ONMGOSUB900,910,920,930,940
420 PRINT:PRINT"MY WORD WAS ":PRINTTW$:GOSUB750
430 CLS:PRINT"WANT TO PLAY     AGAIN Y/N?"
440 A$=INSTR$(1)
450 IFA$="Y"THENGOTO120
460 END
500 CLS
510 PRINT"I WILL PICK A 5  LETTER WORD AND  YOU WILL TRY TO  GUESS IT":PRINT
515 GOSUB700
520 PRINT"EACH TIME YOU    GUESS I WILL TELLYOU HOW MANY     LETTERS YOU GOT"
525 PRINT"RIGHT."
530 GOSUB700
540 PRINT"IF A LETTER      APPEARS IN YOUR  WORD ONLY ONCE   BUT 2 OR MORE"
545 PRINT"TIMES IN MINE YOUWILL GET ONLY ONEPOINT AND VICE   VERSA"
550 GOSUB700
560 PRINT"EACH POINT MEANS YOU GOT A LETTER CORRECT"
561 PRINT"IF YOU GET 10    POINTS YOU GOT   ALL 5 LETTERS"
565 PRINT"BUT NOT THE WORD"
570 GOSUB700
580 PRINT"WHEN YOU WIN YOURSCORE IS YOUR    TOTAL NUMBER OF  POINTS +A BONUS"
583 PRINT"REPRESENTING THE NUMBER OF TURNS  YOU TOOK"
585 GOSUB700
590 PRINT"ANY TIME YOU     WANT TO STOP AND SEE MY WORD JUST TYPE 'I QUIT'"
600 PRINT:PRINT"WHEN YOU'RE READYTO PLAY":GOSUB700
610 RETURN
700 PRINT:PRINT"PRESS ANY KEY TO CONTINUE":A$=INSTR$(1):CLS:RETURN
750 FORX=1TO1000:NEXT:RETURN
```

Jotto, cont.

```
800 PRINT"GEE YOU'RE SMART":RETURN
810 PRINT"LUCKY GUESS":RETURN
820 PRINT"HOW DID YOU KNOW":RETURN
830 PRINT"YOU MUST HAVE    READ MY MIND":RETURN
840 PRINT"HUMPH!!!!!!":PRINT"I THOUGHT I HAD  YOU THAT TIME":RETURN
900 PRINT"NOT SO SMART ARE YOU":RETURN
910 PRINT"HA!HA! I STUMPED YOU":RETURN
920 PRINT"I REALLY KNOW    SOME NEAT WORDS":RETURN
930 PRINT"THOUGHT YOU HAD  ME HUH? WELL YOU DIDN'T YOU BIG   DUMMY!!!":RETURN
940 PRINT"YOU LOSE! I WIN!!THE COMPUTER     TRIUMPHS AGAIN!! DUMDUM!!":RETURN
960 C$=LEFT$(C$,J-1)+" "+RIGHT$(C$,5-J)
970 G$=LEFT$(G$,I-1)+"*"+RIGHT$(G$,5-I)
980 P=P+1:RETURN
Ok
```

```
LIST

10 CLEAR(1000)
20 DIMW$(9),W(49)
30 FORI=0TO9
40 READ W$(I)
50 GOSUB200
60 CSAVE*W
70 NEXT
195 END
200 FORJ=1TO50
210 L$=MID$(W$(I),J,1)
215 JJ=J-1
220 W(JJ)=ASC(L$)
230 NEXT
240 RETURN
400 DATA IGLOOAGILEDEFERBASICEQUIPFALSENICHECARRYOBESELADLE
410 DATA IDIOMPHOTOKOALABOARDOPTICQUAFFGAFFERADARMAIZESCALP
420 DATA USUALJAUNTVALVEROOFSSTUMPJOISTOFFALROOSTTACKYAGREE
430 DATA DEPTHFILLYNIECEBATTYEBONYLAUGHKAYAKPROOFCHALKTILTH
440 DATA IDYLLHANDYUNCLEGAUGEWEEPYTUNICMANNAYUCCAUNZIPPSYCH
460 DATA UMBRAGENIEYACHTKNEELREFERNYMPHZEBRAMEDIADROLLQUEUE
470 DATA WHISHBLOOMSAVVYENSUEFISHYYODELOTTERTIGERSWATHGIDDY
480 DATA WAFERVELDTXERICHAPPYXYLEMVIRUSTHONGIMPLYGNARLJAZZY
490 DATA PEEVEQUOITHAIKULUNARCIRCANOOSEINGOTPOOCHDUSKYKARAT
500 DATA HYDRAMUFFSALIBIFIXEDEPOXYCANDYOPERALAPSEJUNKSALBUM
Ok
```

* * *

BASIC MONITOR

by Walt Hendrickson 2313 W. 181st Torrance, CA 90504

Here is a short BASIC program your readers might find useful. It is a MONITOR program written in BASIC which allows the programmer to ENTER, DUMP, and EXECUTE machine-language routines. It can be used to create machine-language subroutines called from BASIC, to create stand-alone routines, to examine the Interact Rom routines, or to examine and modify Basic itself. The monitor has four commands: DUMP, ENTER, EXECUTE, and BASIC.

COMMAND	ACTION
BASIC	Exits the Monitor and Returns control to the Basic interperter.
DUMP	Dumps data from memory in HEX and ASCII format, four bytes per line. The display will pause after each three lines of data. To continue the listing, type a 'C', to exit dump mode press any other key after it has halted.
ENTER	Examine and change memory data. The display will show the address (in HEX), then the data (in HEX) in that address. If you do NOT want to change the data, press the CR key. The next address and data will then be displayed. To CHANGE the data in an address, type the data (two HEX characters), then press the CR Key. The new data will be entered and the next address and data will be displayed. To Exit the ENTER mode, type a '/' (divide key) followed by a CR.
EXECUTE	Jumps to the entered address and Executes the code found there. If a RET instruction (C9H) is placed at the end of your code, control will return to the MONITOR Program.

LIST

```
10 CLS
20 PRINT"  BASIC MONITOR"
30 PRINT" BY W.HENDRICKSON"
40 PRINT"   (C) 10/8/80"
50 FOR I=1 TO 1000:NEXT
60 :
70 DIM A$(16)
80 A$="0123456789ABCDEF"
90 POKE 19215,25
100 POKE 30462,195:POKE 30463,25:POKE 30464,119
110 CLS
120 :
```

Monitor, cont.

```
130 REM -COMMAND LOOP-
140 :
150 PRINT:PRINT"SELECT: DUMP"
160 PRINT"ENTER, EXECUTE"
170 PRINT"OR BASIC"
180 INPUT C$
190 IF LEFT$(C$,2)="DU" THEN 270
200 IF LEFT$(C$,2)="EN" THEN 630
210 IF LEFT$(C$,2)="EX" THEN 770
220 IF LEFT$(C$,2)="BA" THEN END
230 GOTO150
240 :
250 REM -DUMP-
260 :
270 INPUT"START(HEX)=";H$(1)
280 INPUT"END(HEX)=";H$(2)
290 PRINT
300 FOR I= 1 TO 2
310 H$=H$(I)
320 GOSUB 890
330 D(I)=D
340 NEXT I
350 K=1
360 FOR I=D(1) TO D(2) STEP 4
370 D=I
380 GOSUB1020
390 PRINT MID$(A$,H4+1,1);MID$(A$,H3+1,1);
400 PRINT MID$(A$,H2+1,1);MID$(A$,H1+1,1);"  ";
410 FOR J=0 TO 3
420 D=PEEK(I+J)
430 GOSUB1020
440 PRINT MID$(A$,H2+1,1);MID$(A$,H1+1,1);" ";
450 NEXT J:PRINT
460 :
470 FOR J=0 TO 3
480 D=PEEK(I+J)
490 IF D<32 OR D>127 THEN PRINT TAB(6);"    ";:GOTO 510
500 PRINT TAB(6);CHR$(D);"  ";
510 NEXT J
520 PRINT
530 K=K+1:IF K/4=INT(K/4) THEN 560
540 NEXT I
550 GOTO 150
560 K=1:P$=INSTR$(1)
570 IF LEFT$(P$,1)<>"C" THEN 150
580 NEXT I
590 GOTO 150
600 :
610 REM -ENTER-
620 :
630 PRINT:INPUT"START(HEX)=";H$
640 GOSUB 890:A=D
650 GOSUB1020
660 PRINT MID$(A$,H4+1,1);MID$(A$,H3+1,1);
670 PRINT MID$(A$,H2+1,1);MID$(A$,H1+1,1);"  ";
680 D=PEEK(A):GOSUB1020
```

Monitor, cont.

```
690 PRINT MID$(A$,H2+1,1);MID$(A$,H1+1,1);" ";
700 H$="":INPUT H$:IF H$="" THEN 730
710 IF H$="/" THEN 150
720 GOSUB 890:POKE A,D
730 A=A+1:D=A:GOTO 650
740 :
750 REM -EXECUTE-
760 :
770 PRINT:PRINT"EXECUTE ADD"
780 INPUT"IN HEX=";H1$
790 FOR I=2 TO 4 STEP 2
800 H$=MID$(H1$,I-1,2)
810 GOSUB 890
820 POKE(19475-I/2),D
830 NEXT I
840 Z=USR(0)
850 PRINT:GOTO150
860 :
870 REM -HEX TO DECIMAL-
880 :
890 D=0:RESTORE
900 Z=LEN(H$)
910 FOR K=Z TO 1 STEP -1
920 READ M
930 FOR J=1 TO 16
940 IF MID$(H$,K,1)=MID$(A$,J,1) THEN X=J-1:J=16
950 NEXT J
960 D=D+X*M
970 NEXT K
980 RETURN
990 :
1000 REM -DECIMAL TO HEX-
1010 :
1020 H4=INT(D/4096)
1030 H3=INT((D-H4*4096)/256)
1040 H2=INT((D-((H4*4096)+(H3*256)))/16)
1050 H1=D-((H4*4096)+(H3*256)+(H2*16))
1060 RETURN
1070 END
1080 DATA 1,16,256,4096
```

DOTS

by Kevin VenTrook 12813 Westpark Houston, TX 77082

A connect the dots game. This is a downgrade from a "smarter" program which was too slow. Requires joysicks (left only for play against the computer). The bit pattern in the upper right screen was a trick I used to save memory when the program was smarter and can be removed, but I haven't had time to do it.

To move: Use joystick to move flashing dot to one end of the line you wish to make. Press the fire button and move the joystick in the direction you wish the line to be drawn. Illegal moves result in the dot returning to the center of the board again.

Dots, cont.

```
 5 POKE19215,25
 10 CLS:COLOR7,0,1,6:PRINT" WELCOME TO DOTS":PRINT:PRINT:PRINT:PRINT
 20 DEFFNV(X)=16980+I+32*J:DEFFNMV(X)=17076+A+32*B
 30 DEFFNA(X)=I*C8+C2:DEFFNB(X)=J*C8+C2
 35 C1=0:C2=23:C3=3:C4=4:C5=35:C6=1:C8=8
 40 CS=0:PS=0:PRINT"TYPE YOUR INITIAL":N$=INSTR$(1)
 50 PRINT"DO YOU WISH TO   GO FIRST?":A$=INSTR$(1):CLS
 60 FORI=C1TO5:FORJ=C1TO5:PLOTI*C8+19,J*C8+19,C6:NEXTJ:NEXTI
 70 GOSUB5000:IFA$="Y"THEN1000
 80 OUTPUT"MY TURN",19,67,C3
 90 FORI=C1TOC4:FORJ=C1TOC4:IFPEEK(FNV(X))=C3THEN4000
 100 NEXTJ:NEXTI
 130 GOTO2000
 140 OUTPUT"MY TURN",19,67,C1:GOTO1000
 150 CLS:PRINT"YOU SCORED";PS:PRINT"BOXES":PRINT"I SCORED";CS:PRINT"BOX
ES":IFCS>PSTHEN170
 160 PRINT"I GUESS YOU WIN. JUST LUCKY":GOTO180
 170 PRINT"HA!HA! I WON!"
 180 PRINT"ONE MORE TIME?":IFINSTR$(1)="Y"THEN40
 190 END
 1000 OUTPUT"YOUR TURN",19,67,2:A=C5:B=C5:T=C1
 1010 M=JOY(0):IFM=C4THENB=B+C8
 1020 IFM=C8THENB=B-C8
 1030 IFM=C6THENA=A-C8
 1040 IFM=2THENA=A+C8
 1050 IFA<19ORA>59THEN1000
 1060 IFB<19ORB>59THEN1000
 1080 PLOTA,B,C1
 1090 FORK=0TO40:NEXTK
 1100 PLOTA,B,C6
 1110 IFFIRE(C1)=C1THEN1130
 1120 GOTO1010
 1130 M=JOY(C1)
 1140 IFM=C4ORM=C8THEN1170
 1150 IFM=C6ORM=2THEN1260
 1160 GOTO1130
 1170 D=SGN(5-M):IFB+D>59ORB+D<19THEN1000
 1180 IFPOINT(A,B+D)=C3THEN1000
 1190 FORI=B+DTOB+D*7STEPD:PLOTA,I,C3:NEXTI
 1200 K=(A-19)/C8:J=(B+D*C4-C2)/C8
 1210 FORM=C1TOC6:I=K-M:IFI<C1ORI>C4THEN1240
 1220 POKEFNV(X),PEEK(FNV(X))+1:IFPEEK(FNV(X))<C4THEN1240
 1230 PS=PS+1:OUTPUTN$,I*C8+21,J*C8+25,2:T=C6
 1240 NEXTM
 1250 GOTO1340
 1260 D=SGN(M-1.5):IFA+D>59ORA+D<19THEN1000
 1270 IFPOINT(A+D,B)=C3THEN1000
 1280 FORI=A+DTOA+D*7STEPD:PLOTI,B,C3:NEXTI
 1290 I=(A+D*C4-C2)/C8:K=(B-19)/C8
 1300 FORM=C1TOC6:J=K-M:IFJ<C1ORJ>C4THEN1330
 1310 POKEFNV(X),PEEK(FNV(X))+1:IFPEEK(FNV(X))<C4THEN1330
 1320 PS=PS+1:OUTPUTN$,I*C8+21,J*C8+25,2:T=C6
 1330 NEXTM
 1340 IFCS+PS=25THEN150
 1350 GOSUB5000:IFT=C6THEN1000
 1360 OUTPUT"YOUR TURN",19,67,C1:GOTO80
```

Dots, cont.

```
2000 FORA=C1TO25
2010 I=INT(RND(1)*4.5):J=INT(RND(1)*4.5):IFPEEK(FNV(X))<2THENGOSUB2080
2020 NEXTA
2030 FORI=C1TOC4
2040 FORJ=C1TOC4
2050 IFPEEK(FNV(X))<2THENGOSUB2080
2060 NEXTJ:NEXTI
2070 GOTO3000
2080 IFPEEK(FNV(X)+1)>C6ORPOINT(FNA(X)+C4,FNB(X))=C3THEN2110
2090 FORB=FNB(X)-C3TOFNB(X)+C3:PLOTFNA(X)+C4,B,C3:NEXTB
2100 GOTO2190
2110 IFPEEK(FNV(X)-1)>C6ORPOINT(FNA(X)-C4,FNB(X))=C3THEN2140
2120 FORB=FNB(X)-C3TOFNB(X)+C3:PLOTFNA(X)-C4,B,C3:NEXTB
2130 GOTO2190
2140 IFPEEK(FNV(X)+32)>C6ORPOINT(FNA(X),FNB(X)+C4)=C3THEN2170
2150 FORA=FNA(X)-C3TOFNA(X)+C3:PLOTA,FNB(X)+C4,C3:NEXTA
2160 GOTO2190
2170 IFPEEK(FNV(X)-32)>C6ORPOINT(FNA(X),FNB(X)-C4)=C3THENRETURN
2180 FORA=FNA(X)-C3TOFNA(X)+C3:PLOTA,FNB(X)-C4,C3:NEXTA
2190 GOSUB5000
2200 GOTO140
3000 I=INT(RND(1)*4.5):J=INT(RND(1)*4.5)
3010 IFPEEK(FNV(X))<C4THEN3030
3020 GOTO3000
3030 FORA=-4TOC4STEPC8
3040 IFPOINT(I*C8+C2+A,J*C8+C2)=C3THEN3060
3050 FORB=J*C8+20TOJ*C8+26:PLOTI*C8+C2+A,B,C3:NEXTB:GOTO2190
3060 IFPOINT(I*C8+C2,J*C8+C2+A)=C3THEN3080
3070 FORB=I*C8+20TOI*C8+26:PLOTB,J*C8+C2+A,C3:NEXTB:GOTO2190
3080 NEXTA
3090 GOTO3000
4000 FORI=C1TOC4:FORJ=C1TOC4
4010 IFPEEK(FNV(X))=C3THEN4040
4020 NEXTJ:NEXTI
4030 GOTO3000
4040 FORA=-4TOC4STEPC8
4050 IFPOINT(I*C8+C2+A,J*C8+C2)=C3THEN4070
4060 FORB=J*C8+20TOJ*C8+26:PLOTI*C8+C2+A,B,C3:NEXTB:GOTO4120
4070 IFPOINT(I*C8+C2,J*C8+C2+A)=C3THEN4090
4080 FORB=I*C8+20TOI*C8+26:PLOTB,J*C8+C2+A,C3:NEXTB:GOTO4140
4090 NEXTA
4100 OUTPUT"X",I*C8+21,J*C8+25,C3
4105 CS=CS+1:IFCS+PS=25THEN150
4110 GOSUB5000:GOTO90
4120 IFPEEK(FNV(X)+A/C4)<>C3THEN4100
4130 CS=CS+C6:OUTPUT"X",I*C8+21+2*A,J*C8+25,C3:GOTO4100
4140 IFPEEK(FNV(X)+A*C8)<>C3THEN4100
4150 CS=CS+C6:OUTPUT"X",I*C8+21,J*C8+25+2*A,C3:GOTO4100
5000 FORI=C1TOC4:FORJ=C1TOC4:POKEFNV(X),C1:NEXTJ:NEXTI
5030 FORI=C1TOC4:FORJ=C1TOC4
5050 A=FNA(X):B=FNB(X)
5060 FORK=-4TOC4STEPC8
5070 IFPOINT(A,B+K)=C3THENPOKEFNV(X),PEEK(FNV(X))+C6
5080 IFPOINT(A+K,B)=C3THENPOKEFNV(X),PEEK(FNV(X))+C6
5090 NEXTK:NEXTJ:NEXTI
5100 RETURN
Ok
```

C — MAJOR SCALE

by Jerry Goerz Kentucky Utilities Rm. 667 One Quality St.
Lexington, KY 40501

This program illustrates the general rule for predicting the first parameter of the tone generator:

next higher octave = 2X+8
next lower octave = (X-8)/2

and the general rule for constant tempo throughout the generator range:

Y = T / (X + 8) where T 1408 here

Notes: numbers circled are changed from listing in users manual
numbers "squared" are difficult compromises

```
LIST

10 CLS
20 PRINT"C-MAJOR SCALE BY"
30 PRINT"JERRY W. GOERZ"
200 T=50000
210 FORM=1TO56:READX
220 Y=T/(X+8)
230 TONEX,Y:NEXTM
240 RESTORE
250 DATA1400,1240,1104,1049,936,828,736,696
251 DATA696,616,548,521,464,410,364,344
252 DATA344,304,270,256,228,201,178,168
253 DATA168,148,131,124,110,97,85,80
254 DATA80,70,62,58,51,44,39,36
255 DATA36,31,27,25,21,18,15,14
256 DATA14,13,10,9,7,5,4,3
290 PRINT"REPLAY? Y/N"
291 A$=INSTR$(1):IFA$="Y"GOTO200
Ok
```

JULY — AUGUST DEADLINE

The deadline for advertisements and articles for the July-August issue is July 11. Free advertising space always available to subscribers. One of the reasons for the free advertising is because though there are over 400 Interaction subscribers, many people have sold only a few of their advertised programs. An ad in Interaction does not mean instant success but quality software will sell.

1980 INTERACTIONS STILL AVAILABLE

1980 issues are still available. Due to the demands for back issues, I have had additional copies of the 1980 Interaction Newsletters made. If you would like a set of these newsletter, please send $10. and I'll send you a set of the six newsletters from last year.

INTERACT OPERATING SYSTEM

IOS (Interact Operating System) has been improved! Two new commands have been included above those documented in the March/April INTERACTION. These new commands are: Verify memory, a checksum for a user specified range of addresses and Uncover address references, an 8080 address search & display routine. These 2 new commands replaced 2 of the 5 user commands previously advertised. Also included is an <u>option</u> for a user subroutine to be called each time a character is printed on the screen. This opens all display type IOS commands to RS232 port operations. The multiple breakpoint feature allows for locating and defusing any program bombs searching for the existence of a 2nd ROM like the IOS. Also included is a backspace & complete command cancellation option.

PRICE: $45.00 (postpaid), commented listing available
to IOS purchasers only for $10.00 (+$1.00 postage)

Send to: RICHARD A. FERRIS, 2041 San Sebastian Ct. #66, Houston Tx.
(zip) 77058 HAPPY PROGRAMMING!!!
(phone) (713) 333-2264 most evenings

LEVEL II BASIC PROGRAMS

ROADRACER (Steer through random course using left joystick)
INCOME TAX RECORDING (Enter income & deductions using data statements)
HORSE (Up to 8 players) Send $5 ea. or all three for $12 to
Walter Favorite 132 Peach St. Park Forest, IL 60466

SLAGH SYSTEM SERVICES
BOX 438
YPSILANTI
MI 48197

Slagh System Services Call-in Nite has been changed to Tuesdays.
Call (313) 434-0033 only between 7 and 11 PM EST for help or information
See our ad in the last issue of Interaction.

If you liked FASTLINE you'll love

THE LEONARDO GRAPHICS SYSTEM

This program extends the concept of an overlay to creation of a special-purpose graphics BASIC. Starting with Level II BASIC (or MV's newer 8K BASIC) a set of new graphics commands is inserted in place of string functions that are rarely used in a graphics context. (INSTR$ and CHR$, which are useful here have been retained.) Also the commands CSAVE* and CLOAD* have been replaced by commands that transfer pictorial information between the screen and tape.

The new commands are:

UDO - sets the coordinates of a user-defined origin.
POS - extended from Level II to also return the coordinates of the UDO.
PEN - sets a color for graphics output.
DOT - sets or clears a flag for dotted line output from LINE, MOVE, and ARC.
PLOT - extended from Level II to accept the three coordinate systems that are described below.
LINE - draws a line between any two points.
MOVE - draws line segments to connect a specified sequence of points.
TRI - draws a filled triangle with a specified set of vertices. May be used in combination to draw any straight-sided figure.
BOX - draws a rectangle.
ARC - draws an arc of a circle (including the circle as a special case).
SECT - draws a sector of a circle (including the filled circle as a special case).
CSAVE↑ - writes a tape from all or part of the screen.
CLOAD↑ - loads a tape made with CSAVE↑ .

All commands that specify points on the screen occur in three forms that accept different coordinate systems, e.g

PLOT X,Y - the usual BASIC coordinates.
PLOT! U,V - Cartesians relative to the UDO.
PLOT% L,A - Polar coordinates relative to the UDO.

Relative coordinates simplify writing subroutines to draw the same figure in different places. The polar routines avoid BASIC's slow floating-point SIN and COS functions. Instead they use LEONARDO's own fixed-point routines, which run many times faster (these may also be called from the user's machine-language subroutines).

The program does not reduce the amount of memory that is available for the user's BASIC program. (For graphics there is effectively an increase, because the programs are much more compact.)

With comprehensive documentation and programming examples, $20 postpaid (MI residents + 80¢ tax) from Harry Holloway, Box 2263, Ann Arbor, MI 48106.

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

JULY — SEPT., 1981 VOL II, NO.4

INTERACTION RETURNS

After a long summer's vacation, I've returned with a very late issue of INTERACTION. The reasons for its absense were my need for a break, the painting and remodeling of my house to make room for a new member of the family and the birth of that member, my son, Anthony on August 6th. There still will be the six issues this year. This is the fourth, the fifth should be done by early November and the sixth in December. Thank you to all my subscribers who've patiently awaited their newsletters.

RECONFIGURING BASIC FOR 32K

by Harry Holloway P.O Box 2263 Ann Arbor MI 48106

The new memory expansion for the Interact appears to be generating yet one more version of the Microsoft BASIC. While some modification is needed to permit use of the expanded memory, there is the potential disadvantage that major changes may invalidate a substantial body of user knowledge that has been developed for Level II BASIC. It is also likely that overlay programs for Level II will not work with the new version Thus there may be some benefit to making the minimum changes to Level II that will reconfigure it for a 32K Interact.

It turns out that the required modifications for Level II are quite trivial and that they will also work with the 8K Graphics BASIC, which is a minor variant. BASIC maintains a set of stored addresses (pointers) that keep track of memory usage. Most of the pointers are interdependent so that new addresses need to be inserted at only four locations. These are

The lowest address of the user's program space
- inserted at $6255\text{-}6_h$ and at $626D\text{-}E_h$

The preceeding address + 1 inserted at $6526\text{-}7_h$

The highest address of the user's program space
- inserted at $6258\text{-}9_h$

(In each case the lower-order byte is inserted at the lower memory location.)

RECONFIGURING BASIC FOR 32K, CONT.

After making these changes the modified BASIC needs to be reinitialised before it will run with the user's program in its new location. The most convenient method is to write on tape a copy of the reconfigured BASIC that will initialise after loading. This may be achieved by POKEing BASIC's tape output list and its initialisation flag.

The POKEs that follow were first tested on a 16K Interact by changing the user's program space from $4D21_h$ - $5FBF_h$ to 5000_h - $57FF_h$. The final change to 8000_h - $BFFF_h$ was very kindly tested on a 32K Interact by Ruth Ann Halpern, who also checked the relocation method for user programs that is desribed later.

For Level II the POKEs are

```
19215, 25 - enable PEEK/POKE
25173,  0
25174,128
25176,255
25177,191
25197,  0
25198,128
25894,  1
25895,128
25145,  0 - clear flag
19548,255 - change the tape output list
19549,  0
19550, 76
19551,  6
19552,  0
19553,  0
19554, 76
19569,255
19570,  0
19571, 96
19572,  0
19573, 32
19574,  0
19575, 96
```

After this the command CSAVE (no filename) will write the modified BASIC to tape. The copy will have a short, but adaquate, leader. Just be sure to advance the cassette beyond the plastic physical leader.

For the 8K Graphics BASIC replace the POKEs to 19549 through 19554 with

```
19549, 18
19550, 75
19551,244
19552,  0
19553, 18
19554, 75
```

While you are at it you might as well correct an error in the extended PLOT command (someone forgot to translate 77_d to $4D_h$ by POKEing 19255, 77 .

Old BASIC programs may also be reconfigured to use the upper 16K of RAM. The simplest approach is to use a monitor to load the BASIC program, make changes, and then rewrite to tape. A BASIC program tape has five data blocks:

(i) 2 bytes that load at $4C5A_h$ . . These are the address of the start of the BASIC code (low-order byte first). This address should be changed from $4D22_h$ to 8001_h .

RECONFIGURING BASIC FOR 32K, CONT.

(ii) 2 bytes that load at $4CFD_h$. These are the address of BASIC's variable table. This address should be increased by $8001_h - 4D22_h = 32DF_h$.

(iii) 5 bytes that load into the filename buffer at $4D10_h$. No change.

(iv) 1 byte that loads at $4D1D_h$. This is the number of characters in the filename (0 - 5). No change.

(v) The BASIC code that loads at $4D22_h$. The unchanged data should be rewritten to tape with the load address changed to 8001_h .

The fact that this procedure works is surprising, because each BASIC program line is stored in memory with the format

2 byte link address, 2 byte line number, program material, zero (used as end marker)

The link address points to the beginning of storage for the next program line. Thus, when the program is loaded into a different part of memory, all of the link addresses will be wrong. However, the day is saved by the CLOAD command which fixes up the links before returning control to the user.

INTERACT TAPE FORMAT

Reprinted with permission from THE HI-LO MONITOR documentation

The Interact reads tapes by timing the intervals between successive flux changes in a particular direction. A short interval is used for a zero bit and a larger interval for a one bit, A still longer interval is for a leader and for other purposes that are described later. The nominal timings for these intervals are approximately:

Zero 0.4 msec One 0.9 msec Leader 1.5 msec

Each byte of information is then specified by 8 consecutive intervals.

For a compact description of the Interact tape formats we need some definitions. The following are not standard terms, but there doesn't seem to be a widely accepted standard to adopt.

Tape record - The smallest of organized data. It may contain $1-100_H = 1-256_D$ bytes of information that will be loaded into consecutive locations in memory.

Tape block - A set of one or more tape records that contain data to be loaded into contiguous sections of memory.

Tape file - All of the information between the leader and an end-of- file marker.

INTERACT TAPE FORMAT, CONT.

The Interact tape record has the format

$$n\ data_1\ data_2\ \ldots\ data_n$$

where n is a single byte number that tells the computer how many bytes of data are following it. As a special case n= 00 actually gets interpeted as n=100 so the largest possible tape record is

$$\emptyset\emptyset\ data_1\ data_2\ \ldots\ data_{1\emptyset\emptyset}$$

The tape block comes in two varieties. First, there is the data block. This consists of a set of tape records that are used to load successive sections of memory. Most often, though not necessarily, a data block consists of a sequence of 100_H byte records followed by one shorter record that makes up the "odd" number of bytes. For example, a data block that contains 21_H bytes of data would usually have the form

$$\begin{array}{r} \emptyset\emptyset\ data_1\ data_2\ \ldots\ data_{1\emptyset\emptyset} \\ \emptyset\emptyset\ data_1\ data_2\ \ldots\ data_{1\emptyset\emptyset} \\ 1F\ \emptyset\emptyset\ data_1\ data_2\ \ldots\ data_{1F} \end{array}$$

The other kind of tape block is the fill block. This consists of just one single byte record

$$\emptyset 1\ data_1$$

The single byte is used to fill a sequence of consecutive memory locations. The principle application of this block is to clear the screen at the beginning of a tape load by filling video RAM with zeros.

Having defined the types of tape blocks we need a means for telling the computer which is which and what to do with the contents of each block. This is done by preceding each block with a special kind of record, the header record.

A header record always contains five bytes of data. The header records for data blocks and for fill blocks both have the format

$$\emptyset 5\ addr_l\ addr_h\ count_l\ count_h\ code$$

Here $addr_l$ and $addr_h$ are the low order and high order bytes of the address at which loading of memory is to start. The total number of consecutive memory locations to be loaded is specified by $count_l$ and $count_h$. The final byte is a code that specifies the type of block that follows. The code is FF for a data block and FE for a fill block.

The end of a tape file is signalled by a third kind of header record. The end record, which has the format

$$\emptyset 5\ junk\ junk\ junk\ junk\ FD$$

Here, the values of the bytes that are designated junk are ignored. Only the final code byte, FD, is significant. The end record is not followed by a tape block. Instead, Interact practice is to follow it with an end-of-file mark that consists of four intervals with the sequence

$$zero\ \ leader\ \ zero\ \ leader$$

However, the end-of-file mark should not be needed with a properly-read tape file.

INTERACT TAPE FORMAT, CONT.

Finally, some notes about uses of leader intervals that have not been described so far. The control ROM requires each tape record be preceded by at least one leader interval and will accept any larger number of leader intervals at this point. Interact practice is to precede each record with four leader intervals. A sequence of leader intervals is also inserted after a fill block to allow time for memory to be filled. Leader intervals may also be written after the end-of-file mark. This practice appears to be a nonfunctional relic of punched-tape formats.

ROTATE

By Ken Stuempges 6261 N. Joyce Ave., Milwaukee, Wis. 53224

The game of Rotate is played on a four by four board filled randomly with the letters A through P. In a sense it is like the little plastic games with sliding pieces.

The object of the game is to put the letters in alphabetical order. This is done by rotating groups of four letters clockwise one position. The four numbers to be rotated are specified by positioning the blinking cursor at the top left corner of the group. (The cursor is moved using the left joystick; up or down, left or right.) The fire button will cause the actual rotation. (The fire button must be held down until detected; approximately one second).

Game initialization time will vary, depending on how long it takes to compute the starting board. Occasionally it may take as long as 30 seconds. Special moves, called exchanges, are also allowed. The number of special moves is the skill level. An exchange is requested by positioning the cursor under the X, on the right of the board, and depressing the fire button. Two adjacent letters, in any given row, may be exchanged. The letters are specified by relative position, 1 through 16, i.e., 3,4 will switch the third and fourth letters in the top row. By entering 9,10; the first and second letters in the third row will be exchanged.

The number of exchanges remaining is printed on the bottom right of the screen, and a running total of moves is printed on the bottom left of the screen. Positioning the cursor to the left, under S, will call to end the game. The far right column of letters, and the bottom row of letters; are illegal moves. These will not add to the total moves, however. There is no limit to the number of moves allowed.

Caution! This game can become frustrating, especially if you do not have any exchanges left. A typical game will take at least 30 to 40 moves, depending on the skill level.

The original concept for the game appeared in "More Basic Computer Games"; copyright 1979, by Creative Computing. This version bears little resemblance to the original, thanks to the versality of the Interact.

ROTATE, CONT.

SAMPLE BOARD

BEFORE ROTATION

SELECT POSITION:

```
S              X
     F L E P
     A I H N
     M C K B
     O J G D
0              5
```

AFTER ROTATION

SELECT POSITION:

```
S              X
     A F E P
     I L H N
     M C K B
     O J G D
1              5
```

```
1 GOSUB600
2 CLS:SOUND3,24
5 OUTPUT"A STARTING BOARD",10,70,6
6 OUTPUT"IS NOW",40,45,5
7 OUTPUT"BEING COMPUTED",15,20,7
11 DIMB(16),B$(16)
140 FORI=1TO16:B$(I)="0":NEXTI:COLOR4,3,7,0
150 FORI=1TO16
160 T$=CHR$(INT(16*RND(1)+65))
165 FORJ=1TOI
170 IFB$(J)=T$THEN160
175 NEXTJ
180 B$(I)=T$:NEXTI
190 M=0:S=0
191 GOSUB400
200 WINDOW18:SOUND7,4096:PRINT" ";M;"          ";E
201 OUTPUT"SELECT POSITION:",10,75,6:OUTPUT"S",26,66,6:OUTPUT"X",86,66,6
202 GOSUB1000:IFI=0THEN330
205 IFI=-1THEN510
210 IFI=4ORI=8ORI=12THEN700
215 IFI>12THEN700
220 M=M+1:T$=B$(I)
230 B$(I)=B$(I+4):B$(I+4)=B$(I+5):B$(I+5)=B$(I+1):B$(I+1)=T$
240 GOSUB400
305 FORI=1TO16
310 IFCHR$(I+64)<>B$(I)THEN200
315 NEXTI
320 PRINT:PRINT"YOU ORDERED THE  BOARD IN";M;"MOVES.":M1=M1+M:G=G+1
325 PRINT CHR$(7):FORI=1TO15
330 PRINT"PLAY AGAIN (Y-N)?
331 C$=INSTR$(1)
332 IFC$="N"THENCOLOR1,0,0,7:GOTO550
333 CLS:GOSUB600:CLS:SOUND3,24:OUTPUT"A NEW BOARD",25,70,6
334 OUTPUT"IS NOW",40,45,5:OUTPUT"BEING COMPUTED",15,20,7:GOTO140
340 PRINT:PRINT"YOU PLAYED";G
341 PRINT"GAMES AND ORDEREDTHE BOARD IN AN  AVERAGE OF";M1/G
350 PRINT"MOVES PER GAME."PRINT:GOTO999
400 CLS:SOUND3,24:L=66:FORI=1TO13STEP4
410 OUTPUT B$(I),38,L,5:OUTPUT B$(I+1),50,L,5
411 OUTPUT B$(I+2),62,L,5:OUTPUT B$(I+3),74,L,5
420 L=L-12:NEXTI:RETURN
510 INPUT"EXCHANGE WHICH   TWO POSITIONS";X,Y
```

ROTATE, CONT.

```
515 IFX=0ORY=0THENPRINT"ILLEGAL MOVE!!":GOTO510
520 IFX<>Y+1ANDX<>Y-1THENPRINT"ILLEGAL. AGAIN...":GOTO510
530 E=E-1:IFE=-1THENPRINT"NO MORE SPECIALS!":FORX=1TO500:NEXT:E=0:GOTO200
540 T$=B$(X):B$(X)=B$(Y):B$(Y)=T$:GOTO240
550 CLS:OUTPUT"HOPE YOU HAD FUN",12,38,6
555 GOTO999
600 CLS:S=0
610 OUTPUT"INPUT SKILL LEVEL",5,46,6:OUTPUT"1 TO 9 - -",30,38,6
620 D$=INSTR$(1)
630 IFD$="0"THEN600
640 E=ASC(D$)-48:RETURN
700 PRINT:PRINT
702 OUTPUT"ILLEGAL MOVE",15,15,6:GOSUB1430
705 FORC=1TO8
710 COLOR3,5,2,3:SOUND3,16416
715 FORD=1TO10:NEXT
720 COLOR7,1,2,4
725 FORD=1TO10:NEXT
730 NEXTC
731 SOUND7,4096
735 COLOR4,3,0,7:PRINT:PRINT:GOTO200
999 END
1000 A=38:B=64
1010 OUTPUT"_",A,B,6:GOSUB1040
1015 IF FIRE(0)=0THENRETURN
1020 OUTPUT"_",A,B,0:GOSUB1040
1025 IFFIRE(0)=0THENRETURN
1030 GOTO1010
1040 FORX=1TO3
1050 IF JOY(0)=2 THEN 1110
1060 IFJOY(0)=1THEN1140
1070 IFJOY(0)=4THEN1170
1080 IFJOY(0)=8THEN1200
1090 IFFIRE(0)=0THEN1250
1100 NEXTX
1105 RETURN
1110 GOSUB1230
1120 A=A+12:IFA>87THENA=38
1130 RETURN
1140 GOSUB1230
1150 A=A-12:IFA<25THENA=74
1160 RETURN
1170 GOSUB1230
1180 B=B+12:IFB>70THENB=28
1190 RETURN
1200 GOSUB1230
1210 B=B-12:IFB<20THENB=64
1220 RETURN
1230 OUTPUT"_",A,B,0:TONE117,74:RETURN
1250 IFA=38ANDB=64THENI=1
1260 IFA=50ANDB=64THENI=2
1270 IFA=62ANDB=64THENI=3
1280 IFA=74ANDB=64THENI=4
1290 IFA=38ANDB=52THENI=5
1300 IFA=50ANDB=52THENI=6
1310 IFA=62ANDB=52THENI=7
1320 IFA=74ANDB=52THENI=8
```

ROTATE. CONT.

```
1330 IFA=38ANDB=40THENI=9
1340 IFA=50ANDB=40THENI=10
1350 IFA=62ANDB=40THENI=11
1360 IFA=74ANDB=40THENI=12
1370 IFA=38ANDB=28THENI=13
1380 IFA=50ANDB=28THENI=14
1390 IFA=62ANDB=28THENI=15
1400 IFA=74ANDB=28THENI=16
1410 IFA=26ANDB=64THENI=0
1420 IFA=86ANDB=64THENI=-1
1430 OUTPUT"_",A,B,0:RETURN
5000 S=0:COLOR3,0,4,0
5010 CLS:OUTPUT"INPUT SKILL LEVEL",4,45,6:OUTPUT"1 TO 9 - -",30,38,6
5020 D$=INSTR$(1)
5030 IFD$="0"THEN5000
5050 E=ASC(D$)-48:S=S+1:PRINTS
5055 IFS>ETHENPRINT"GOOD":FORX=1TO1000:NEXTX:GOTO5010
5060 IFS<ETHENPRINT"BAD":FORX=1TO1000:NEXT:GOTO5010
5070 GOTO5010
Ok
```

HELP!

A friend and I are ham radio operators. We have built a modem type unit with which we can send and receive messages using M.V. RS-232 port, along with their communications program. Using our VHF 2 meter radios, the copy is perfect. We can also run a printer using the RS-232 BASIC L list and L print commands. If one person puts in the communicator tape and the other person uses the RS-232 BASIC to send L list or L print information, the person using the communicator program can also copy that information on the screen, if the proper POKE statements for the 300 baud rate are used in RS-232 BASIC. The parity must be set to "odd" in the communicator program. We found that 110 and 150 baud rates from RS-232 BASIC would not work well. The 300 baud rate, however, works fine. I assume it has something to do with improper timing at lower speeds.

The problem is this. Does anyone out there know how to receive information with the RS-232 BASIC, either in BASIC or possibly a "USR" routine, using the RS-232 port so that we can load programs into memory? In other words , we want a way in which we can exchange programs between Interacts via the modems and still have full control of the computer, to then run the loaded program, or at least keep it in memory, so we can review the information or CSAVE it on the Interact's tape recorder.

I am sorry to say that we have no knowledge of assembly language. We hope that someone out there can help us. Wouldn't it be great to run one Interact with another, or exchange programs over the phone or via radio with the modems?

If you can help us, please contact me. Thank you,

WB8REH James Theisen R. 5 4692 Theisen Rd. Gaylord, MI 49735

ATTENTION HAMS! I work all bands. If you want to make a sked. on the air, please contact me.

FINANCIAL WIZARD

by Jerry Goerz KY Utilities Rm 667 One Quality St
Lexington, KY 40507

This program will calculate a number of financial functions. They are Inflation of salary or goods and the rate of increase, Accumulated earnings, Savings Account, Retirement Fund and Annuity Plan. The menu and prompts provide all the instructions needed.

```
LIST

10 CLS
20 PRINT"JERRY W. GOERZ'S FINANCIAL WIZARD"
30 PRINT:PRINT"   SYSTEM MENU:"
40 PRINT"1=INFLATION OF     SALARY OR GOODS"
50 PRINT"2=RATE OF":PRINT"  INCREASE"
60 PRINT"3=ACCUMULATED      EARNINGS"
70 PRINT"4=CONTINUE MENU"
80 PRINT"8=QUIT"
90 PRINT:PRINT"PRESS 1,2,3,4,8"
100 A$=INSTR$(1)
110 IFA$="1"GOTO300
120 IFA$="2"GOTO500
130 IFA$="3"GOTO700
140 IFA$="8"GOTO1500
150 PRINT:PRINT"5=SAVINGS ACCOUNT"
160 PRINT"6=RETIREMENT FUND"
170 PRINT"7=ANNUITY PLAN"
180 PRINT"8=QUIT"
190 PRINT:PRINT"PRESS 5,6,7,OR 8"
200 A$=INSTR$(1)
210 IFA$="5"GOTO900
220 IFA$="6"GOTO1100
230 IFA$="7"GOTO1300
240 GOTO1500
300 CLS
310 INPUT"PRESENT SALARY   OR COST";Z
320 INPUT"ANNUAL % INCREASE";I
330 INPUT"HOW MANY YEARS    FROM NOW";N
340 W=Z*(1+I/100)^N
350 PRINT:PRINT"SALARY OR COST";N;"YEARS FROM NOW"
360 PRINT"WILL BE";W:PRINT
370 PRINT"DIFFERENT YEAR? Y/N"
380 A$=INSTR$(1):IFA$="Y"GOTO330
390 PRINT"SIMILAR PROBLEM? Y/N"
400 A$=INSTR$(1):IFA$="Y"GOTO300
410 GOTO30
500 CLS
510 PRINT"PRESENT SALARY OR"
520 INPUT"COST";W
530 PRINT"STARTING SALARY"
540 INPUT"OR COST";Z
550 INPUT"NUMBER OF YEARS";N
560 I=(((W/Z)^(1/N))-1)*100
570 PRINT:PRINT"RATE OF INCREASE=";I;"%"
```

FINANCIAL WIZARD, CONT.

```
580 PRINT:PRINT"ANOTHER? Y/N"
590 A$=INSTR$(1):IFA$="Y"GOTO500
600 GOTO30
700 CLS
710 INPUT"STARTING SALARY  ";Z
720 PRINT"CONTINUOUS":PRINT"(NOT STEP)"
730 INPUT"ANNUAL % INCREASE";I:I=I/100
740 INPUT"NUMBER OF YEARS  ";N
750 E=ABS(Z*((((1+I)^N)-1)/LOG(1+I)))
760 PRINT:PRINT"ACCUMULATED":PRINT"EARNINGS=":PRINTE
770 PRINT:PRINT"ANOTHER? Y/N"
780 A$=INSTR$(1):IFA$="Y"GOTO700
790 GOTO30
900 CLS
910 PRINT"PRINCIPAL":INPUT P
920 PRINT"COMPOUNDED HOW":PRINT"MANY TIMES EACH"
930 PRINT"YEAR? (NOT 0)":INPUT Q
940 INPUT"% INTEREST";R:R=R/100
950 INPUT"NUMBER OF YEARS  ";N
960 A=P*(1+R/Q)^(N*Q)
970 PRINT:PRINT"AMOUNT OF SAVINGS"
980 PRINT"AFTER";N;"YEARS":PRINT"IS";A
990 PRINT:PRINT"ANOTHER YEAR? Y/N"
1000 A$=INSTR$(1):IFA$="Y"GOTO950
1010 PRINT"SIMILAR PROBLEM? Y/N"
1020 A$=INSTR$(1):IFA$="Y"GOTO900
1030 GOTO30
1100 CLS
1110 INPUT"% ANNUAL INTEREST";R:R=R/100
1120 INPUT"NUMBER OF YEARS  ";N
1130 PRINT"FIRST END-OF-YEAR":INPUT"DEPOSIT";P
1140 PRINT"FIXED OR INCREAS-ING DEPOSIT? F/I"
1150 A$=INSTR$(1):IFA$="F"GOTO1220
1160 INPUT"% ANNUAL DEPOSIT  INCREASE";I:I=I/100
1170 A=0
1180 FORY=1TON
1190 A=P+A*(1+R):P=P*(1+I)
1200 NEXTY
1210 GOTO1230
1220 A=P*(((1+R)^N)-1)/R
1230 PRINT:PRINT"VALUE OF FUND":PRINT"AFTER";N;"YEARS"
1240 PRINT"IS";A
1250 PRINT:PRINT"ANOTHER? Y/N"
1260 A$=INSTR$(1):IFA$="Y"GOTO1100
1270 GOTO30
1300 CLS
1310 PRINT"DESIRED ANNUAL":INPUT"ANNUITY";A
1320 PRINT"% ANNUAL INTEREST"
1330 PRINT"ON UNUSED BALANCE"
1340 INPUTR:R=R/100
1350 PRINT"NUMBER OF YEARS"
1360 PRINT"OF PAYMENTS FROM":INPUT"PLAN";N
1370 P=A*(1-(1+R)^(-N))/R
1380 PRINT:PRINT"TO HAVE AN":PRINT"ANNUITY OF".
1390 PRINTA;"PER YEAR":PRINT"FOR";N;"YEARS,"
```

FINANCIAL WIZARD, CONT.

```
1400 PRINT"INVEST";P:PRINT"NOW!"
1410 PRINT:PRINT"ANOTHER? Y/N"
1420 A$=INSTR$(1):IFA$="Y"GOTO1300
1430 GOTO30
1500 CLS:END
Ok
```

GAME OF LIFE

by Ruth Ann Halpern 1 Tamarack Rd Natick, MA 01760

The game is adapted from the Game of Life, by J.H. Conway, Scientific American, Oct. 1970. It is a simulation of population dynamics, in which a number of organisms are placed in a matrix. The maximum number of neighbors any organism, in the center can have is 8. In each generation, if an organism has fewer than two neighbors, it will die of lonliness, if it has more than 3 it will die of overpopulation. Any time an empty space has exactly three living neighbors an organism will be born there. You may choose how many organisms you start with and how many generations (it could conceivably go on forever). Hint choose at least 25 generations since it can be very frustrating to have a terrific population going only to have it terminate because you had allocated too few generations.(you can always hit reset if it gets boring). Organisms are entered at the prompt by row, column (ie. 4,5). The matrix is 15x15 corresponding to 0-14. The program terminates either when your number of generations is reached, when the population dies off, or when you reach a stable situation. The program was written using Graphics Basic, those using Level II should either write a loop to plot 2 x 2 squares or drop the last two arguments in Line 940 and draw smaller squares. The game when running uses all but about 160 bytes of memory.

```
LIST

10 COLOR 0,4,1,7
20 CLS
30 OUTPUT"GAME OF LIFE",21,34,1
40 DIMNE$(14,14),OL$(14,14)
50 B$=" ":C$="*"
60 GE=1:AL=0
70 FORI=0TO14
80 FORJ=0TO14
90 NE$(I,J)=B$:OL$(I,J)=B$
100 NEXT:NEXT
110 CLS
120 PRINT"HOW MANY ORGANISMS WOULD YOU LIKE TO START WITH"
130 INPUT Y
140 CLS
150 PRINT"HOW MANY GENERATIONS DO YOU WANT TO TRY"
160 INPUTG:CLS
```

GAME OF LIFE, CONT.

```
200 CT=1
210 FORX=1TOY
220 PRINT"ENTER ORGANISM ";X
230 INPUT I,J
240 GOSUB1670
250 OL$(I,J)=C$
260 NEXT
270 GOSUB800
280 GOSUB890
290 FORGE=2TOG
300 CH=0:CT=1
310 BI=0:DE=0:SU=0:AL=0
320 SI=I1:LI=I2:SJ=J1:LJ=J2
330 IFSI=0THENSI=1
340 IFLI=14THENLI=13
350 IFSJ=0THENSJ=1
360 IFLJ=14THENLJ=13
370 FORI=SI-1TOLI+1
380 FORJ=SJ-1TOLJ+1
390 NB=0
400 IFI=0ANDJ=0THENGOSUB970
410 IFI=0ANDJ<>0ANDJ<>14THENGOSUB1010
420 IFI=0ANDJ=14THENGOSUB1070
430 IFI<>0ANDI<>14ANDJ=14THENGOSUB1110
440 IFI=14ANDJ=14THENGOSUB1170
450 IFI=14ANDJ<>14ANDJ<>0THENGOSUB1210
460 IFI=14ANDJ=0THENGOSUB1270
470 IFI<>0ANDI<>14ANDJ=0THENGOSUB1310
480 IFI<>0ANDI<>14ANDJ<>0ANDJ<>14THENGOSUB1370
490 IFOL$(I,J)=B$ANDNB=3THENNE$(I,J)=C$:CH=CH+1:BI=BI+1:GOSUB1670
500 IFOL$(I,J)=C$THENIFNB=3ORNB=2THENNE$(I,J)=C$:SU=SU+1:GOSUB1670
510 IFOL$(I,J)=C$THENIFNB>3ORNB<2THENNE$(I,J)=B$:CH=CH+1:DE=DE+1
520 IFOL$(I,J)=B$ANDNB<>3THENNE$(I,J)=B$
530 NEXT:NEXT
540 IFCH=0GOTO670
550 FORI=0TO14
560 FORJ=0TO14
570 OL$(I,J)=NE$(I,J)
580 NE$(I,J)=B$
590 NEXT:NEXT
600 GOSUB800
610 IFAL=0GOTO710
620 GOSUB890
630 NEXT
640 PRINT"YOU HAVE SUCCESSFULLY COMPLETED ";G;"GENERATIONS"
650 PRINT
660 GOTO740
670 CLS
680 PRINT"YOUR POPULATION IS STABLE AND WILL NOT CHANGE!"
690 PRINT
700 GOTO740
710 PRINT
720 PRINT"EVERYONE HAS DIED IN GENERATION ";GE
730 PRINT
740 PRINT"WOULD YOU LIKE TO PLAY AGAIN? TYPE YES OR NO"
750 A$=INSTR$(2)
```

GAME OF LIFE, CONT.

```
760 IFA$="YE"THENGOTO60
770 END
780 FORI=1TO1000:NEXT
790 RETURN
800 CLS
810 PRINT"GENERATION ";GE
820 PRINTAL;" ARE ALIVE"
830 IFGE=1GOTO870
840 PRINT"BIRTHS= ";BI
850 PRINT"DEATHS= ";DE
860 PRINT"SURVIVORS= ";SU
870 GOSUB780
880 RETURN
890 CLS
900 FORI=0TO14
910 FORJ=0TO14
920 IFOL$(I,J)=B$THENX=1
930 IFOL$(I,J)=C$THENX=2
940 PLOT28+4*J,67-4*I,X,2,2
950 NEXT:NEXT
960 RETURN
970 GOSUB1460
980 GOSUB1480
990 GOSUB1500
1000 RETURN
1010 GOSUB1460
1020 GOSUB1480
1030 GOSUB1500
1040 GOSUB1520
1050 GOSUB1540
1060 RETURN
1070 GOSUB1500
1080 GOSUB1520
1090 GOSUB1540
1100 RETURN
1110 GOSUB1500
1120 GOSUB1520
1130 GOSUB1540
1140 GOSUB1560
1150 GOSUB1580
1160 RETURN
1170 GOSUB1540
1180 GOSUB1560
1190 GOSUB1580
1200 RETURN
1210 GOSUB1600
1220 GOSUB1460
1230 GOSUB1540
1240 GOSUB1560
1250 GOSUB1580
1260 RETURN
1270 GOSUB1580
1280 GOSUB1600
1290 GOSUB1460
```

GAME OF LIFE, CONT.

```
1300 RETURN
1310 GOSUB1600
1320 GOSUB1460
1330 GOSUB1480
1340 GOSUB1500
1350 GOSUB1580
1360 RETURN
1370 GOSUB1460
1380 GOSUB1480
1390 GOSUB1500
1400 GOSUB1520
1410 GOSUB1540
1420 GOSUB1560
1430 GOSUB1580
1440 GOSUB1600
1450 RETURN
1460 IFOL$(I,J+1)=C$THENNB=NB+1
1470 RETURN
1480 IFOL$(I+1,J+1)=C$THENNB=NB+1
1490 RETURN
1500 IFOL$(I+1,J)=C$THENNB=NB+1
1510 RETURN
1520 IFOL$(I+1,J-1)=C$THENNB=NB+1
1530 RETURN
1540 IFOL$(I,J-1)=C$THENNB=NB+1
1550 RETURN
1560 IFOL$(I-1,J-1)=C$THENNB=NB+1
1570 RETURN
1580 IFOL$(I-1,J)=C$THENNB=NB+1
1590 RETURN
1600 IFOL$(I-1,J+1)=C$THENNB=NB+1
1610 RETURN
1620 IFI<I1THENI1=I
1630 IFI>I2THENI2=I
1640 IFJ<J1THENJ1=J
1650 IFJ>J2THENJ2=J
1660 AL=AL+1:RETURN
1670 IFCT=1THENI1=I:I2=I:J1=J:J2=J
1680 GOSUB1620
1690 CT=CT+1
1700 RETURN
Ok
```

* * *

The Interact Real Time Clock, Revisited

by Richard Pasco
235 College Ave.
Mountain View, Calif. 94040
(415) 968-9867

I have found that digital clocks based on my article "On Greater Accuracy for Interact Real Time Clock" [*Interaction*, Vol. 1, No. 5, October-November, 1980] still do not run as accurately as might be desired. This is because my section 1 assumed that video from the Interact conforms to the NTSC standards for broadcast video, which it does not. The integer in locations 24559-24560 is incremented at the vertical sweep rate, which isn't 60 Hz, but it isn't 59.94006 Hz, either. The following information, obtained from examination of the video signal and Interact schematics, corrects section 1 of that article.

> The color subcarrier is 315/88 = 3.579545 MHz.
> There are 228 cycles of the subcarrier for every horizontal sweep. [For NTSC it's 227.5]
> Thus, the horizontal sweep rate is 13125/836 = 15.699761 kHz. [For NTSC it's 15734.266]
> Thus, there are 262 lines per field. [For NTSC it's 262.5]
> Thus, the vertical sweep rate is 1640625/27379 = 59.92275 Hz. [For NTSC it's 59.94006]

The following listing is a digital clock program which is accurate within seconds per day. (The sources of error are crystal inaccuracies and floating point roundoff error.) For reasons described in my previous article, BASIC doesn't POKE into the Real Time Clock address; it only watches it.

```
LIST

1 REM "ACCLK" ACCURATE DIGITAL CLOCK FOR INTERACT
2 REM REVISED 3/23/81
3 REM COPYRIGHT (C) 1981 BY RICHARD PASCO
4 REM PERMISSION GIVEN TO REPRINT IN "INTERACTION"
9 POKE19215,25     :REM ENABLE POKES
10 CK=24559        :REM RTC ADDRESS
14 CF=3579545      :REM NTSC COLOR FREQ
16 HF=CF/228       :REM HORIZ SWEEP FREQ
18 VF=HF/262       :REM VERT SWEEP
19 RT=VF           :REM REAL-VALUED THRESHOLD
20 WINDOW 18
22 CLS
24 COLOR0,7,5,3
30 INPUT "HOUR";H
40 INPUT "MIN";M
50 INPUT "SEC";S
100 T=INT(RT)      :REM INTEGER THRESHOLD
120 IF ((PEEK(CK)-T) AND 255)<128 THEN 120
125 RT=RT+VF
140 S=S+1
150 IFS=60 THEN S=0:M=M+1
160 IF M=60 THEN M=0:H=H+1
170 IF H=13 THEN H=1
200 PRINTH;M;S
220 GOTO100
Ok
```

INTERACTION Newsletter

1980 INDEX

Compiled by:
Jerry Lerner

NOTE: I,2,3 = Vol. I, No. 2, Page 3

PRIME NUMBER GENERATOR

Jerry Lerner, 520 Illinois St., Park Forest, IL 60466

Prime numbers are integers such as 7,19,31,etc., that have no divisors except themselves and one. This BASIC program employs the "sieve" method to generate up to 1000 prime numbers, using the TV screen memory for both storage of candidate integers and simultaneous display of the prime-selection process. The integers up to 7920 are "mapped" on an area of 110x72 pixels, with the bottom line representing 0 through 109, the next line 110-219, etc. By changing the color of pixels corresponding to multiples of the consecutive primes 2,3,5,7,11,13,17,19,... we "cancel" these integers; survivors must be primes.

The variation in pattern as the calculation proceeds provides a graphic display that is both entertaining and instructive. Since storage of an array of 1000 numbers requires 4000 bytes and leaves very little room for programming, explanatory remarks are omitted, the initial primes are "bootstrapped"into the array to avoid data lists, and the program must be entered without imbedded blanks. Briefly, lines 10-50 are initializing, 60-90 cancel composite (non-prime) numbers, 100-130 locate the next prime, 150-190 transfer primes from the display into the array, and 200-280 provide options for listing, taping or exit. We cannot disturb the screen until all of the primes are stored, therefore the boundary between the cancellation and transfer phases is indicated with a tone signal.

Timing for the full 1000 primes is roughly 6 min. for cancellation and 3 min. for transfer; listing requires an additional one min. per hundred lines. For faster programs that generate fewer than 1000 primes, W and H in line 20 can be changed to give a smaller product; note, however, that W and H must not exceed 110 and 75 respectively. To find the first 100,200,300,.....900 primes substitute 32x17, 51x24, 71x28, 56x49, 76x47, 70x63, 96x55, 104x59, 100x70 for 112x72.

```
10 DIMP(1000):CLS
20 W=110:H=72
30 A=W*H:S=SQR(A):N=1:I=2
40 DX=(112-W)/2:DY=(77-H)/2
50 P(N)=I:N=N+1
60 B=I+I:IFI=2THENB=2
70 FORJ=I*ITOASTEPB
80 Y=INT(J/W):X=J-W*Y
90 PLOTX+DX,Y+DY,3:NEXTJ
100 I=I+1
110 Y=INT(I/W):X=I-W*Y
120 IFPOINT(X+DX,Y+DY)>0GOTO100
130 IFI<SGOTO50
140 TONE100,900
150 FORJ=P(N-1)+2TOASTEP2
160 Y=INT(J/W):X=J-W*Y
170 IFPOINT(X+DX,Y+DY)>0GOTO190
180 P(N)=J:N=N+1
190 NEXTJ:TONE50,900
200 PRINT:PRINT"TYPE LIST, TAPE  OR EXIT"
210 B$=INSTR$(4)
220 IFB$="LIST"GOTO260
230 IFB$="TAPE"GOTO280
240 IFB$="EXIT"THENSTOP
250 GOTO200
260 FORI=1TON-1:PRINTI,P(I)
270 NEXTI:GOTO200
280 CSAVE*P:GOTO200
Ok
```

8080 FOR EVERYONE

Now, everything you need to know about the language of the 8080 and the Interact ROM in one comprehensive book. If you have been struggling along trying to make some sense out of available books on Machine Language programming and ROM functions, this is for you! Included in the book are: (1) Introduction to the 8080 and its functions. What are bits, bytes and hex? What about registers and stacks? What about MOVE, JUMP, and CALL? (2) Complete listing in Hex Code, Mnemonic, and description of the 8080 Instructions. (3) A complete listing of the Interact ROM with Hex Code, Mnemonic and description. (4) Discussion of the ROM Routines and functions with some Byte-Saver tips on their use and function. Make your own set of characters with any dot matrix. Make graphics, tones and sound. (5) Several original Routines to show the power of this Language and to save you memory.

Send $10.00 for the book 8080 FOR EVERYONE to:

George A. Leggett 52895 Bunker Hill New Baltimore, MI 48047

FULL ASCII Character Set in Terminal Emulator Program for Interact with Slash RS-232 Port. Sends upper- and lower- case using "SHIFT", accepts hex code for others (ESC, Rub-Out, etc). Displays all 95 upper- and lower- case characters, rings bell, scrolls on CR, etc. $10.00. Richard Pasco, 235 College Avenue, Mountain View CA 94040.

ANNOUNCING

HOME OWNERSHIP PACKAGE: Keep track of the major costs & benefits of owning a home. $6.95 for program tape and instruction booklet, includes listings. Contact:
Edward Zolandz 7200 Pinemont #906 Houston, TX 77040

BORED WITH SOME OF YOUR PRESENT TAPES? LET'S EXCHANGE!
Contact Scott Parker 5775 Kristin Beaumont, Texas 77706

SPACE INTRUDERS - Action game based on popular arcade (and bar room) game. You must shoot invaders before they reach earth (or shoot you). Machine language, 16K machines only. Send $7.00 to DAVID YOHE, 2910 PHEASANT RUN DRIVE, APT. L, JACKSON, MI 49202.

WANTED: Program to reconcile credit card charge slips with monthly statements and to show balances remaining in credit lines when charge slips and/or payments are posted every few days. Should provide for up tp 10 accounts with as many as 10 transactions each per month. Can assume accounts are current with no finance charges. Send cost and availability to R. E. Bates 9320 Ridgeway Ave Evanston, IL 60203

* * *

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

OCT - NOV, 1981 Vol. II, no. 5

INTERACTION INTO '82

To answer the inquiries from subscribers, yes, INTERACTION continues next year. Again there will be six issues, I'll try to warn you if I take another summer break.

However, printing costs have tripled over the last two years and the postal situation is completely out of hand. Standard third class mail essentially does not exist and bulk rate has proved the only alternative to keep costs down at the additional work of sorting and labeling. Also once I deliver the newsletters to the post office it takes as much as two weeks for the regional post office to sort and distribute them. That means two weeks even for Detroit delivery. Since bulk mail is not forwarded, keep me up to date on address changes even temporary ones, if you expect a newsletter in that time.

The result of the printing and postage inflation is that a subscription will cost $15 for 1982. bulk mail delivery. Any newsletters lost in mailing will be replaced if you notify me. Currently the post office loses 2-5 issues per mailing out of 400+ newsletters. First Class delivery (Canada and Mexico) will be $20.

PROGRAM FILE MANAGER

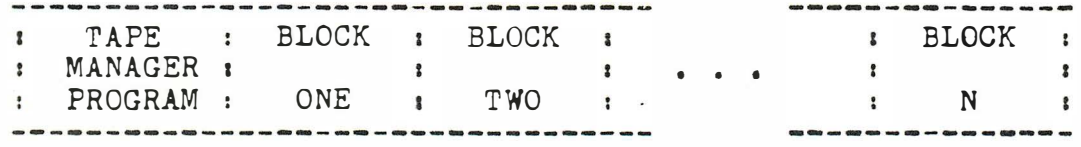
by Edward Zolandz 7200 Pinemont no. 906 Houston, TX 77040

Like many other users of the Interact computer, I have been acquiring a library of programs in BASIC for use on it. As this library has grown I have found it hard to keep track of exactly what is on a given tape and to load a program that is say the tenth one on the tape. Because of this I decided to write a tape file manager so that I could determine what was on a tape as well as speeding up the process of loading programs that are on the tape.

With this in mind I designed a tape manager that would keep an index of all of the programs that are loaded on a given tape and provide a means to locate any program quickly using the fast forward on the tape drive.

Program File Manager, cont.

The program I wrote works by assuming that each program occupies a fixed size block on the tape . Since all of these blocks are the same length any block can be located by rewinding the tape and then fast forwarding for a specific number of seconds. The diagram below shows how each tape is formatted to allow this program to work.



By assigning each block as five seconds worth of tape at the fast forward speed, and allowing one extra second of tape for the tape leader, I can locate any program by knowing which block it is in and then fast forwarding the required number of seconds. In addition by using data statements in the tape manager program I keep a list of what programs are on that tape.

The program that I developed (see listing) uses the real time clock in BASIC to time how long to do the fast foreard. After this it will prompt to have the tape drive put in the read mode and then will load the program into memory.

When I want to add a new program to a tape I first load the tape manager program and add the needed data statement for the new program name. I then save this revised program and execute it to locate the block where the new program is to go. After this I remove the tape from the drive, load the new program into memory from my development tape, replace the original tape into the drive and save the program on the tape.

```
LIST

1 REM PROG FILE SYS
2 REM BY: ISC
3 REM V1.0 6-3-81
4 REM COPYRIGHT 1981
5 REM
10 COLOR4,1,2,7
20 CLS
30 WINDOW11
40 OUTPUT"MENU",43,76,3
50 OUTPUT"1: LIST PROGRAMS",7,69,3
60 OUTPUT"2: SELECT PROGRAM",4,62,3
70 INPUT"OPTION";O
80 IFO<1ORO>2GOTO70
90 ONOGOTO100,240
100 CLS
110 WINDOW77
120 RESTORE
130 N=1
150 READN$
160 IFN$="END"GOTO220
180 PRINTN;N$
190 GOSUB900
```

Program File Manager, cont.

```
200 N=N+1
210 GOTO150
220 GOSUB900
230 GOTO20
240 CLS
250 OUTPUT"PROGRAM",34,48,3
260 OUTPUT"NUMBER",37,41,3
270 INPUTO
280 CLS
290 OUTPUT"DEPRESS REWIND",13,62,3
300 OUTPUT"ANY KEY TO",25,48,3
310 OUTPUT"START",40,40,3
320 N$=INSTR$(1)
330 REWIND
340 CLS
350 OUTPUT"DEPRESS",34,69,3
360 OUTPUT"FAST FORWARD",19,62,3
370 OUTPUT"ANY KEY TO",25,48,3
380 OUTPUT"CONTINUE",31,41,3
390 N$=INSTR$(1)
400 CLS
410 OUTPUT"SEARCHING",28,48,3
420 O=O*5+1
430 CK=24559
440 T=O
450 C=O
460 POKE19215,25
470 POKECK,O
480 POKE4096,84
490 IF(PEEK(CK)-T AND255)<60GOTO490
500 T=T+60AND255
510 C=C+1
520 IFC>=OGOTO540
530 GOTO490
540 POKE4096,20
550 CLS
560 OUTPUT"DEPRESS READ",19,62,3
570 OUTPUT"ANY KEY TO",25,48,3
580 OUTPUT"LOAD",43,41,3
590 N$=INSTR$(1)
600 CLOAD
700 REM PROGRAM NAMES
710 DATA PROGRAM 1
720 DATA PROGRAM 2
730 DATA PROGRAM 3
740 DATA PROGRAM X
880 REM END OF NAMES
890 DATA END
900 FORC=1TO1000
910 NEXTC
920 RETURN
Ok
```

1980 INTERACTION ERROR UNCOVERED

Jerry Lerner 520 Illinois St. Park Forest, IL 60466

There are two "Typo Bugs" in INTERACTION Vol. I, No. 6, page 4 in the program to list BASIC variables: Line 1200 should be: "FOR Z4 = Ø ...", instead of "FOR ZØ = Ø ..." and in Line 1330 the divisor of the last term should be " 2/(24-Z2) " instead of just " (24-Z2) ".

Some further comments: In the example in the text of X1 = 44, the values actually located in the various bytes are as follows:

Byte	Returned by PEEK	Listed in INTERACTION
1	49	31 (Hex = 49_{10})
2	88	58 (Hex = 88_{10})
3	0	0
4	0	0
5	48	176(48 + 128)
6	134 = 128 +(8-2)	130

Apparently BASIC resets the high-order bit of byte 5 to "Ø" for positive rather than negative numbers, which contradicts the text but is the way your variable-listing program is actually written.

It is true that at least one variable must precede Line 1000, but it is also important that no variable follow Line 1400, else some of the routine's variables will be mangled; i.e., the routine must have the highest line numbers.

A deduction from the program (not mentioned in the text) is that the address listed in 19713, 19714 is the termination of the BASIC variables tables.

A SPACE SAVER

Vincent Chobot 2716 S. Culyer Ave. Berwyn, IL 60402

A hint some of your readers might use that saves memory and typing, in Level II BASIC when using PRINT the quotation marks at the end of the line aren't needed; also internal semi-colons aren't needed , only at the end of the line.

For example, in the Basic Monitor program in the last issue (Incidently those POKES in line 100 sure opened a lot of memory to PEEK)

```
   20  PRINT "BASIC MONITOR                                  Saves:
                                                               1 Byte
  400  PRINT MID$(A$,H2+1,1)MID$(A$,H1+1)" ";                  2 Bytes
  500  PRINT TAB(6)CHR$(D)" ";                                 2 Bytes
```

This is particularly helpful in long programs with a lot of PRINT statements. It works on my machine without any SN error codes.

A WORD OF ADVICE

Ruth Ann Halpern 1 Tamarack Rd · Natick, MA 01760

If anyone is planning to buy a new TV to be used with the Interact or another computer do NOT buy a Panasonic or Quasar. They are designed so that they don't work with the computer well. I was able to find a TV technician who modified the set at at a reasonable cost and didn't void my warranty, but it has been more annoyance than its worth.

CAMEL

Enjoy an exciting trek across the great arid desert. The original listing in More Computer Basic Games was so full of bugs that I should have discarded the program when I finished typing it in. However, I didn't and I corrected and modified CAMEL to make it a playable and winnable game.

It now has two skill levels and if you forget the commands hitting any other key will review them for you. The program is probably too kind and should make you die of thirst and get caught by nomads more often. Losing constantly is much too frustrating though. Playing instructions are in the program.

```
LIST

1 REM  *****   CAMEL  *****
2 REM  * INTERACT VERSION *
3 REM  *  COPYRIGHT 1981  *
4 REM  *  BY STEPHEN COOK *
5 REM  **********************
10 CLS
20 PRINTTAB(6);"CAMEL"
35 PRINT:PRINT:PRINT
100 PRINT"  INSTRUCTIONS        (Y - N)?";
110 GOSUB2500:PRINT
120 IFY=30THENPRINT:GOTO270
130 CLS:PRINT"WELCOME TO CAMEL.THE OBJECT IS TO TRAVEL 200 MILES"
140 PRINT"ACROSS THE GREAT ARID DESERT. A"
150 PRINT"TRIBE OF SAVAGE  NOMADS WILL BE"
160 PRINT"CHASING YOU. YOU WILL BE ASKED    COMMANDS EVERY SOOFTEN"
170 GOSUB2000:PRINT
180 GOSUB3000
190 GOSUB2000:PRINT:PRINT
200 PRINT"YOU HAVE 1 QUART OF WATER WHICH   WILL LAST YOU FOR"
210 PRINT"6 DRINKS. YOU MAYRENEW YOUR WATER SUPPLY COMPLETELY"
220 PRINT"AT AN OASIS. YOU GET HALF A QUART IF FOUND BY HELP."
230 GOSUB2000:PRINT:PRINT
240 PRINT"IF HELP DOES NOT FIND YOU AFTER   USING COMMAND 6"
250 PRINT"YOU LOSE AND DIE!"
260 GOSUB2000:PRINT:CLS
270 D=2:PRINT"EASY OR DIFFICULTGAME? (E - D)";
```

Camel, cont.

```
280 GOSUB2500:PRINT
290 IFY=20THEND=0
300 Z=4:S=6:C=0:C1=0:F=0:P=0:Q=0
310 PRINT"  GOOD LUCK AND    GOOD CAMELING!!"
320 GOSUB2000:PRINT:CLS
330 PRINT"YOU ARE IN THE   MIDDLE OF THE    DESERT AT AN     OASIS."
340 IFC>199THEN1210
350 Z=Z-1
355 IFZ=1THENPRINT"-----WARNING-----GET A DRINK!"
360 IFZ<0THEN1630
370 P=P+1
380 X2=INT(12*RND(1)-D)
390 IFQ>0THEN930
400 IFP<4THEN470
410 C1=C1+X2
420 IFC1<CTHEN460
430 PRINT"THE NOMADS HAVE  CAPTURED YOU!    THEY LOVE CAMEL"
440 PRINT"AND PEOPLE STEW!!"
450 GOTO1560
460 PRINT"THE NOMADS ARE"C-C1;"MILES BEHIND YOU."
470 PRINT"YOU HAVE GONE"C:PRINT"MILES TOTAL."
480 PRINT"YOUR COMMAND?";
490 GOSUB2500
495 IFY<1ORY>6THENGOSUB3000:GOTO480
500 ONYGOTO830,610,680,760,790
550 GOSUB2700
560 IFA<10-3*DTHEN1200
570 PRINT"HELP HAS FOUND YOU IN A STATE OF UNCONSCIOUSNESS."
580 S=3
590 Z=4
600 GOTO340
610 F=F+1
620 IFF=8THEN1190
630 X1=INT(10*RND(1)+D)
640 PRINT"YOUR CAMEL LIKES THIS PACE."
650 C=C+X1
660 GOSUB880
670 GOTO340
680 F=F+3
690 IFF>7THEN1190
700 X1=INT(20*RND(1)+D)
710 PRINT"YOUR CAMEL IS    BURNING ACROSS   THE DESERT SANDS."
720 C=C+X1
730 GOSUB880
740 PRINT
750 GOTO340
760 PRINT"YOUR CAMEL THANKS YOU!"
770 F=0
780 GOTO350
790 PRINT"YOUR CAMEL HAS ";7-F;" GOOD DAYS LEFT."
800 PRINT"YOU HAVE ";S;" DRINKS LEFT IN YOUR CANTEEN."
810 PRINT"YOU CAN GO ";Z;" COMMANDS WITHOUT DRINKING."
820 GOTO480
830 S=S-1
840 IFS<0THEN1630
850 PRINT"BETTER WATCH FOR AN OASIS !"
```

Camel, cont.

```
860 Z=4
870 GOTO480
880 GOSUB2700
890 IFA>3-DTHEN1120
900 PRINT"WILD BERBERS HID-DEN IN THE SAND  HAVE CAPTURED YOU"
910 PRINT"LUCKILY THE LOCALSHEIK HAS AGREED TO THEIR RANSOM"
920 PRINT"DEMANDS BUT .... WATCH OUT FOR THENOMADS!"
925 GOSUB2000:PRINT:PRINT
930 PRINT"NOW YOU HAVE 2   NEW COMMANDS:"
940 PRINT"7>ATTEMPT AN":PRINT"  ESCAPE"
950 PRINT"8>WAIT FOR":PRINT"  PAYMENT"
960 PRINT"YOUR SUBCOMMAND?";
970 GOSUB2500
975 Q=1:IFY<7ANDD=0GOTO380
980 IFY=8THEN1060
990 GOSUB2700
1000 IFA<40-10*DTHEN1040
1010 IFA>70+10*DTHENPRINT"YOUR ESCAPE WAS  FOILED! TRY AGAIN":PRINT:GOTO93
1020 PRINT"CONGRADULATIONS, YOU SUCCESSFULLY ESCAPED!!!!"
1030 Q=0:GOTO340
1040 PRINT"YOU WERE MORTALLYWOUNDED WHILE    TRYING TO ESCAPE!"
1050 Q=0:GOTO1410
1060 GOSUB2700
1080 IFA>24THEN1100
1090 PRINT"YOUR RANSOM HAS  BEEN PAID AND YOUARE FREE TO GO."
1095 Q=0:GOTO340
1100 PRINT"THE LOCAL SHEIK  IS COLLECTING....JUST WAIT...."
1110 FORL=1TO2000:NEXTL:Q=0:GOTO340
1120 GOSUB2700
1130 IFA>15+3*DTHEN1240
1140 PRINT"YOU HAVE ARRIVED AT AN OASIS ____ YOUR CAMEL IS"
1150 PRINT"FILLING YOUR CAN-TEEN AND EATING  FIGS."
1160 Z=4:S=6
1180 RETURN
1190 PRINT"YOU DIRTY RAT!!  YOU RAN YOUR POORCAMEL TO DEATH!"
1200 GOTO1410
1210 PRINT"YOU WIN, A PARTY IS BEING GIVEN INYOUR HONOR!!!!"
1220 PRINT"THE NOMADS ARE   PLANNING TO COME TOO!!"
1230 GOTO1560
1240 GOSUB2700
1250 IFA>7-DTHEN1350
1260 PRINT"YOU HAVE BEEN    CAUGHT IN A SAND-STORM. GOOD LUCK!"
1270 GOSUB2700:X5=INT(A/10)
1280 GOSUB2700
1290 IFA<50THEN1320
1300 C=C+X5
1310 GOTO1330
1320 C=C-X5
1330 PRINT"YOUR NEW POSITIONIS";C;"MILES":PRINT"SO FAR!"
1340 RETURN
1350 GOSUB2700
1360 IFA>7-DTHENRETURN
1370 C=C-X1:C1=C1+1
1380 PRINT"YOUR CAMEL HURT  HIS HUMP."
1390 PRINT"LUCKILY FOR YOU  THE NOMADS WERE  TIRED."
1400 RETURN
```

Camel, cont.

```
1410 GOSUB2700
1415 PRINT:FORL=1TO500:NEXTL
1420 PRINT"YOU DIED IN THE  DESERT."
1430 IFA>10THEN1460
1440 PRINT"THE INTERNATIONALCAMEL UNION IS   NOT ATTENDING    YOUR FUNERAL!
"
1450 GOTO1560
1460 IFA>30THEN1490
1470 PRINT"YOUR BODY WAS    EATEN BY VULTURESAND JACKELS!"
1480 GOTO1560
1490 IFA>50THEN1520
1500 PRINT"THE NOMADS NOW   USE YOUR SKULL   FOR AN ASHTRAY!"
1510 GOTO1560
1520 IFA>70THEN1550
1530 PRINT"TOTAL IDIOTS     SHOULD STAY OUT  OF THE DESERT!"
1540 GOTO1560
1550 PRINT"TURKEYS SHOULD   FLY, NOT RIDE    CAMELS!!!!!"
1560 PRINT
1570 PRINT
1580 PRINT"WANT A NEW CAMEL AND A NEW GAME?"
1585 PRINT" ( Y - N )";
1590 GOSUB2500:PRINT
1600 IFY=41THENCLS:GOTO270
1620 GOTO1650
1630 PRINT"YOU RAN OUT OF   WATER!":PRINT"SORRY CHUM...."
1640 GOTO1410
1650 CLS:PRINT"******************"
1660 PRINT:PRINT" NEXT TIME TAKE      THE BUS!":PRINT
1670 PRINT"******************"
1690 END
2000 PRINT" < HIT ANY KEY >";
2020 D$=INSTR$(1)
2050 RETURN
2500 Y$=INSTR$(1)
2510 PRINTY$
2520 Y=ASC(Y$)-48
2550 RETURN
2700 A=INT(100*RND(1))
2750 RETURN
3000 REM COMMAND LIST
3010 PRINT:PRINT"COMMANDS:"
3020 PRINT"1>DRINK FROM YOUR  CANTEEN"
3030 PRINT"2>AHEAD MODERATE   SPEED"
3040 PRINT"3>FULL SPEED"
3050 PRINT"4>STOP OVERNIGHT"
3060 PRINT"5>STATUS CHECK"
3070 PRINT"6>HOPE FOR HELP"
3100 RETURN
Ok
```

WANTED for radio amateur club - Used INTERACT computers - 8K or 16K - working or not - Send description and asking price, (incl. phone number and time we can contact you) to:

Glenn Manthey W60 N661 Jefferson Ave. Cedarburg, WI 53012

AVERAGE DAILY TEMPERATURES

by George Newcomer 413 Pearl Charlotte, MI 48813

This program was developed so that I could compare the high and low average daily temperatures of different cities. I plan to retire in about five years and am thinking of a warmer climate. I used the 8K Fast graphics BASIC, but it will run alright if lines 50, 60, and 1010 are eliminated and CLS is used in line 50.

I used two tapes in my program, one for lines 10 to 1290 and another for the data. The append feature of the Ezedit program is used to bring the two together. They are appended in this listing.

```
LIST

10 REM AVERAGE DAILY TEMPERATURE'S
20 REM  WRITTEN BY GEORGE NEWCOMER
30 REM  413 PEARL
40 REM  CHARLOTTE, MICHIGAN  48813
50 PLOT1,1,1,112,77
60 PLOT0,0,0,114,77
70 DIMAH(32),AL(32),DH(32),DL(32),LH(32),LL(32)
80 A1=0:A2=0:D1=0:D2=0:L1=0:L2=0
90 T$="TEMPERATURE"
100 OUTPUTT$,56-3*LEN(T$),40,1
110 READMO$
120 OUTPUTMO$,56-3*LEN(MO$),30,1
130 H1=AH(1)
140 H3=DH(1)
150 H5=LH(1)
160 READX
170 FORI=1TOX
180 READ AH(I),AL(I),DH(I),DL(I),LH(I),LL(I)
190 IFAH(I)<-71THEN420
200 IFH1>AH(I)THEN220
210 H1=AH(I)
220 IFH3>DH(I)THEN240
230 H3=DH(I)
240 IFH5>LH(I)THEN260
250 H5=LH(I)
260 A1=A1+AH(I)
270 A3=A1/X
280 A2=A2+AL(I)
290 A4=A2/X
300 D1=D1+DH(I)
310 D3=D1/X
320 D2=D2+DL(I)
330 D4=D2/X
340 L1=L1+LH(I)
350 L3=L1/X
360 L2=L2+LL(I)
370 L4=L2/X
380 A5=A3+A4
390 D5=D3+D4
400 L5=L3+L4
410 NEXT I
```

Average Daily Temperatures, cont.

```
420 CLS:PRINT"1-ASHEVILLE"
430 PRINT"2-DENVER"
440 PRINT"3-LANSING"
450 PRINT"WHAT CITY DO YOU":PRINT"WANT?"
460 I$=INSTR$(1)
470 IFI$="1"THEN500
480 IFI$="2"THEN650
490 IFI$="3"THEN810
500 CLS:PRINTTAB(4)"ASHEVILLE"
510 PRINT:PRINT"HIGH="INT(A3+.5)
520 PRINT"LOW="INT(A4+.5)
530 IFA5>L5THENPRINT"IS";INT((A5-L5)+.5);"WARMER":PRINT"THAN LANSING"
540 IFA5>D5THENPRINT"IS";INT((A5-D5)+.5);"WARMER":PRINT"THAN DENVER"
545 IFA5<D5THENPRINT"IS";INT((D5-A5)+.5);"COOLER":PRINT"THAN DENVER"
550 IFA5<L5THENPRINT"IS";INT((L5-A5)+.5);"COOLER":PRINT"THAN LANSING"
560 PRINT"<HIT ANY KEY>"
570 I$=INSTR$(1)
580 PRINT"HIGH TEMPERATURE"
590 PRINT"FOR MONTH IS"
600 PRINT;H1
610 GOSUB1030
620 PRINT"LOW TEMPERATURE"
630 PRINTTA
640 GOTO960
650 CLS:PRINTTAB(5);"DENVER"
660 PRINT:PRINT"HIGH="INT(D3+.5)
670 PRINT"LOW="INT(D4+.5)
680 IFD5>A5THENPRINT"IS";INT((D5-A5)+.5);"WARMER":PRINT"THAN ASHEVILLE"
690 IFD5>L5THENPRINT"IS";INT((D5-L5)+.5);"WARMER":PRINT"THAN LANSING"
700 IFD5<A5THENPRINT"IS";INT((A5-D5)+.5);"COOLER":PRINT"THAN ASHEVILLE"
710 IFD5<L5THENPRINT"IS";INT((L5-D5)+.5);"COOLER":PRINT"THAN LANSING"
720 PRINT"<HIT ANY KEY>"
730 I$=INSTR$(1)
740 PRINT"HIGH TEMPERATURE"
750 PRINT"FOR MONTH IS"
760 PRINTH3
770 GOSUB1120
780 PRINT"LOW TEMPERAURE"
790 PRINTTD
800 GOTO960
810 CLS:PRINTTAB(5);"LANSING"
820 PRINT:PRINT"HIGH="INT(L3+.5)
830 PRINT"LOW="INT(L4+.5)
840 IFL5>A5THENPRINT"IS";INT((L5-A5)+.5);"WARMER":PRINT"THAN ASHEVILLE"
850 IFL5>D5THENPRINT"IS";INT((L5-D5)+.5);"WARMER":PRINT"THAN DENVER"
860 IFL5<A5THENPRINT"IS";INT((A5-L5)+.5);"COOLER":PRINT"THAN ASHEVILLE"
870 IFL5<D5THENPRINT"IS";INT((D5-L5)+.5);"COOLER":PRINT"THAN DENVER"
880 PRINT"<HIT ANY KEY>"
890 I$=INSTR$(1)
900 PRINT"HIGH TEMPERATURE"
910 PRINT"FOR MONTH IS"
920 PRINTH5
930 GOSUB1210
940 PRINT"LOW TEMPERATURE"
950 PRINTTL
960 PRINT"DO YOU WISH"
```

Average Daily Temperatures, cont.

```
970 PRINT"ANOTHER CITY?"
980 C$=INSTR$(1)
990 IFC$="Y"THEN420
1000 PLOT1,1,1,112,77
1010 PLOT0,0,0,114,77
1020 END
1030 RESTORE
1040 TA=AL(1)
1050 READMO$,X
1060 FORI=1TOX
1070 READAH(I),AL(I),DH(I),DL(I),LH(I),LL(I)
1080 IFTA<=AL(I)THEN1100
1090 TA=AL(I)
1100 NEXTI
1110 RETURN
1120 RESTORE
1130 TD=DL(1)
1140 READMO$,X
1150 FORI=1TOX
1160 READAH(I),AL(I),DH(I),DL(I),LH(I),LL(I)
1170 IFTD<=DL(I)THEN1190
1180 TD=DL(I)
1190 NEXTI
1200 RETURN
1210 RESTORE
1220 TL=LL(1)
1230 READMO$,X
1240 FORI=1TOX
1250 READAH(I),AL(I),DH(I),DL(I),LH(I),LL(I)
1260 IFTL<=LL(I)THEN1280
1270 TL=LL(I)
1280 NEXTI
1290 RETURN
1300 DATA FEBRUARY 1981,28
1310 DATA 40,20,28,7,38,25
1320 DATA 36,24,49,-3,27,7
1330 DATA 29,11,40,11,10,1
1340 DATA 39,8,34,8,10,0
1350 DATA 40,13,40,12,19,3
1360 DATA 40,25,40,14,27,19
1370 DATA 50,18,50,21,35,22
1380 DATA 51,32,51,14,27,15
1390 DATA 49,17,31,22,24,13
1400 DATA 42,33,1,-5,39,13
1410 DATA 53,18,36,-12,37,-2
1420 DATA 38,6,53,24,13,-4
1430 DATA 45,11,62,25,29,6
1440 DATA 45,26,62,25,38,20
1450 DATA 60,23,61,35,44,28
1460 DATA 58,35,69,34,44,38
1470 DATA 53,45,67,37,44,36
1480 DATA 54,52,67,37,53,42
1490 DATA 54,50,71,39,57,40,
1500 DATA 57,40,68,31,40,35
1510 DATA 57,44,68,29,44,29
1520 DATA 59,30,56,24,47,33
```

Average Daily Temperatures, cont.

```
1530 DATA 53,39,61,32,47,34
1540 DATA 53,31,61,32,42,27
1550 DATA 67,25,68,33,36,30
1560 DATA 69,30,60,31,32,26
1570 DATA 73,26,58,35,32,21
1580 DATA 73,33,58,30,48,36
OK
```

PRODUCT REVIEWS

Review - SPACE INTRUDER

The alien invaders don't have the same cute appearance and waving arms of the arcade version, but David Yohe's creatures are just as fast moving. They also viciously shoot back as good as the coin version and only the three barricades save you from almost instant destruction. The action in this machine language game (Remember, that means no need to load BASIC, only the program tape) is very smooth. Even the only noticeable slowdown, when the mother ship appears across the top, is not enough to make your shooting easy. In addition to your gun, the three fortresses, the 55 aliens and the occasional mother ship, the screen display also shows your score, the bonus points if you hit the ship and the highest score for the previous games. At $7 this is an exceptional value for a game and would still be worth buying for twice the price.

Review - GOLD MINER

Having played a version of this game on a PET computer, I knew what to expect when I received GOLD MINER. What was unexpected was that Bill Deacon had dumped his original Basic Version and had written a machine code version of the program. Being unable to write machine code myself, I find any machine language program fascinating and this game is interesting in both form and content.

In GOLD MINER you graphicly descend a mine shaft and dig tunnels at various levels in search of gold. However, there are dangers and difficulties. Moving to some points may cause a cave-in and possible loss of the gold you've accumulated. Making a trip up the shaft to the bank saves your gold from loss. When tunneling you may hit granite which needs to be blasted, costing you extra money(any digging costs money and you can spend more than you make). If you tap an underground spring you flood the tunnel you're in and all the tunnels on levels below you. To use these tunnels again you must first pump out the water, another added expense. The game has no win/lose situation. The game ends when all the gold is recovered or you spend all your funds, you're initially grubstaked with $100.

The playability of this program and its good use of color graphics makes it another program worth more than it cost. See Bill's ad elsewhere in this issue to order.

Product Reviews, cont.

Review Interact Chat

Lower case on the TV screen from an Interact! Yes, by not using the ROM based character set, Rich Pasco's new Slagh port based Terminal Program displays 8 lines of 17 upper and lower case characters. The program has its own 5 X 7 character set and enables the user to more readily use many time sharing systems where lower case entries are needed. What is even more useful to me is INTERACT CHAT's ability to transmit BREAK, a control often used in mainframe systems such as the Wayne County Intermediate School District's HP 3000 which I often access. BREAK is the only way you can stop or leave a program on this HP 3000 as it does not recognize a CONTROL-C or ESCAPE. Numerous times I was forced to hang up the phone in order to leave a program, not a recommended proceedure. Rich also seems to have used a slower or better keyboard input routine. When I logged on MicroNET previously I usually had to type in my password several times to get it right. With CHAT I rarely get lost or extra characters.

With INTERACT CHAT, to enter control characters you type CONTROL-A, the screen then reverses - it becomes white letters on a black field - and output continues until you enter the two digits of the hex value needed. It is then transmitted and the screen changes to the normal black on white. While this means three inputs instead of the usual one to enter ASCII values, they are used only occasionally anyway and thus you are no longer limited as to what values you could send by what is available on the keyboard. Any value between ØØ and 7F hex can be sent. 8Ø to FF will send the long space for BREAK.

In conclusion while I still can not send or receive files and programs with my Interact, CHAT seems to be the best Interact terminal program I've used so far.

TYPOGRAPHICAL ERROR

In the last issue of INTERACTION there were two typographical errors in Harry Holloway's articles. My error, not his. One on page 2, second paragraph, it should be $4D21_h$ - $5FBF_h$. The second on page 4, second paragraph, should be

ØØ $data_1$ $data_2$ ... $data_{1ØØ}$
ØØ $data_1$ $data_2$ ... $data_{1ØØ}$
1F $data_1$ $data_2$ ... $data_{1F}$

L GAME

Bob Draganski 14301 Harrison Livonia, MI 48154

This program is written in Level II BASIC and the board display could be speeded up if it was converted to Fastline or Graphics Basic. However the game is given in Level II for the benefit of those who do not have those options.

L GAME is played on a 4 X 4 grid with each player having a 3 X 2 L and 2 neutral square boxes to move. The object is to place your L and the boxes so that the computer cannot move its L. To place your L type in the squares location numbers when prompted. The same with the boxes. None of the pieces can overlap. To leave a box in place, type in its current position. Remember the computer is trying to trap your L. If you cannot move your L on your turn, you lose, and of course if you pin the computer's L you win.

```
LIST

1 REM L GAME
2 REM INTERACT VERSION BY BOB DRAGANSKI
3 REM COPYRIGHT 1981
5 G=RND(-PEEK(24599))
10 CLEAR:CLS:WINDOW18:OUTPUT"THE L GAME",18,50,2
20 GOSUB4000:CLS
40 DIMO(4),C(4),D(4),T(4),B(16),M(100),N(16)
120 DATA1,16,2,7,6,6,11,7,10,15,10,3,14,11,3,2,2,0,0,2,1,0,0,2,1,0,0,1,1
130 DATA3:READB1,B2:FORX=1TO4:READC(X),O(X),T(X):NEXT:FORX=1TO16:READB(X)
140 NEXT
210 OUTPUT"DO YOU WISH TO   START(Y/N)?",6,50,2:A$=INSTR$(1)
215 CLS:FORX=29TO74STEP11:FORY=21TO65:PLOTX,Y,2:NEXT:NEXT
216 FORY=21TO65STEP11:FORX=29TO73:PLOTX,Y,2:NEXT:NEXT
220 GOSUB2270:IFA$="N"THEN500
240 PRINTCHR$(7):PRINT"YOUR MOVE FOR L":INPUTD(1),D(2),D(3),D(4)
245 PRINTD(1);D(2);D(3);D(4):PRINT"O.K.FOR L (Y/N)?";:A$=INSTR$(1):PRINT:PR
INT
246 G=0:IFA$="N"THEN240
265 FORX=1TO4:FORY=2TO4:IFD(Y)>D(Y-1)THEN295
270 Z=D(Y):D(Y)=D(Y-1):D(Y-1)=Z
295 NEXT:NEXT
310 FORX=1TO4:N(X)=D(X):NEXT:N1=4:F1=0:GOSUB1800:IFM1<>4THEN1220
345 FORX=1TO4:IFB(M(X))>1THEN1220
355 NEXT:FORX=1TO4:B(O(X))=0:NEXT:FORX=1TO4:B(M(X))=1:O(X)=M(X):NEXT
400 GOSUB2270:PRINTCHR$(7):B(B1)=0:B(B2)=0
405 PRINT"MOVE FOR BOXES?":INPUTX,Y:PRINTX;Y;"O.K.(Y/N)?";:A$=INSTR$(1):PRI
NT:PRINT
410 IFA$="N"THEN405
415 G=1:IFABS(X)+ABS(Y)<>X+YORX=YORB(X)+B(Y)>0THEN1220
430 B1=X:B2=Y:B(B1)=3:B(B2)=3:GOSUB2270
500 FORX=1TO4:B(C(X))=0:NEXT
502 PRINTCHR$(7)
505 PRINT"COMPUTER THINKINGABOUT THE L";
530 GOSUB1420:F1=1:N1=Z:GOSUB1800:IFM1=0THEN1300
585 GOSUB1370:FORE=0TOM1-4STEP4:FORF=1TO4:FORG=1TO4
620 IFM(E+F)<>T(G)THEN640
```

L Game, cont.

```
630 N(E/4+1)=N(E/4+1)+1
640 NEXT:NEXT:NEXT:GOSUB1500:Y=(Z-1)*4:FORX=1TO4:C(X)=M(X+Y):B(C(X))=2
645 NEXT:GOSUB2270:PRINT:PRINT:PRINT"COMPUTER THINKINGABOUT THE BOXES";
646 B(B1)=0:B(B2)=0:FORI=1TO4:FORJ=1TO4:IFO(I)=T(J)THEN870
830 NEXT:NEXT:GOTO1020
870 FORX=1TO4:IFB(T(X))>0THEN920
890 B1=T(X):B(B1)=3:GOTO950
920 NEXT:GOTO1020
950 FORX=1TO4:IFB(T(X))>0THEN1000
970 B2=T(X):B(B2)=3:GOTO1080
1000 NEXT:GOTO1050
1020 GOSUB1610:B1=B3:B(B1)=3
1050 GOSUB1610:B2=B3:B(B2)=3
1080 FORX=1TO4:B(O(X))=1:NEXT:GOSUB2270:FORX=1TO4:B(O(X))=0:NEXT
1100 PRINT:PRINT:PRINT
1130 GOSUB1420:F1=2:N1=Z:GOSUB1800:IFM1=0THEN1280
1180 FORX=1TO4:B(O(X))=1:NEXT:GOTO240
1220 TONE90,100:PRINT"ILLEGAL MOVE":FORI=1TO1000:NEXT:IFG=1THEN405
1222 GOTO240
1280 OUTPUT"COMPUTER WINS",18,72,2:TONE190,200:GOTO1320
1300 PRINT:PRINT:OUTPUT"YOU WIN",30,72,2:SOUND3,2604:GOSUB4000:SOUND7,4096
1320 OUTPUT"PLAY AGAIN(Y/N)?",6,18,2:A$=INSTR$(1):IFA$="Y"THEN10
1330 END
1370 FORX=1TO16:N(X)=0:NEXT:RETURN
1420 Z=0:FORX=1TO16:IFB(X)>0THEN1470
1450 Z=Z+1:N(Z)=X
1470 NEXT:RETURN
1500 Y=0:Z=1:FORX=1TOM1/4:IFN(X)<YTHEN1580
1510 IFN(X)>YTHEN1560
1520 IFRND(1)>.5THEN1580
1560 Y=N(X):Z=X
1580 NEXT:RETURN
1610 FORX=1TO4:B(O(X))=0:NEXT:GOSUB1420:F2=1:N1=Z:GOSUB1800:GOSUB1370
1690 FORX=1TOM1:N(M(X))=N(M(X))+1:NEXT:FORX=1TO4:N(O(X))=0:NEXT:M1=64
1760 GOSUB1500:B3=Z:RETURN
1800 M1=0:J=4:K=1:GOSUB1880:J=1:K=4:GOSUB1880:RETURN
1880 P=0
1890 P=P+1:A(1)=N(P):X=P
1920 X=X+1:IFX>N1THEN2050
1940 IFN(X)-A(1)<>JTHEN1920
1950 A(2)=N(X)
1960 X=X+1:IFX>N1THEN2050
1980 IFN(X)-A(2)<>JTHEN1960
1990 A(3)=N(X):FORE=1TON1:IFABS(N(E)-A(1))=KTHEN2060
2020 IFABS(N(E)-A(3))=KTHEN2060
2030 NEXTE:GOTO1890
2050 IFP<N1-2THEN1890
2055 RETURN
2060 A(4)=N(E):FORF=1TO4:IFA(F)/4<>INT(A(F)/4)THEN2130
2090 FORG=1TO4:IFA(G)=A(F)+1THEN2030
2120 NEXTG
2130 NEXTF:FORY=1TO4:IFF1=1THEN2190
2140 IFA(Y)<>O(Y)THEN2210
2170 NEXTY:GOTO2030
2190 IFA(Y)<>C(Y)THEN2210
```

L Game, cont.

```
2200 GOTO2170
2210 FORY=1TO4:M(M1+Y)=A(Y):NEXTY:M1=M1+4:GOTO2030
2270 L=1:FORE=1TO13STEP4:FORG=0TO3:IFE=13THENE=16
2275 X=G*11+30:Y=75-INT(SQR(E))*11
2279 FORO=0TO5STEP5
2280 FORI=0TO5STEP5:OUTPUT"O",X+I,Y-O,B(L+G):OUTPUT"*",X+I,Y-O,B(L+G)
2281 NEXTI:NEXTO:IFB(L+G)=0ANDL+G<10THENOUTPUTL+G,X-4,Y-2,2
2282 IFB(L+G)=0ANDL+G>9THENOUTPUTL+G,X-7,Y-2,2
2285 IFB(L+G)=3THENOUTPUT"X",X+2,Y-2,2
2290 TONE30,100
2300 NEXTG:L=L+G:NEXTE:RETURN
4000 FORI=1TO1000:NEXTI:RETURN
Ok
```

LETTERS TO INTERACTION

George A. Leggett
52895 Bunker Hill Blvd
New Baltimore, MI 48047
July 15, 1981

In response to the Interact Display Format by Richard Pasco (Interactions Vol. 2 No. 3 May-June. 1981) I wish to make a few corrections and clarifications. On my 13-inch color portable TV I cannot see the coordinate 5,5. However, on my 25-in TV I can see the coordinate 1,1. The point is that each TV is slightly different in it vertical and horizontal settings. 4000 Hex as shown in the Display Format can be seen. Although the highest X coordinate I can see is 112, the screen memory uses up to 127. All of the ROM subroutines use this memory when dealing with the screen. I suppose one could download this onto a printer or other terminal to make full use of the 127 X coordinates. I have been very successful at dumping this into other memory.

It was also stated that Hex addresses 4980 to 49FF do not appear on the screen. Even though some sets may not display it the ROM subroutines still use addresses 4980 to 499F as screen memory. For instance the Clear Screen Routine in the ROM will destroy any information between 4000 and 499F. The memory from 49A0 on is useable memory except for those who have a ROM from W. Hendrickson which uses the 4A00 block.

Letters, cont.

Richard C. Pasco
235 College Avenue
Mountain View CA 94040

This is in reply to George A. Leggett's response to my article "Interact Display Format."

Mr. Leggett's comments are mostly correct and well taken, but they indicate a misunderstanding of the purpose of the table. It was not my intent to describe the interaction between the Interact video signal and any "typical" television (there is too much variation between sets for that); rather, my table presents an explicit and concise mapping among screen coordinates, memory addresses and the video signal.

I would be very much surprised if Mr. Leggett can really see coordinate 1,1 on his 25-inch color TV, because only the 76 rows for Y=2 through Y=77 appear between the vertical blanking intervals in the video signal, and no TV can display a signal that isn't there. Conversely, the row Y=2 does appear in the video signal, but whether one can see it of course depends on the amount of overscan in the TV. [*Overscan*, by definition, is the amount by which the unblanked portion of the video raster extends beyond the boundaries of the visible screen.] I agree that each TV is different in the amount of overscan, both horizontal and vertical.

Regarding addresses 4980 to 49FF, Mr. Leggett says, "Even though *some* sets may not display it..." In fact, *no* set will display the content of this area, as it does not appear in the video. I agree that it would be unwise, however, to attempt to use these addresses for valuable data, because as Mr. Leggett correctly observed, the screen-manipulation firmware does alter their contents.

One might hypothesize that Interact Electronics originally had planned for there to be two more rows of pixels (Y=0 and Y=1, addresses 4980 to 499F), but after the firmware was written, the video circuit was changed so these rows were blanked. And it is remotely possible that Mr. Leggett's Interact differs from mine by displaying these extra rows.

Regarding columns X=112 through X=127, we agree that they are not displayed on most TV's even though they are accessed by the ROM screen-manipulation routines. Timing analysis indicates that it is impossible for all 128 columns to fit between horizontal blanking intervals (given the time taken by the first 112). But I did not measure where the exact cutoff was, as I had done for the vertical sweep. Hence my only (admittedly misleading) allusion to "most TVs."

ANNOUNCEMENTS

Due to current unavailability of some of their program offerings, Creative Mind Workshop does not have a catalog to offer at this time. Please do not request it as per their ad in the last issue.

More information and an updated ad should be forthcoming in future INTERACTIONS.

Due to a change in employment, Mark Slagh can no longer offer his Call-In Night. Slagh System Services will continue to sell and support their products through their address of P.O. Box 438 Ypsilanti, MI 48197

If you liked FASTLINE you'll love
THE LEONARDO GRAPHICS SYSTEM

This program extends the concept of an overlay to creation of a special-purpose graphics BASIC. Starting with Level II BASIC (or MV's newer 8K BASIC) a set of new graphics commands is inserted in place of string functions that are rarely used in a graphics context. (INSTR$ and CHR$, which are useful here have been retained.) Also the commands CSAVE* and CLOAD* have been replaced by commands that transfer pictorial information between the screen and tape.

The new commands are:
UDO - sets the coordinates of a user-defined origin.
POS - extended from Level II to also return the coordinates of the UDO.
PEN - sets a color for graphics output.
DOT - sets or clears a flag for dotted line output from LINE, MOVE, and ARC.
PLOT - extended from Level II to accept the three coordinate systems that are described below.
LINE - draws a line between any two points.
MOVE - draws line segments to connect a specified sequence of points.
TRI - draws a filled triangle with a specified set of vertices. May be used in combination to draw any straight-sided figure.
BOX - draws a rectangle.
ARC - draws an arc of a circle (including the circle as a special case).
SECT - draws a sector of a circle (including the filled circle as a special case).
CSAVE↑ - writes a tape from all or part of the screen.
CLOAD↑ - loads a tape made with CSAVE↑ .

All commands that specify points on the screen occur in three forms that accept different coordinate systems, e.g
PLOT X,Y - the usual BASIC coordinates.
PLOT! U,V - Cartesians relative to the UDO.
PLOT% L,A - Polar coordinates relative to the UDO.

Relative coordinates simplify writing subroutines to draw the same figure in different places. The polar routines avoid BASIC's slow floating-point SIN and COS functions. Instead they use LEONARDO's own fixed-point routines, which run many times faster (these may also be called from the user's machine-language subroutines).

The program does not reduce the amount of memory that is available for the user's BASIC program. (For graphics there is effectively an increase, because the programs are much more compact.)

With comprehensive documentation and programming examples, $20 postpaid (MI residents + 80¢ tax) from Harry Holloway, Box 2263, Ann Arbor, MI 48106.

"GOLD MINE" Strike it rich!

Play the colorful game of strategy and chance. Graphics, sounds and a novel game that you won't soon tire of.
Now in machine language - still only $5.95

Contact: William Deacon 12353 Big Lake Rd Davidsburg, MI 48019

SPACE INTRUDERS - Action game based on popular arcade (and bar room) game. You must shoot invaders before they reach earth (or shoot you). Machine language, 16K machines only. Send $7.00 to DAVID YOHE, 2910 PHEASANT RUN DRIVE, APT. L, JACKSON, MI 49202.

INTERACT CHAT

FULL ASCII Character Set in Terminal Emulator Program for Interact with Slagh RS-232 Port. Sends upper- and lower- case using "SHIFT" accepts hex code for others (ESC, Rub-Out, etc). Displays all 95 upper- and lower- case characters, rings bell, scrolls on CR, etc. $10.00. Richard Pasco, 235 College Avenue, Mountain View CA 94040.

The SABRE Port

An RS232 level serial printer port for the Interact. Price includes Level II Basic overlay program, installation instructions, and operating manual. Installation requires small amount of soldering. $24.99 plus $2.00 for shipping and handling from:
SABRE, 1415 Creek Hollow Dr., Seabrook TX. 77586 (713)870-8315

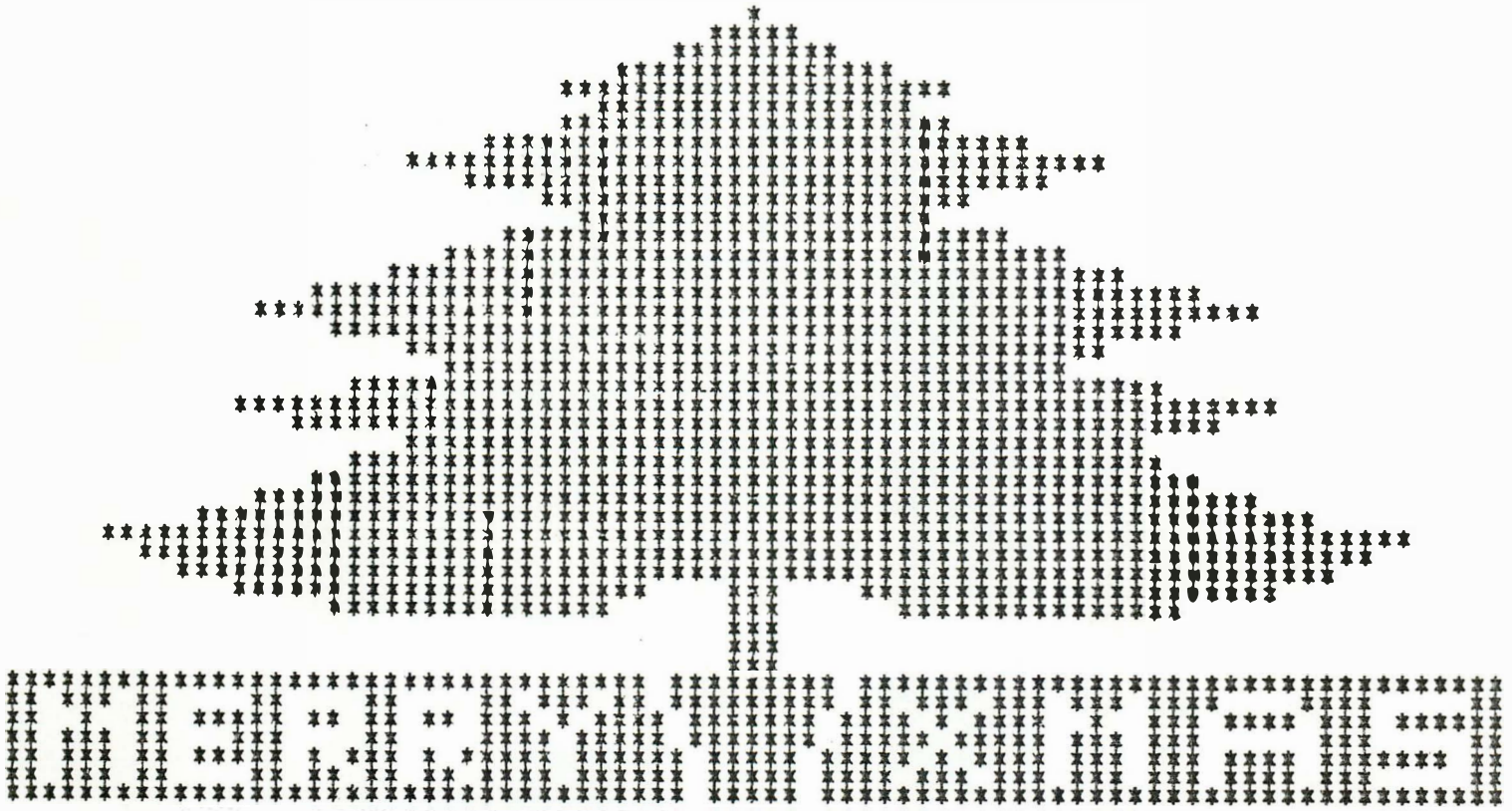
INTERACT EXPANSION - All 8080 signals buffered (except RST) to 40 conductor ribbon cable - quality double sided, plated, solder-masked P.C. board - allows memory expansion to 60K, floppy disc expansion and capability for S-100. For info, send large SASE to: Walter H. Jopke 5016 W. 99th St. Bloomington, Minnesota 55437

BASIC LEVEL II PROGRAMS $7.00 EACH: BASEBALL (1 OR 2 PLAYERS)
MOON LANDER (GRAPHIC DISPLAY - REACT OR CRASH)
8K GRAPHICS PROGRAMS $7.00 EACH: FLASH CARDS (JUMBO MATH FUN)
SLOT MACHINE (LIKE LAS VEGAS WITH GRAPHICS DISPLAY)
ALVY ALBERT - 1704 CADILLAC CIR. S. - MELBOURNE, FL. - 32935

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

DECEMBER, 1981 VOL. II, no. 6



This issue finishes another year of INTERACTION. I hope all of you have enjoyed and benefited from the programs and articles. I take personal pride in trying to provide a quality newsletter for you. I hope I can continue to meet your expectations in 1982.

I would like to thank all my contributors both published and unpublished. I welcome all your contributions and even appreciate your criticism. Please continue. I am already planning and soliciting materials for next year. Harry Holloway has offered to write some articles on the workings of Microsoft BASIC. George Leggett is going to do a machine language question and answer column. Please feel free to send me any article or information that you think would be suitable for publication.

LAST ISSUE FOR 1981

TO SUBSCRIBE FOR 1982 TURN TO PAGE 23

Editorial, cont.

I have a number of programs still to be published and I welcome new contributions. I will send you a cassette if you want to submit a program. This helps to eliminate the typing errors that have happenned in the past. I have avoided publishing improvements of programs already published. I think while these are often very good, I'm sure you would appreciate something new instead of a re-hash of a old program. Let me know if you think differently.

One repeat program next year will be a newly enhanced version of Kevin TenBrook's LUNAR LANDER. His new program is greatly speeded up and offers better control for landing. It also has a fantastic animated lunar module landing for the beginning of the program. Also upcoming next year is a household budget and analyzer program, a connect four program, a graph plotting program by Harry Holloway, some dice games including a BASIC version of Interact Knockdown, a sunrise - sunset time program, a forest firefighting simulation and numerous other useful and entertaining programs.

Please join me and other Interact owners in making INTERACTION a vital part of your Interact software library and documentation.

FIRE 3

Kevin TenBrook 12813 Westpark Houston, TX 77082

This is a naval artillery game for 2 players or 1 player and the computer. Playing is done with the joysticks. The pot controls the angle of elevation and pressing the joystick forward increases the shell velocity while pulling back decreases the velocity. The computer's shots are slightly random giving you a chance to get the range on its ship and sink it before it sinks you. The ships are moving towards each other making range readjustments constantly necessary.

Entering the program is not as easy as playing it. FIRE 3 features a machine language interrupt subroutine as well as a BASIC listing. Joining a machine language routine may be new to many of you so I'll try to help you along.

You can enter the machine code by POKEing from BASIC, or using a BASIC MONITOR such as the one in the May-June, 1981 INTERACTION, or using a machine language monitor such as Micro-Video's or better still the HI-LO MONITOR by Harry Holloway or one of the ROM Monitors.

To POKE from BASIC convert each hex value in the listing following to a decimal value and start POKEing at 22528_{10}. You should end at 22703_{10}. Using the BASIC MONITOR you avoid having to do the hex conversion. With a machine code monitor use a substitute command and type in the values starting at 5800_h as per the left hand column of the listing. The values you enter are the second column. These are given for your reference as are the comments on the right. The tables and clear routine follow in consecutive sequence the end of the interrupt routine and end at $58AF_h$. If your monitor does a CHECKSUM you should have $37C1_h$ for 5800_h to $58AF_h$.

Fire 3,cont.

Now to combine the two sections, the easiest way is to load the BASIC program into your monitor (having typed the program in with BASIC. It will not run without the machine code though.) and load in the interrupt routine also. Now both programs are in memory and writing a tape output list to write to tape $4D22_h$ - $58AF_h$ will save the completed program. If you cannot do this, load both programs into BASIC by entering the machine code and then CLOADing the previously saved BASIC portion. Check to make sure the program runs. Now POKE 19586,57 and POKE 19587,29. This will change BASIC's tape output list to include the interrupt routine and CSAVEing will write to tape $4D22_h$ through $58AF_h$. For more information on tape output lists see the documentation accompanying the HI-LO Monitor by H. Holloway or the IOS ROM by R. Ferris.

NOTE- the method for saving machine language subroutines in Micro-Video's newsletter last spring does not work. Do not use their method of POKEing 19709 and 19710.

```
list

10 REM **FIRE3**
15 REM***COPYRIGHT 1981 BY KEVIN TENBROOK***
20 POKE19215,25:CLS
30 POKE19473,160:POKE19474,88
40 POKE24545,104:POKE24546,88
50 POKE24563,0:POKE24564,88
60 COLOR7,6,0,4
70 DIMVEL(1),ANGLE(1),AIR(1),DMG(1),SHIP(1)
80 CLEAR
90 LETSHIP(0)=15:SHIP(1)=95
100 PRINT"DO YOU WISH TO   PLAY AGAINST THE COMPUTER(1), OR"
110 PRINT"AGAINST ANOTHER  PLAYER(2)?"
120 GAME=VAL(INSTR$(1))-1:IFGAME<0ORGAME>1THEN100
130 GOSUB300
140 REM **MAINLINE LOOP**********
150 GOSUB500
160 FORI=0TO1
170 IFFIRE(I)=0ANDAIR(I)=0THENGOSUB700
180 IFAIR(I)=1ANDPEEK(22676+I*5)=0THEN800
190 NEXTI
200 IFGAME=0THENGOSUB1000
210 IFINT(RND(1)*50)=0THENGOSUB1200
220 GOTO140
230 CLS:IFI=0THEN260
235 IFGAME=0THEN245
240 PRINT"LEFT PLAYER WINS.":GOTO250
245 PRINT"I GUESS YOU WIN..LUCKY."
250 PRINT"CARE TO PLAY     AGAIN?":IFINSTR$(1)="Y"THENRUN
255 END
260 IFGAME=0THEN270
265 PRINT"RIGHT PLAYER     WINS.":GOTO250
270 PRINT"HA! HA!          I WIN!!!":GOTO250
300 CLS:B=USR(0)
310 FORI=0TO1
320 TX=5+I*55
325 VEL(I)=9:ANGLE(I)=5
330 FORK=13TO37STEP12
340 [illegible]
```

Fire 3,cont.

```
350 OUTPUTCHR$(6),J,K,3
355 NEXTJ
360 OUTPUT"0",TX,K,0:OUTPUT"9",TX+38,K,0
370 IFK=13THENOUTPUT"0",TX+44,K,0
390 NEXTK
400 OUTPUT"ANGLE",TX,7,0
410 OUTPUT"VELOCITY",TX,19,0
420 OUTPUT"DAMAGE",TX,31,0
422 GOSUB590
424 GOSUB630
430 NEXTI
440 FORJ=3TO37
450 PLOT56,J,3:PLOT57,J,3:NEXTJ
460 OUTPUTCHR$(2),SHIP(0)-3,43,2
470 OUTPUTCHR$(3),SHIP(1)-3,43,2
480 RETURN
500 FORI=0TOGAME
510 STK=JOY(I):KNOB=INT((POT(I)-3)/15):IFKNOB>9THENKNOB=9
520 IFSTK<>0THENGOSUB560
530 IFKNOB<>ANGLE(I)THENGOSUB610
540 NEXTI
550 RETURN
560 OUTPUTCHR$(1),12+I*55+VEL(I)*3,25,1
570 IFSTK=4ANDVEL(I)<9THENVEL(I)=VEL(I)+1
580 IFSTK=8ANDVEL(I)>0THENVEL(I)=VEL(I)-1
590 OUTPUTCHR$(1),12+I*55+VEL(I)*3,25,2
600 RETURN
610 OUTPUTCHR$(1),12+I*55+ANGLE(I)*3,13,1
620 ANGLE(I)=KNOB
630 OUTPUTCHR$(1),12+I*55+ANGLE(I)*3,13,2
640 RETURN
700 ADDR=22676+I*5:AIR(I)=1
710 XV=INT(VEL(I)*COS(ANGLE(I)/5.7296))+1:IFI=1THENXV=256-XV
720 YV=255-INT(VEL(I)*SIN(ANGLE(I)/5.7296))
730 POKEADDR+1,SHIP(I):POKEADDR+2,37
740 POKEADDR+3,XV:POKEADDR+4,YV
750 T=2*(256-YV)+1
760 SOUND1,1400
770 FORJ=0TO15:NEXTJ
780 SOUND1,515
790 POKEADDR,T:RETURN
800 TX=PEEK(22677+I*5):AIR(I)=0
810 IFABS(TX-SHIP(1-I))<4THEN890
820 IFABS(TX-SHIP(I))<4THENI=1-I:GOTO890
830 IFTX>110ORTX<4THEN190
840 FORJ=0TO2:SOUND1,8770
850 OUTPUTCHR$(5),TX-3,44,1
860 OUTPUTCHR$(5),TX-3,44,0:NEXTJ
870 SOUND7,4096
880 GOSUB1200:GOTO190
890 COLOR0,7,3,4:SOUND1,514
900 OUTPUTCHR$(4),TX-3,46,2
905 FORJ=0TO10:NEXTJ
910 OUTPUTCHR$(4),TX-3,46,0
920 COLOR7,6,0,4:SOUND1,515
930 I=1-I:OUTPUTCHR$(1),12+I*55+DMG(I)*3,37,1
940 DMG(I)=DMG(I)+4-ABS(TX-SHIP(I))
950 IFDMG(I)>9THEN230
```

Fire 3,cont.

```
960 OUTPUTCHR$(1).12+I*55+DMG(I)*3,37,2
970 GOSUB1200:GOTO170
1000 I=1:IFAIR(I)=1THENRETURN
1020 ANGLE(I)=4.5:M=2
1030 D=(2*M+2*RND(1)-1)*(SHIP(1)-SHIP(0))/(2*M)
1040 VEL(I)=(SQR(D/2)-1)/.707
1050 GOSUB700
1060 RETURN
1200 FORI=0TO1
1210 OUTPUTCHR$(2+I),SHIP(I)-3,43,0
1220 SHIP(I)=SHIP(I)+1-2*I
1230 OUTPUTCHR$(2+I),SHIP(I)-3,43,2
1240 NEXTI
1250 IFSHIP(1)-SHIP(0)<8THEN1270
1255 I=0
1260 RETURN
1270 COLOR0,7,3,4:SOUND1,514:FORJ=0TO10
1280 OUTPUTCHR$(4),SHIP(0),46,2
1290 OUTPUTCHR$(4),SHIP(0),46,0
1300 NEXTJ:SOUND1,515
1310 CLS:PRINT"VESSELS CRASHED. NO SURVIVORS."
1320 FORJ=0TO800:NEXTJ
1330 GOTO250
Ok
```

FIRE3 INTERRUPT ROUTINE

```
5800  3A EF 5F    LDA 5FEF        GET REAL TIME CLOCK VALUE
5803  E6 07       ANI 07          IF BOTTOM THREE BITS ARE ...
5805  C2 14 58    JNZ 5814        NOT ALL ZERO, THEN DON'T PROCESS DATA
5808  01 94 58    LXI B,5894      POINT AT PARAMETER TABLE FOR LEFT PLAYER
580B  CD 17 58    CALL 5817       PROCESS LEFT PLAYER DATA
580E  01 99 58    LXI B,5899      POINT AT PARAMETER TABLE FOR RIGHT PLAYER
5811  CD 17 58    CALL 5817       PROCESS RIGHT PLAYER DATA
5814  C3 72 01    JMP 0172        RETURN FOR END OF INTERRUPT CODE
5817  0A          LDAX B          GET TIME ALOFT VALUE FROM TABLE (T)
5818  B7          ORA A           IF T=0, THEN NO SHELL IN AIR...
5819  C8          RZ              SO DON'T PROCESS DATA
581A  3D          DCR A           DECREMENT REMAINING...
581B  02          STAX B          AND STORE BACK IN TABLE
581C  CD 3F 58    CALL 583F       FIND SCREEN ADDRESS AND PIXEL MASK
                                  FOR SHELL
581F  C2 26 58    JNZ 5826        JUMP IF SHELL IS NOT ON SCREEN
5822  1A          LDAX D          GET PIXEL MASK FOR SHELL
5823  2F          CMA             SET MASK FOR ERASING PIXEL
5824  A6          ANA M           ERASE SHELL FROM SCREEN
5825  77          MOV M,A
5826  03          INX B           POINT TO Y VELOCITY
5827  03          INX B
5828  0A          LDAX B          GET Y VELOCITY
                                  (NEGATIVE=UP, POSITIVE=DOWN)
5829  5F          MOV E,A
582A  3C          INR A           DOWNWARD ACCELERATION
582B  02          STAX B          SAVE MODIFIED Y VELOCITY BACK IN TABLE
582C  0B          DCX B
582D  0A          LDAX B          GET X VELOCITY
                                  (POSITIVE =RIGHT, NEGATIVE=LEFT)
```

Fire 3,cont.

```
582E  57          MOV D,A
582F  0B          DCX B          GET Y POSITION
5830  0A          LDAX B
5831  83          ADD E          ADD Y VELOCITY TO IT
5832  02          STAX B         STORE NEW Y POSITION IN TABLE
5833  0B          DCX B
5834  0A          LDAX B         GET X POSITION
5835  82          ADD D          ADD X VELOCTY TO IT
5836  02          STAX B         STORE NEW POSTION IN TABLE
5837  CD 41 58    CALL 5841      GET NEW SCREEN ADDRESS AND PIXEL MASK
                                 OF SHELL
583A  C0          RNZ            DONE IF SHELL NOT ON SCREEN
583B  1A          LDAX D         GET PIXEL MASK
583C  B6          ORA M          TURN ON PIXEL ON SCREEN
583D  77          MOV M,A
583E  C9          RET
583F  03          INX B          SET POINTER AT X POSITION
5840  0A          LDAX B         GET X POSITION
5841  03          INX B
5842  FE 70       CPI 70         IF GREATER THAN 70 THEN SHELL OFF SCREEN...
5844  D0          RNC            SO RETURN
5845  0F          RJE            DIVIDE X POSITION BY 4
5846  0F          RRC
5847  E6 3F       ANI 3F         MASK OFF JUNK BITS FROM DIVISION
5849  5F          MOV E,A        RESULT TO E (X ADJUSTMENT)
584A  16 40       MVI D,40       TOP OF SCREEN OFFSET
584C  0A          LDAX B         GET Y POSITION
584D  FE 4D       CPI 4D         IF GREATER THAN 40...
                                 THEN SHELL IS OFF SCREEN
584F  D0          RNC            SO RETURN WITHOUT ZERO SET
5850  6F          MOV L,A        Y POSITION
5851  26 00       MVI H,00
5853  29          DAD H          MULTIPLY Y POSTION TIMES 32
5854  29          DAD H          (NUMBER OF BYTES PER SCREEN LINE)
5855  29          DAD H
5856  29          DAD H
5857  29          DAD H
5858  19          DAD D          ADD X ADJUSTMENT AND SCREEN OFFSET
                                 TO GET ADDRESS
5859  0B          DCX B
585A  0A          LDAX B         GET X POSTION
585B  03          INX B
585C  E6 03       ANI 03         GET BOTTOM 2 BITS (PIXEL POSITION)
585E  11 4B 05    LXI D,054B     ADDRESS OF PIXEL MASK TABLE IN ROM
5861  B7          ORA A          IF PIXEL POSITION SHIFTED TO ZERO THEN...
5862  C8          RZ             DE POINTS AT CORRECT MASK,
                                 SO RETURN WITH ZERO SET
5863  13          INX D          IF NOT, POINT TO NEXT MASK...
5864  3D          DCR A          DEBUMP PIXEL POSITION...
5865  C3 61 58    JMP 5861       AND LOOP BACK
```

GRAPHICS CHARACTERS TABLE 5868 TO 5892

```
07 08 C0 C0 C0 C0 C0 00 00 10 38 FF FE 00 00 00
08 1C FF 7F 00 00 00 10 54 38 FE 38 54 10 10 38
BA 7C 38 00 00 80 80 80 80 80 00 00
```

Fire 3,cont.

PARAMETER TABLE FOR LEFT PLAYER 5894 TO 5898

00 3C 25 02 03

PARAMETER TABLE FOR LEFT PLAYER 5899 TO 589F

00 26 25 FB 06 00 00

PARTIAL CLEAR SCREEN ROUTINE 58A0 TO 58AF

21 C0 44 01 40 05 36 55 23 0B 79 B0 C2 A6 58 C9

ACCURATE DIGITAL CLOCK ERROR

I'm sure you've all read and benefited from Rich Pasco's articles and noticed how well they are done. Well, in his Accurate Digital Clock article I thought his listing was a little light, his printer uses a purple ribbon. Since once last year I got the newsletters back from the printers with an unreadable listing, I decided to re-type his program. Last year I had to have that page reprinted.

The point is I left out one line in the program. The effect of omitting this line is that the clock will run accurately at first, but floating point roundoff error will gradually increase, until after a day or so of operation the clock will be wildly inaccurate.

The missing line in the July - Sept. issue was

```
126  IF RT>255 THEN RT = RT - 256
```

I sincerely apologize to my readers and to Rich Pasco for both my mistake in typing and proofreading. I hope this does not reflect unfavorably on Rich's credibility as a valued contributor.

VEGA$

Rob Shumaker, Ph.D. 27200 Franklin Rd #212 Southfield,MI 48034

This program simulates a fifty cent slot machine. The computer randomly assigns the amount of money you start with (from $1.00 to $10.00) and gives you the option to continue or "cash in" following each play.

VEGA$ is my 1st "full length" program (if I got paid by the hour I'd probably be a millionaire!). I worked on this program prior to obtaining the 8K Fast Graphics BASIC, so either BASIC will suffice but improvements in the Graphics can certainly be made, especially with the new Fast Graphics. GOOD LUCK!

Editor's note - Glenn Darling also submitted a slot machine program to Interaction. Because there was not a significant difference in

Vega$, cont.

```
1 REM "VEGA$" BY ROB SHUMAKER,PHD NOV. 1980
2 REM-BYTES USED 4222
10 CLS:COLOR 4,2,3,7:FOR E=1TO10
20 OUTPUT "'VEGA$'",35,52,1:OUTPUT "IS A 50 CENT",20,42,2
30 OUTPUT "SLOT MACHINE!",17,32,2:NEXT E
40 COLOR 4,3,0,7
50 CLS:PRINT"   THIS CASINO":PRINT" REQUIRES YOU TO":PRINT" TYPE YOUR NA
ME"
60 PRINT"INTO THE COMPUTER":PRINT"  FOR A ROUTINE":PRINT"  CREDIT CHECK.
"
70 PRINT:PRINT CHR$(7)
80 INPUT A$:CLS
90 SOUND 3,12:OUTPUT "CREDIT CHECK",21,40,1
100 OUTPUT "IN PROGRESS!",21,33,1
110 FOR F=1TO1500:NEXT F:CLS:SOUND 7,4096
120 PRINT" YOU'RE CLEAR TO":PRINT"  PLAY 'VEGA$'!":PRINT:PRINT:PRINT:PRI
NT:PRINT
130 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
140 CLS:COLOR 4,1,7,2
150 FOR X=5TO105 STEP 12
160 OUTPUT "*",X,75,1
170 OUTPUT "*",X+6,75,2
180 OUTPUT "*",X,9,2
190 OUTPUT "*",X+6,9,1
200 NEXT X
210 FOR Y=9TO75 STEP 12
220 OUTPUT "*",5,Y,2
230 OUTPUT "*",5,Y+6,1
240 OUTPUT "*",105,Y,1
250 OUTPUT "*",105,Y+6,2
260 NEXT Y
270 OUTPUT LEFT$(A$,6),41,60,1
280 OUTPUT "NOW PLAYING",23,44,2
290 OUTPUT "VEGA$ !",36,28,3
300 FOR X=1TO5
310 FOR Y=1TO33
320 COLOR 4,1,7,3:NEXT
330 FOR Y=1TO33
340 COLOR 4,7,1,7:NEXT
350 NEXT X
360 CLS:COLOR 4,3,0,7
370 PRINT" 2 CHERRIES PAY:":PRINT"      $2.00":PRINT
380 PRINT"  2 LIMES PAY:":PRINT"      $1.00":PRINT
390 PRINT"  2 PRUNES PAY:":PRINT"      $0.00"
400 PRINT:FOR K=1TO1000:NEXT K
410 TONE 80,20:CLS
420 PRINT" 3 CHERRIES PAY:":PRINT"      $3.75":PRINT
430 PRINT"  3 LIMES PAY:":PRINT"      $2.25":PRINT
```

WANTED for radio amateur club - Used INTERACT computers - 8K or 16K - working or not - Send description and asking price, (incl. phone number and time we can contact you) to:

Glenn Manthey W60 N661 Jefferson Ave. Cedarburg, WI 53012

ega$, cont.

```
440 PRINT"  3 PRUNES PAY:":PRINT"      $1.50":PRINT
450 FOR K=1TO1000:NEXT K
460 TONE 80,20:CLS
470 PRINT"  THE COMPUTER":PRINT"RANDOMLY ASSIGNS":PRINT"YOU THE FOLLOWIN
G"
480 PRINT"AMOUNT TO GAMBLE":PRINT"      WITH:"
490 PRINT"($1.00 TO $10.00)
500 PRINT:PRINT:FOR V=1TO300:NEXT V
510 LET G=INT(10*RND(1))+1
520 LET M=G
530 PRINT:TONE 234,23:PRINT"      $";G
540 FOR V=1TO750:NEXT V
550 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:CLS
560 FOR H=1TO20
570 OUTPUT "GOOD LUCK,",7,39,1
580 OUTPUT LEFT$(A$,6),67,39,1
590 OUTPUT "!",103,39,1
600 NEXT H
610 CLS:COLOR 6,2,0,1
620 WINDOW 48
630 FOR X=20TO90
640 PLOT X,50,2
650 PLOT X,65,2
660 NEXT X
670 FOR Y=50TO65
680 PLOT 20,Y,2
690 PLOT 42,Y,2
700 PLOT 66,Y,2
710 PLOT 90,Y,2
720 NEXT Y
730 LET A=INT(3*RND(1))+1
740 LET B=INT(3*RND(1))+1
750 LET C=INT(3*RND(1))+1
760 OUTPUT "HIT FIRE BUTTON!",10,30,3
770 IF FIRE(0)=1 THEN 770
780 OUTPUT "HIT FIRE BUTTON!",10,30,0
790 SOUND 0,888
800 FOR P=1TO500:NEXT P
810 TONE 77,10
820 FOR Q=1TO40
830 OUTPUT "*",29,59,A
840 OUTPUT "'",30,61,2
850 NEXT Q
860 TONE 77,10
870 FOR Q=1TO40
880 OUTPUT "*",52,59,B
890 OUTPUT "'",53,61,2
900 NEXT Q
910 TONE 77,10
920 OUTPUT "*",76,59,C
930 OUTPUT "'",77,61,2
940 SOUND 7,4096
950 LET M=M-.5
960 IF A=B AND B<>C AND B=2 OR B=CAND B<>A AND B=2 THEN GOTO 1390
970 IF A=C AND A<>B THEN GOTO 1440
980 IF A=1 AND B=1 AND C<>1 OR B=1AND C=1 AND A<>1 THEN GOTO 1480
990 IF A=3 AND B=3 AND C<>3 OR B=3AND C=3 AND A<>3 THEN GOTO 1540
1000 IF A=B AND B=C THEN 1400
```

Vega$, cont.

```
1010 PRINT"NOT EVEN A PAIR!!"
1020 IF M=0 OR M<0 THEN 1890
1030 PRINT:PRINT"  YOU NOW HAVE":PRINT"     $";M
1040 FOR V=1TO750:NEXT V
1050 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1060 PRINT" TO CONTINUE---":PRINT"PUSH JOYSTICK UP."
1070 PRINT"<><><><><><><><><"
1080 PRINT"  TO CA$H IN---":PRINT" JOYSTICK DOWN."
1090 TONE 75,25
1100 IF JOY(0)=0 THEN 1100
1110 IF JOY(0)=4 THEN 610
1120 IF JOY(0)=4 THEN 1140
1130 IF JOY(0)=8 THEN 1140
1140 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1150 PRINT"YOU STARTED WITH:":PRINT"      $";G
1160 PRINT" YOU ENDED WITH:":PRINT"      $";M
1170 LET W=M-G
1180 IF M<G THEN GOSUB 1220
1190 IF M>G THEN 1270
1200 IF M=G THEN PRINT" YOU BROKE EVEN!"
1210 GOTO 1330
1220 PRINT"  YOU LOST"
1230 OUTPUT "$",69,17,1
1240 OUTPUT ABS(W),71,17,1
1250 OUTPUT "!",100,17,3
1260 GOTO1330
1270 PRINT"  YOU WON"
1280 FOR E=1TO6:READ A,B:TONE A,B:NEXT
1290 DATA 168,33,124,45,97,58,80,140,97,58,80,250
1300 OUTPUT "$",66,17,1
1310 OUTPUT W,67,17,1
1320 OUTPUT "!",100,17,3
1330 FOR S=1TO1500:NEXT S
1340 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT" COME BACK SO
ON,"
1350 PRINT TAB(6);LEFT$(A$,6)
1360 OUTPUT "!",81,17,3
1370 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1380 PRINT PORT("RESET",0)
1390 SOUND 4,4
1400 PRINT"  NO PAYOFF FOR":PRINT"    2 PRUNES!"
1410 PRINT:PRINT:PRINT:PRINT:PRINT
1420 SOUND 7,4096
1430 GOTO 1020
1440 SOUND 4,4
1450 PRINT"     CLOSE--":PRINT"  BUT NO CIGAR!"
1460 SOUND 7,4096
1470 GOTO 1020
1480 FOR E=1TO5
1490 PRINT CHR$(7)
1500 NEXT E
1510 PRINT"2 LIMES WIN $1 !"
1520 LET M=M+1
1530 GOTO 1020
1540 FOR E=1TO7
1550 PRINTCHR$(7)
1560 NEXT E
1570 PRINT"   2 CHERRIES":PRINT"   WIN $2.00 !"
```

Vega$, cont.

```
1580 LET M=M+2
1590 GOTO 1020
1600 OUTPUT "A",10,20,3
1610 OUTPUT "WINNER!",64,20,3
1620 FOR Q=1TO6
1630 FOR D=1TO3
1640 SOUND 3,552
1650 IF A=1 THEN OUTPUT "$2.25",25,20,1
1660 IF A=2 THEN OUTPUT "$1.50",25,20,1
1670 IF A=3 THEN OUTPUT "$3.75",25,20,1
1680 OUTPUT "$",10,59,1
1690 OUTPUT "$",97,59,1
1700 SOUND 7,4096
1710 NEXT D
1720 FOR U=1TO5
1730 IF A=1 THEN OUTPUT "$2.25",25,20,2
1740 IF A=2 THEN OUTPUT "$1.50",25,20,2
1750 IF A=3 THEN OUTPUT "$3.75",25,20,2
1760 OUTPUT "$",10,59,0
1770 OUTPUT "$",97,59,0
1780 NEXT U
1790 NEXT Q
1800 OUTPUT "A",10,20,0
1810 IF A=1 THEN OUTPUT "$2.25",25,20,0
1820 IF A=2 THEN OUTPUT "$1.50",25,20,0
1830 IF A=3 THEN OUTPUT "$3.75",25,20,0
1840 OUTPUT "WINNER!",64,20,0
1850 IF A=1 THEN M=M+2.25
1860 IF A=2 THEN M=M+1.5
1870 IF A=3 THEN M=M+3.75
1880 GOTO 1020
1890 PRINT" YOU LOST ALL OF":PRINT"   YOUR MONEY,":PRINT TAB(5);LEFT$(A$
,6)
1900 OUTPUT "!",75,17,3
1910 TONE 160,200
1920 FOR N=1TO25:NEXT N
1930 TONE 160,120
1940 FORV=1TO10:NEXT V
1950 TONE 120,375
1960 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1970 GOTO 1150
1980 END
Ok
```

CHRISTMAS TREE

This program was used to print the Christmas tree and greeting on the front page. It is written for use with the Slagh port and the P80I2 overlay. The cover picture differs slighty in that I used my Epson's fractional line spacing to compress the picture.

```
list

1 REM  *      CHRISTMAS  TREE      *
2 REM * INTERACT PROGRAM BY S. COOK *
5 PRINT"LETTER TO PRINT"
10 C$=INSTR$(1)
```

Christmas Tree, cont.

```
 15 PORT2
 20 FORN=1TO298
 30 READD
 40 IFD<0THEN80
 50 IFD>0THEN90
 60 PRINT:NEXT
 70 PORT1:END
 80 PRINTSPC(ABS(D));:NEXT
 90 FORP=1TOABS(D)
 92 PRINTC$;
 94 NEXT:NEXT
100 DATA-39,1,0,-37,5,0,-35,9,0,-32,15,0,-29,21,0,-31,17,0,-29,21,0
110 DATA-25,29,0,-21,37,0,-24,31,0,-28,23,0,-30,19,0,-26,27,0,-23,33,0
120 DATA-20,39,0,-16,47,0,-13,53,0,-17,45,0,-21,37,0,-23,33,0,-18,43,0
130 DATA-12,55,0,-15,49,0,-21,39,0,-18,43,0,-16,47,0,-13,53,0,-10,59,0
140 DATA-5,69,0,-7,65,0,-9,61,0,-12,22,-4,3,-4,22,0,-17,15,-6,3,-6,15,0
150 DATA-38,3,0,-38,3,0,-38,3,0,34,-1,9,-1,34,0,2,-1,3,-1,2,-4,2,-4,2,-4
160 DATA2,-1,3,-1,2,-1,9,-1,2,-1,3,-1,3,-1,3,-1,4,-4,4,-5,2,0,2,-2,1,-2,2
170 DATA-1,5,-1,2,-1,2,-1,2,-1,3,-1,1,-1,4,-1,7,-1,4,-1,1,-1,4,-2,1,-2,3
180 DATA-1,4,-1,3,-1,6,0,2,-1,3,-1,2,-3,3,-4,2,-4,4,-1,5,-1,7,-1,5,-1,5
190 DATA-1,3,-1,3,-6,3,-5,2,0,2,-1,3,-1,2,-1,5,-1,1,-1,3,-1,1,-1,5,-1,6
200 DATA-1,5,-1,5,-1,1,-1,4,-1,3,-1,3,-1,4,-1,7,-1,2,0,2,-1,3,-1,2,-4,2
210 DATA-1,2,-1,2,-1,2,-1,4,-1,6,-1,5,-1,4,-1,3,-1,3,-1,3,-1,3,-1,4,-1,3
220 DATA-5,2,0,36,-1,5,-1,36,0
Ok
```

WARI

by Fred Schlesinger 29 New England Lake Hiawatha, NJ 07034

WARI or AWARI is an ancient game probably from Egypt or Arabia and has been played for over 3500 years. It is part of a group of pit and pebble games termed mancala games by anthropologists. Another variation called KALAH has been programmed by Bob Draganski and should be published next year in INTERACTION.

WARI is played against the computer. The board consists of two rows of 6 boxes numbered 1 to 12. Your side is numbered 1 to 6. At the beginning of each game, each box contains 4 pebbles. All pebbles (actually just numbers) are used by both the computer and you.

A turn consists of picking up the pebbles from one of the boxes and sowing (placing) them one at a time in each box moving counter clockwise (toward the higher numbered boxes). The turn ends when all of the pebbles from the starting box are used. If 12 or more pebbles had been accumulated, you skip the starting box and continue around. All of this is accomplished just by picking the starting box number.

If the last pebble dropped makes the total in that box equal 2 or 3 (and the box is located on hte computer's side in your case) they are captured and added as your score. Additionally, if the next to last box also has 2 or 3 they are captured and so are each consecutively, preceding boxes having 2 or 3.

Wari, cont.

The object is to capture over half of the 48 pebbles. The first person or computer to capture more than 24 pebbles wins. By typing H the computer will pick your best move. By typing S or SAVE you can save the game data on tape and continue playing later.

A few special rules are that you cannot capture all the computer's pebbles. Additionally, if the computer uses all its pebbles when sowing, you must sow pebbles into its boxes. The computer must also observe these rules. If no legal move can be made, the game is a tie. WARI is a game of skill and the computer takes 5 - 60 seconds to make its move. You can expect to lose most of the games at first.

```
list

 10 REM WARI
 15 PRINTCHR$(8)
 17 OUTPUT CHR$(1),10,10,1
 20 CLS
 25 PRINT"      WARI":COLOR0,1,2,3
 100 CLEAR200:J=1:K=1:Q=14:P=13:F=50:D=12
 120 IF R$="Y" THEN CLS:GOTO140
 130 DIMT(Q),Y(Q)
 131 DIM W(Q),V(6),E(6),B(Q)
 140 ZB=RND(0):ZB=ZB/Q:ZA=.25+ZB:ZB=.25-ZB:GOSUB750
 145 WINDOW18:MN=1:PRINT"TAPE INPUT?"
 146 R$=INSTR$(1):IF R$="Y"THEN GOSUB990:GOSUB4000:GOTO148
 147 FOR J=1TOD:B(J)=4:NEXT:B(P)=0:B(Q)=0
 148 :GOSUB990:GOSUB900
 150 REM THE GAME   BEGINS
 160 INPUT"WANT TO START";R$
 170 GOSUB990:R$=LEFT$(R$,1)
 175 IF R$="Y"THEN 250
 180 IF R$<>"N" THEN PRINT"WHAT":GOSUB990:GOTO160
 190 PRINT"I'M THINKING":GOSUB510
 195 IF M<1 THEN 2000
 200 PRINT"MY MOVE IS ";M
 210 FOR J=1TOQ:T(J)=B(J):NEXT:GOSUB350
 220 FOR J=1TOQ:B(J)=T(J):NEXT:GOSUB900
 230 IF B(Q)<24 THEN 250
 240 PRINT"I WIN":GOTO 810
 250 PRINT"MOVE NO.";MN:INPUT"YOUR MOVE";R$
 252 IF R$="M" THEN GOSUB4500:GOTO250
 253 IF R$="S" OR R$="SAVE"THEN GOSUB4100:GOTO250
 255 IFR$="H"THENGOSUB6100:GOSUB510:PRINT"TRY MOVING ";(M-6):GOSUB6200:GO
 TO250
 260 M=INT(VAL(R$)):IF M<1ORM>6 THEN 330
 270 FOR J=1TOQ:T(J)=B(J):NEXT
 280 GOSUB350:IFM<0THEN 330
 290 FOR J=1TOQ:B(J)=T(J):NEXT
 300 MN=MN+1:GOSUB900
 310 IF B(P)<24 THEN 190
 320 PRINT"YOU WIN":GOSUB990:GOTO 810
 330 GOSUB990:PRINT"ILLEGAL MOVE":FOR J=1TO400:NEXT
 332 GOSUB990:PRINT"M FOR MENU"
 335 FOR J=1TO400:NEXT:GOTO250
 350 IF T(M)=0 THEN M=-1:RETURN
```

Wari, cont.

```
360 R$="H":IF M>6 THEN R$="C":GOTO 380
370 FOR J=1TO Q:Y(J)=T(J):NEXT:GOTO400
380 FOR J=1TO6:Y(J)=T(J+6):Y(J+6)=T(J):NEXT
390 Y(P)=T(Q):Y(Q)=T(P):M=M-6
400 C=M:N=Y(C):FOR J=1TON:C=C+1
410 IF C=P THENC=1
420 IF C=M THENC=C+1:GOTO 410
430 Y(C)=Y(C)+1:NEXT:Y(M)=0:L=C
440 IF L<7ORY(L)>3ORY(L)<2 THEN 460
450 Y(P)=Y(P)+Y(L):Y(L)=0:L=L-1:GOTO440
460 S=0:FOR J=7TOD:S=S+Y(J):NEXT
470 IF S=0THENM=-2:RETURN
480 IF R$="H"THEN FOR J=1TOQ:T(J)=Y(J):NEXT:RETURN
490 FORJ=1TO6:T(J)=Y(J+6):T(J+6)=Y(J):NEXT
500 T(Q)=Y(P):T(P)=Y(Q):RETURN
510 FOR A=1TO6:M=A+6:IF B(M)=0THENE(A)=-F:GOTO690
530 FOR J=1TOQ:T(J)=B(J):NEXT:GOSUB350
540 IF M<0THENE(A)=-F:GOTO690
550 IFT(Q)>23 THEN M=A+6:RETURN
560 FOR J=1TOQ:W(J)=T(J):NEXT:FOR K=1TO6
570 IF T(K)=0THENV(K)=F:GOTO670
580 FOR J=1TOQ:T(J)=W(J):NEXT:M=K:GOSUB 350
590 IF M<0THENV(K)=F:GOTO670
600 FA=0:FB=0:FC=0:FD=0:FORJ=7TOD
610 FB=FB+T(J):IF T(J)>0THENFA=FA+1
620 IF T(J)<3THEN FC=FC+1
630 IF T(J)>FD THEN FD=T(J)
640 NEXT:FE=FB:FOR J=1TO6:FE=FE+T(J):NEXT
650 FA=FA/6:FD=1-FD/FB:FC=1-FC/6:FB=FB/FE
660 V(K)=ZA*(FA+FB)+ZB*(FC+FD)+T(Q)+B(P)-B(Q)-T(P)
670 NEXT:E(A)=F:FORJ=1TO6:IF V(J)<E(A) THEN E(A)=V(J)
680 NEXT
690 PRINT"STILL THINKING":NEXT:PRINT"DONE":M=0:FA=-F:FOR J=1TO6
700 IF E(J)>FA THEN FA=E(J):M=J+6
710 NEXT:RETURN
750 FORI=5TO87STEP13
751 FORH=25TO65
752 PLOT I,H,2
753 NEXT:NEXT
760 FORI=5TO83
761 FORH=26TO64STEP19
762 PLOTI,H,2
763 NEXT:NEXT
769 C=0
770 FORI=3TO70STEP13
771 C=C+1
772 OUTPUTC,I,24,1
773 NEXT
774 FORI=70TOOSTEP-14
775 C=C+1
776 OUTPUTC,I,70,1
777 NEXT
780 OUTPUT"ME",90,60,1:OUTPUT"YOU",87,40,1
790 RETURN
810 WINDOW24:IF MN<10 THENPRINT"SHORT AND SWEET"
811 IF MN=>10 AND MN<=30 THEN PRINT"GOOD GAME-        VERY CHALLENGING"
812 IF MN>30 AND MN<=100 THEN PRINT"GOOD GAME--------A TRIFLE LONG"
813 IF M>100 THEN PRINT"WHEW! I'M POOPED YOU'RE OK"
```

Wari, cont.

```
814 IF MN>300 THEN FORI=ITO3000:NEXT:PRINT"DAMN GOOD,AS A    MATTER OF F
ACT"
830 FOR I=1TO1000:NEXT
840 INPUT"WANT TO TRY AGAIN";R$
850 R$=LEFT$(R$,1):IF R$="Y" THEN120
860 IF R$<>"N" THEN840
870 PRINT"SEE YOU LATER":FOR J=1TO1000:NEXT:WINDOW77:CLS:PRINT"BYE":END
900 GOSUB3000
901 K=0
905 FORJ=1TO65STEP13
906 FORJ=1TO66STEP13
910 K=K+1
915 OUTPUT B(K),J,40,1
920 NEXT
925 FORJ=66TO0STEP-13
930 K=K+1
940 OUTPUT B(K),J,60,1
945 NEXT
950 OUTPUT B(Q),85,50,1
955 OUTPUT B(P),85,30,1
960 RETURN
990 PRINT:PRINT:RETURN
2000 GOSUB990:PRINT"NO LEGAL MOVES":GOTO840
3000 CL$=CHR$(1)
3010 FORJ=6TO71STEP13
3020 OUTPUTCL$,J,40,0
3022 OUTPUTCL$,J+4,40,0
3024 OUTPUTCL$,J+7,40,0
3026 OUTPUTCL$,J,60,0
3028 OUTPUTCL$,J+4,60,0
3030 OUTPUTCL$,J+7,60,0
3035 NEXT
3040 FOR J=85 TO 100 STEP 6
3041 OUTPUT CL$,J,50,0
3042 OUTPUT CL$,J,30,0
3043 NEXT
3080 RETURN
4000 PRINT"LOADED?...":R$=INSTR$(1)
4010 IF R$<>"Y"THEN4000
4015 PRINT"WORKING
4020 CLOAD*B
4030 PRINT"DONE"
4040 RETURN
4060 RETURN
4100 PRINT"Y TO SAVE":R$=INSTR$(1)
4110 IF R$<>"Y" THEN RETURN
4120 PRINT"LOADED?...":R$=INSTR$(1)
4130 IF R$<>"Y" THEN 4120
4140 PRINT"WORKING":CSAVE*B
4145 PRINT"BOARD SAVED":FOR I=1TO200:NEXT
4150 PRINT"CONTINUE ?":R$=INSTR$(1)
4160 IF R$<>"Y" THEN GOTO810
4170 RETURN
4500 PRINT"S=SAVE":FOR J=1TO400:NEXT
4510 PRINT"H=HELP":FOR J=1TO400:NEXT
4550 RETURN
6000 FOR I=1TO10
6005 PRINTCHR$(1)
```

Wari, cont.

```
6006 PRINTI
6007 PRINTCHR$(I)
6010 NEXT
6100 FOR J=1TO6:T(J)=B(J+6):T(J+6)=B(J):NEXT
6120 FOR J=1TO12:B(J)=T(J):NEXT
6130 RETURN
6200 GOSUB6100
6205 FOR I=1TO1000:NEXT
6210 RETURN
Ok
```

THE TWELVE DAYS OF CHRISTMAS

This short program prints out a copy of the traditional Christmas carol. The program is written for use with the Slagh port and the P8012 overlay. With this overlay PORT 2 transfers all print statements to the port. PORT 1 returns all output to the screen. PORT 3 if it had been used would output to both screen and printer.

```
1 REM *TWELVE DAYS OF CHRISTMAS*
2 REM *INTERACT PROGRAM BY S. COOK*
3 REM *USING SLAGH PORT AND EPSON PRINTER*
10 DIMA$(12),B$(12)
15 PORT2  :  REM  PRINTER ON FOR SLAGH PORT
30 PRINTCHR$(14);TAB(4);"THE TWELVE DAYS OF CHRISTMAS"
40 PRINT:PRINT
50 FORI=1TO12
60 READA$(I),B$(I)
70 NEXTI
80 C$=""
90 FORI=1TO12
100 PRINT"ON THE ";B$(I);" DAY OF CHRISTMAS"
110 PRINT"MY TRUE LOVE GAVE TO ME"
120 FORJ=ITO1STEP-1
130 IFJ<>1THEN170
140 PRINTC$;A$(J)
150 C$="AND "
160 GOTO180
170 PRINTA$(J)
180 NEXTJ
190 PRINT
200 NEXTI
299 PORT1
300 DATAA PARTRIDGE IN A PEAR TREE,FIRST,TWO TURTLE DOVES,SECOND
310 DATATHREE FRENCH HENS,THIRD,FOUR CALLING BIRDS,FOURTH
320 DATAFIVE GOLD RINGS,FIFTH,SIX GEESE A-LAYING,SIXTH
330 DATASEVEN SWANS A-SWIMMING,SEVENTH,EIGHT MAIDS A-MILKING,EIGHTH
340 DATANINE DRUMMERS DRUMMING,NINTH,TEN PIPERS PIPING,TENTH
350 DATAELEVEN LADIES DANCING,ELEVENTH,TWELVE LORDS A-LEAPING,TWELFTH
Ok
```

HORSE RACE

Glenn Darling 1111 Barnes Rd Mason, MI 48854

This is one of very few games that more than two people can play at once on the Interact. The program can accommodate up to 10 people, making it ideal for parties, etc. HORSE RACE was extended and adapted from a program by Larry Friedman and originally appeared in a magazine called HOBBY COMPUTER HANDBOOK.

```
list

2 COLOR 0,1,2,4:PRINTCHR$(8):POKE19215,25
4 DATA 168,30,124,41,97,52,80,63,80,30,80,30,80,63,97,52,97,25,97,25
5 DATA 97,52,124,41,97,52,124,41,168,60,168,30,124,41,97,52,80,63
6 DATA 80,40,80,40,80,73,97,62,124,50,168,38,168,26,168,26,168,38,124,90
7 DATA 168,40,168,20,168,40,168,20,168,40,168,20,168,40,168,20,110,50
8 DATA 131,25,110,50,131,25,110,50,131,25,168,350
9 CLEAR 200
10 DEF FNRAN(G)=INT(RND(1)*G)+1
11 CLS
14 FOR G=1TO88:READ Z:NEXT G
15 DATA "WIN","PLACE","SHOW","SLOW","SICK"
20 FOR G=1TO5:READ S$(G):NEXTG
35 F=1:RESTORE
38 FOR G=1TO58:READ Z:NEXT G
40 FOR G=1 TO 15
42 READ O:READ Q
44 TONE O,Q:NEXTG
55 H$=CHR$(19):B$=CHR$(1)
150 DATA 32,42,52,62,72
155 FOR G=1TO5:READ Z$(G):NEXTG
160 FOR I=1TO5:READ D(I)
165 NEXT I
170 T1=0
200 R=0:DIM A$(5)
220 DATA "ASTRO TURF ","BLACK BART ","CHESTNUT   ","DILLY FILLY"
222 DATA "EVIL EGBERT"
230 FORI=1TO5:READA$(I):NEXTI
260 GOSUB 1400
275 PRINT:CLS
282 COLOR 3,2,1,0
285 PRINT "WELCOME TO"
290 PRINT "DARLING DOWNS"
292 PRINT
294 PRINT "TODAYS HORSES ARE"
295 PRINT
296 PRINT "(A)  ";A$(1)
298 PRINT "(B)  ";A$(2)
300 PRINT "(C)  ";A$(3)
302 PRINT "(D)  ";A$(4)
304 PRINT "(E)  ";A$(5)
370 FORI=1TO1000:NEXT I:CLS
374 PRINT:COLOR 7,4,1,2
376 PRINT "EACH PLAYER":PRINT "BEGINS WITH $ 50":PRINT
378 PRINT "BETS ARE LIMITED":PRINT"TO TEN DOLLARS":PRINT
380 FOR D=1TO5:P(D)=0:NEXT D
390 IF U$="N"THEN 420
```

Horse Race, cont.

```
420 R=1
430 PRINT "RACE NUMBER ";R:T=0:F=1
437 FOR I=1TO5
438 F$(I)="X":P(I)=0
450 NEXT I
455 FOR I=1TO5
460 R(I)=FNRAN(3)+2
470 NEXT I
480 PRINT "HORSE        ODDS":PRINT
490 FOR I=1TO5
492 PRINT A$(I);R(I);"-1"
494 NEXT I
495 T$(A) = A$(1)
496 T$(B) = A$(2)
497 T$(C) = A$(3)
498 T$(D) = A$(4)
499 T$(E) = A$(5)
500 PRINT:PRINT:PRINT:PRINT:PRINT:WINDOW35
514 FOR G = 1 TO H
516 PRINT N$(G);"..."
517 PRINT "ENTER THE FIRST"
518 PRINT "LETTER OF THE "
519 PRINT "HORSE YOU BET ON"
520 INPUTL$(G)
521 IFL$(G)<"A"ORL$(G)>"E"THEN516
522 C(G)=ASC(L$(G))-64
528 PRINT "ENTER AMOUNT"
529 PRINT "OF BET"
530 INPUT B(G)
532 IFB(G)>100RB(G)<1THEN528
534 PRINT N$(G);" BETS $";B(G)
536 PRINT "ON:";A$(C(G))
537 FOR J=1TO 500:NEXT J
538 NEXT G
540 WINDOW 77:CLS:R2=1
675 COLOR 2,4,1,0
680 FOR L=16481 TO 16507
690 POKEL,170:NEXTL
710 FOR L=18497TO18523
720 POKEL,170:NEXTL
730 FOR L=16482 TO 18498 STEP 32
740 POKEL,20:NEXTL
751 OUTPUT "A",4,72,2
752 OUTPUT H$,12,62,3
753 OUTPUT "B",4,62,2
754 OUTPUT H$,12,52,3
755 OUTPUT "C",4,52,2
756 OUTPUT H$,12,42,3
757 OUTPUT "D",4,42,2
758 OUTPUT H$,12,32,3
759 OUTPUT "E",4,32,2
760 OUTPUT H$,12,72,3
764 RESTORE
766 FOR G=1TO29
768 READ O:READ Q
770 TONE O,Q:NEXTG
774 FOR G=32 TO 72 STEP 10
775 OUTPUT B$,12,G,0:NEXT G
```

Horse Race, cont.

```
777 OUTPUT "AND THEY'RE OFF!",14,8,1
780 FOR I=1TO5
781 SOUND 3,16436
790 R1(I)=FNRAN(R(I)+1)
800 NEXT I
805 SOUND 7,4096
810 FOR I=1TO5
815 SOUND 3,16436
820 P(I)=P(I)+INT(12/R1(I))
825 IF P(I)>94 THEN P(I)=94
828 IF T=5 THEN 831
830 GOTO 832
831 OUTPUT B$+B$+B$+B$+B$+B$+B$+B$+B$+B$+B$+B$+B$+B$+B$+B$,14,8,0
832 IF P(I)<>94 THEN 845
833 IF F$(I)="Y"THEN 845
834 IF F=1THENOUTPUT A$(6-I)+" WINS",4,8,1
837 F$(I)="Y"
838 OUTPUT S$(F),12,D(I),1
839 F=F+1:J(6-I)=F-1
845 T=T+1:SOUND7,4096:NEXTI
850 FOR J=1TO5
855 SOUND 3,16436
860 IF R2 =1 THEN P(J)=P(J)+3
865 OUTPUT B$,K(J)+18,D(J),0
870 OUTPUT H$,P(J)+18,D(J),3
910 K(J)=P(J):SOUND7,4096
919 IF F>=6THEN950
920 NEXT J
930 R2=0:GOTO 780
950 FOR I=1TO 800:NEXT I
952 CLS
953 FORI=1TOH:PRINT:IFJ(C(I))>=4THEN1100
957 F=1
960 IF J(C(I))=1THEN F=4
965 IF J(C(I))=2THEN F=2
967 AM=F*R(C(I))*B(I)
970 PRINT N$(I);" WINS $";AM
980 GOTO 1200
1100 PRINT N$(I);" LOSES $";B(I)
1150 AM=-B(I)
1200 PRINT N$;"NOW HAS $";W(I)+AM
1205 IF W(I)<0THENPRINT"YOU OWE US.PAY UP"
1210 W(I)=W(I)+AM
1245 NEXTI
1250 R=R+1:COLOR7,1,2,4
1300 PRINT "ANOTHER RACE ?     (Y/N)"
1305 INPUTV$:IFV$="N"THEN9999
1315 CLS
1320 PRINT "SAME PLAYERS      (Y/N)"
1325 INPUTU$:IFU$="Y"THEN430
1330 IF U$="Y"THEN 430
1335 GOSUB 1400
1340 GOTO 430
1400 PRINT "MORE THAN ONE"
1401 H=1:PRINT "PLAYER (Y OR N)"
1404 PRINT:PRINT:INPUTU$
1410 IFU$="N"THEN1420
1412 PRINT "HOW MANY PLAYERS"
```

Horse Race, cont.

```
1414 PRINT:INPUTH
1418 IF H<0 OR H>10 THEN 1412
1420 PRINT:PRINT
1424 FOR I=1TOH
1425 W(I)=50
1426 PRINT "PLAYER ";I;"        ENTER YOUR NAME"
1428 INPUTN$(I):PRINT:NEXTI
1499 RETURN
1500 GOTO 430
9999 WINDOW 77
Ok
```

32K PROGRAMS

Subscribers have been asking whether Interaction will publish 16K programs for use with MicroVideo's memory expansion. Currently no one has submitted any such programs and I don't have an expanded unit yet. Another problem is size. A current program listing using 4.5K of memory takes up about 4 pages of space. A 16K program might take over 12 pages of space. This would be a lot of useless listing for owners of unexpanded units. Also 16K is a lot of typing for you.

Possibly we can give a program description and instructions only, the listing being available upon request with a stamped self addressed envelope and/or the program tape from George Leggett or the program author if they so desire. Please send me any suggestions on this.

DICE

by Vincent Chobot 2716 s. Cuyler Berwyn, IL 60402

In response to my comments on the slow dice routine in PETALS AROUND THE ROSE in last December's INTERACTION, Vince has sent me this dice demonstration program, as well as, two other programs of dice games which will probably be published in an issue next year. This routine is fast enough to fill the screen with 99 dice in under 30 seconds. This program is just a demonstration of how to use his dice routine, though it can be used as a set of computerized dice by selecting option 2. The real purpose though, is for you to write your own program using dice. Vince recommends a book "The Way To Play - The Illustrated Encyclopedia of the Games of the World" by the Diagram Group, published by Paddington Press Ltd. He used this book for the dice games in his other programs.

list

```
5 DIMD(11):PRINTCHR$(8)
10 CLS:COLOR2,7,0,3
20 PRINT:PRINT"1-FILL SCREEN     2-THROW DICE     3-CHARACTERS
30 A$=INSTR$(1)
40 ONVAL(A$)GOTO100,200,300
50 GOTO30
```

Dice, cont.

```
100 CLS:X1=5:H=11
110 FORY1=73TO9STEP-8
120 GOSUB2010
130 NEXT:FORX=1TO1000:NEXT:GOTO10
200 PRINT:PRINT:PRINT"AT PAUSE:TO STOP HIT 'S' KEY, TO  CHANGE 'C', FOR
 SAME ANY KEY
210 PRINT:PRINTCHR$(35)" DICE,X,Y";:INPUTH,X1,Y1:CLS
220 GOSUB2000
230 A$=INSTR$(1)
240 IFA$="S"GOTO10
250 IFA$="C"GOTO210
260 GOTO220
300 CLS:PRINT"ENTER ANY 2 CHAR-ACTERS":A$=INSTR$(2)
310 CLS:H=1:OUTPUTA$,0,5,1
320 FORL=0TO10
330 FORK=1TO5:IFPOINT(L,K)=1THENGOSUB350
340 NEXTK,L
345 A$=INSTR$(1):GOTO300
350 X1=8*L+20
360 Y1=8*K+20
370 GOSUB2010:RETURN
2000 FORI=1TOH:D(I)=0:NEXT
2005 FORX=X1TOX2+20STEP5:FORY=Y1TOY1+2:OUTPUTCHR$(1),X,Y,0:NEXTY,X
2010 FORI=1TOH
2020 D(I)=INT(6*RND(1)+1)
2025 Y2=Y1-3:Y3=Y1-1:Y4=Y1+1
2030 X2=X1+(I-1)*10:X3=X2+1:X4=X2+5
2100 FORX=X2TOX2+2STEP2
2110 FORY=Y1TOY1+2STEP2
2120 OUTPUTCHR$(1),X,Y,1
2130 NEXTY,X
2140 OND(I)GOTO2170,2160,2160,2150,2150,2145
2145 PLOTX3,Y3,2:PLOTX4,Y3,2
2150 FORY=Y2TOY4STEP4:PLOTX3,Y,2:PLOTX4,Y,2:NEXT
2155 OND(I)-3GOTO2180,2170,2180
2160 PLOTX3,Y4,2:PLOTX4,Y2,2
2165 IFD(I)=2THEN2180
2170 PLOTX3+2,Y3,2
2180 IFI=HTHENRETURN
2190 NEXT
Ok
```

CHARACTER LIMITATIONS

Robert H. Murphy 109 Poplar St. Conneaut, OH 44030

This is a handy little routine that could very easily be incorporated into not only strings like inputting names but also into data statements as well. It was developed to overcome some limitations of the 17 character line.(Note - This cannot make more than 17 characters, it only makes some kind of sense out of what is displayed.)

Character Limitations, cont.

```
10 INPUT A$
20 IF LEN(A$)<=17 THEN PRINT A$
30 X=LEN(A$)-17
40 IF LEN(A$)>17 THEN PRINT LEFT$(A$,16);"-";RIGHT$(A$,X+1)
```

Example:

```
? BOB
BOB

? ROBERT HAROLD MURPHY
ROBERT HAROLD MU-
RPHY
```

HOLIDAY SUPRISE

This listing of data statements will print another holiday picture Simply substitute these lines for the data statements in the Christmas Tree program and change line 20 to `20 FOR N = 1 TO 194`

```
list100

100 DATA-33,1,-2,1,0,-27,2,-3,1,-2.1,0,-21,2,-5,2,-2,1,-2,1,0,-23,2,-5,2
110 DATA-2,1,-4,1,-3,1,0,-25,2,-6,3,-1,2,-2,2,0,-22,2,-3,3,-6,2,-2,1,-5,2
120 DATA0,-24,10,-2,2,-2,1,-3,2,0,-30,9,-2,1,-1,2,0,-27,3,-7,7,0,-17,2
130 DATA-15.12,-2.3,0,-17,2,-19,12.0,-17,8,-14,9,0,-16,12,-11,5,0,-16,14
140 DATA-9,6,0,-15,17,-8,7,0,-15,19,-7,7.0,-14,23,-2,9,0,-12,6,-2,28,0,-7
150 DATA7,-4,3,-4,23,0,-5,2,-5,6.-10,19,0,-4,2,-3,4,-18,16,0,-2,2,-3,2
160 DATA-26,14,0,2,-3,2,-31,6,-3,4,0,2,-1,2,-36,4,-4,2,0,-3,2,-39,2,-5,2
170 DATA0,-46,2,-4,2,0,-47,2,-4,2,0,-48,2,-4,2,0,-50,2,-4,2,0,-51,2,0
Ok
```

★★★

1981 SURVEY AND QUESTIONAIRE

Do you have or anticipate getting a printer?

If yes, what kind?

Would you like more printout programs like Christmas Tree?

Do you have a 32K MicroVideo Expansion?

Would you like to see 16K programs in Interaction?

What was your favorite program in the 1981 Interactions?

LUNAR LANDER was the 1980 favorite by the way

What one thing was the most wrong about Interaction in 1981?

To renew your subscription for 1982 please send -

- O $15.00 USA subscription (bulk mail delivery)
- O $20.00 Canada and Mexico (first class mail)
- O $25.00 all other foriegn countries (air mail - small packet delivery)

Please make your checks payable to
STEPHEN COOK Interaction Newsletter

Canadians - please mark your checks payable in US Funds
or send a Postal money order in US Funds
Overseas subscribers - an international money order or bank draft
is the easiest way for me to receive funds.

Send your subscription to
STEPHEN COOK Interaction Newsletter 15356 Prevost Detroit, MI 48227