MICRO *M* VIDEO ™

EZEDIT PROGRAM EDITOR
Program Guide

INTRODUCTION

EZEDIT, Micro Video's program editor, helps you get your Level II BASIC
programs up and running faster. Few people can type in an entire program
without making mistakes that cause syntax errors; correcting them can be
time-consuming and frustrating. EZEDIT simplifies the process of entering
and editing your programs. Its commands are easy-to-use and have names
that are English words you might use to describe the functions they perform.
For example, the LOCATE command finds a number, word, or group of words in
you program quickly—you don't have to spend time listing your program and
searching through the lines. The SUBSTITUTE command allows you to make
changes in your program lines with single commands—no more retyping whole
lines to correct typographical errors!

LOADING THE PROGRAM

Insert the EZEDIT program tape into the cassette deck. Press the RESET
button, then the "L" key. Press the REWIND button on the cassette deck to
rewind the tape if necessary. Press the READ button on the cassette deck
to load the program into the computer.

CREATING NEW PROGRAMS

To enter a new program after loading the EZEDIT tape or saving your current
program on tape with the CSAVE command, type the NEW command. This clears
the memory for entry of a new program. Then type in the program lines just
as you would using Level II BASIC, or use the AUTO command to number your
program lines automatically.

EDITING PROGRAMS STORED ON TAPE

Use the CLOAD command to load your program into the computer for editing.

EDITING PROGRAMS LOADED FROM BASIC

If you have loaded the Level II BASIC cassette and entered a program from a
data tape or from the keyboard, simply load EZEDIT as instructed above. DO
NOT TYPE A "NEW" COMMAND! Your BASIC program is still in memory and can be
edited.

RUNNING EDITED PROGRAMS

After you are finished creating or editing your program, press the RESET button and load the Level II BASIC tape as you would normally. DO NOT TYPE A "NEW" COMMAND IN BASIC! Your edited program is still in memory and can be run. Note that you can use EZEDIT or Level II BASIC in any order without reloading your program, but it's always a good idea to save your program on tape before switching between programs.

SAVING PROGRAMS ON TAPE

You may save any program on tape that you have been working on using EZEDIT. Use the CSAVE command to write a program to tape.

ABBREVIATING EZEDIT COMMANDS

The individual EZEDIT commands are described below and on the following pages. All EZEDIT commands may be abbreviated as long as the abbreviation uniquely identifies the command. Use a period at the end of any abbreviated command word, for example, LI., SUB., LOC.

COMMAND DEFAULTS

Some commands have several forms. All forms for each command are shown. The shorter forms use default values for the items specified in the longer forms. That is, if you do not specify a value for an item when you type the command, a value is assumed for that item. Default values are identified in each command description as appropriate.

                              EZEDIT COMMANDS

AUTO
AUTO n
AUTO n,m

Automatically assigns and prompts you with line numbers for program lines after you type the AUTO command. To stop automatic line numbering, press only the "CR" key when a line number appears on the screen. The "n" parameter gives the line number at which automatic numbering is to begin; the "m" parameter gives the increment between line numbers. The default value for both "n" and "m" is 10. That is, line numbers 10, 20, 30, and so on are assigned to the program lines as you type them in. Please note that 65529 is the largest legal line number in BASIC. Also note that any existing program lines with the same line numbers as assigned by the AUTO command are overwritten by what you type in.

    AUTO 50,5   — starts automatic numbering at line 50, and adds lines in
                       increments of 5, that is 50, 55, 60, 65, etc.

APPEND
APPEND "name"

Appends a program from a cassette to the end of the program currently in
memory. Appended lines are renumbered in increments of 10, beginning with a
number 10 higher than the last line in the program you are adding to. All
GOTO and GOSUM references in the appended section are changed to reflect the
new line numbers. If you do not give a name in the APPEND command, the first
available program on the tape is appended. If you give a name in quotation
marks, EZEDIT searches the tape for a program with the specified name, and
appends it to your program. If no program with the given name exists on the
tape, you must press RESET and "R" to cancel the APPEND command. To use
APPEND, insert the cassette containing the program to be added. Use the
REWIND command to reposition the tape if necessary.

Then press the READ cassette button and type the APPEND command. After the
appended program loads, the screen will go blank for a few seconds while
the program lines are renumbered. Note that you can only append to a program
that occupies less than half the total programmable memory space. The
program to be appended must not use more than the remaining memory space.
Use the ?FRE(0) command to determine how many free bytes of storage you have.

APPEND "MYPROG" - searches the data tape for the program named "MYPROG"
                 and adds it to the end of the BASIC program currently
                 in memory. If the last line of the original program is
                 662, lines in the appended section will be numbered
                 672, 682, 692, etc.

CLOAD
CLOAD "name"

Loads a program from cassette tape and makes it available for editing. Any
program already in memory is overwritten. If you do not give a name, the
first available program on the tape is loaded. If you give a name in
quotation marks, EZEDIT searches the tape and loads the program that has
the name you give. If no program exists on the tape with the given name,
you must press RESET and type "R" to cancel the CLOAD command. To use CLOAD,
insert the cassette containing your program into the tape deck. Use the
REWIND command to reposition the tape if necessary. Then press the READ
cassette button and type the CLOAD command and press the "CR" key.

CSAVE
CSAVE "name"

Saves a copy of your program on tape. The tape copy is unnamed unless you
give a name in quotation marks when you enter the command. Note that your
program still remains in memory for further editing. To use CSAVE, insert
a blank data tape into the cassette deck. Use the REWIND command to
reposition the tape if necessary. Then press both the READ and WRITE buttons
simultaneously and type the CSAVE command. Note that if you rewind the tape,
CSAVE writes over anything already saved on the tape.

DELETE n-m

Deletes lines numbered n through m, inclusive, from your program. To remove a single line, simply type the line number and press the "CR" key.

DELETE 50-120   - removes lines numbered 50 through 120 from the program in memory.

60 (or DELETE 60)   - removes the line numbered 60 from the program.

EXTRACT n
EXTRACT n-m

Used with "n" only, deletes all lines except line number n from your program. With "n-m", deletes all lines except lines n-m from your program. EXTRACT is especially useful for isolating and saving parts of one program to append to another at a later time.

EXTRACT 50-170   - removes all lines in the program except those with numbers between 50 and 170, inclusive.

EXTRACT 130   - only line 130 will remain in memory.

LAST

Displays the last line in your program. LAST is useful for determining where to begin adding lines to your program.

LIST
LIST n
LIST n-m

Displays lines from your program on the TV screen. Used alone, LIST displays your whole program, beginning with the first line. LIST n lists the program, beginning with line number n. LIST n-m displays only lines numbered n through m, inclusive. See SPECIAL CHARACTERS for additional controls that can be used with LIST.

RESEQUENCE
RESEQUENCE n
RESEQUENCE n,m

Assigns new line numbers to your program lines. The "n" parameter gives the number to assign to the first program line; "m" specifies the increment between successive line numbers. The default for both "n" and "m" is 10. That is, the first line is assigned line number 10, and numbering proceeds by 10's—10, 20, 30, 40, etc. All GOTO, GOSUB, THEN, and ON references in your program lines are automatically changed to reflect the new line numbers. Note that the screen goes blank for a few seconds while the program is resequenced.

RESEQUENCE   - renumbers the lines in your program, beginning with the
                      first line, in increments of 10. Program lines will then
                      be 10, 20, 30, 40, etc.

RESEQUENCE 50   - renumbers the lines in your program in increments of 10,
                      assigning line number 50 to the first program line.
                      Program lines will be numbered 50, 60, 70, etc.

RESEQUENCE 70,5   - renumbers the lines in your program in increments of 5,
                      assigning line number 70 to the first program line.
                      Program lines will be numbered 70, 75, 80, 85, etc.

REWIND

Turns on the tape motor for repositioning of a cassette tape. Press the
REWIND cassette button to rewind the tape, FFWD to fast-forward it, READ
to move the tape forward slowly. Pressing both the READ and WRITE buttong
simultaneously erases the tape. Press any key on the keyboard to turn off
the tape motor when repositioning is complete.

?FRE(0)

Displays the number of bytes of RAM used by your program and the number of
bytes available for data or additional program lines (Same as in BASIC).

LOCATING AND CHANGING TEXT

EZEDIT provides two commands to help you locate and change text within one
or more program lines without listing or retyping them.

LOCATE "text"
LOCATE n-m "text"

Prints program lines containing the set of characters you specify in quotes.
LOCATE n-m restricts the search for "text" to program lines n through m,
inclusive. If you do not specify a range of lines, searches all lines of
your program for "text". Blank spaces within the quoted text string are
meaningful, so be sure the spacing in "text" is the same as in the string you
are trying to locate.

LOCATE "SHOW"   - prints all lines in your program that contain the text
                      string "SHOW".

LOCATE 50-120 "50" - prints all lines between lines 50 and 120 that contain
                      the number 50.

Certain words are "reserved," that is, they have special meaning to BASIC.
(For example, PRINT, GOTO, COLOR) To conserve memory space and to make

programs run faster, these words are stored and used by BASIC as numbers,
rather than as strings of characters. To you it looks like PRINT, but to
BASIC it's a number, such as 128. Therefore, to locate a string containing
a reserved word, you must refer to that number. You do this indirectly by
typing a TAB character immediately before the reserved word in "text". The
TAB character appeas on your screen as a square containing a plus sign (+).

LOCATE " +GOTO" - prints all lines in your program which contain the program
                   instruction GOTO.

A complete list of BASIC reserved words is given in RESERVED WORDS at the
end of this document.

SUBSTITUTE "newtext" FOR "oldtext"
SUBSTITUTE n "newtext" FOR "oldtext"
SUBSTITUTE n-m "newtext" FOR "oldtext"

Changes the set of characters you specify in quotes as "oldtext" to the
characters you enter in quotes as "newtext". SUBSTITUTE n changes the given
text string in program line n only. SUBSTITUTE n-m changes text in only
program lines n through m, inclusive. Blank spaces are significant in the
quoted text string, so be sure the spacing in "oldtext" is the same as in
the program line(s) you want to change. Also note that EZEDIT changes only
the first occurrence of any string in a single program line. Therefore, if
a text string is repeated within a single line, you will have to repeat the
SUBSTITUTE command to change the second occurrence.

SUBSTITUTE "ABC" FOR "DEF" - changes the first occurrence of "DEF" to "ABC"
                              in all lines of your program which contain the
                              string "DEF".

SUBSTITUTE 50 "AB" FOR "CD" - change the first occurrence of "CD" to "AB" in
                              program line 50 only.

SUBSTITUTE 50-90 "AB" FOR "CD" - changes the first occurrence of "CD" to "AB"
                                 in program lines 50-90, inclusive.

Certain words are "reserved," that is, they have special meaning to BASIC.
(For example, PRINT, GOTO, COLOR) To conserve memory space and to make
programs run faster, these words are stored and used by BASIC as numbers,
rather than as strings of characters. To you it looks like PRINT, but to
BASIC it's a number, such as 128. Therefore, to change a string containing
a reserved word, you must refer to that number. You do this indirectly by
typing a TAB character immediately before the reserved word in the quoted
text string. The TAB character appears on your screen as a square containing
a plus sign (+).

For example, suppose you misspelled one of the reserved words in one of your
program lines:

        120 PRNT "HI THERE"

You would correct the spelling of PRINT by entering:

        SUBSTITUTE 120 " PRINT" FOR "PRNT"

Note that the TAB character is required in "newtext" but is not used in "oldtext" in this case. You change it as you would change a text string that does not contain a reserved word.

Now, suppose you typed "GOTO" in a section of your program (say, lines 120-350), but you really meant "GOSUB". To make that change, you would type:

        SUBSTITUTE 120-350 "   GOSUB" FOR "   GOTO"

The TAB character is required in both the "oldtext" and "newtext" strings. If it is omitted from "oldtext", SUBSTITUTE will not find a matching string and will print an error message. If the TAB character is omitted from "newtext", the change will be recorded incorrectly, and a syntax error will result when you try to run the program. The line, when listed, will appear to be syntactically correct, however.

A complete list of BASIC reserved words is given in RESERVED WORDS at the end of this document.

SPECIAL CHARACTERS

The "CONTROL/" characters listed below are typed by holding down the CONTROL key and striking the indicated letter simultaneously.

PERIOD (.)	is used to abbreviate command words.
CONTROL/C	is used to halt a LIST command and return to EZEDIT command level.
CONTROL/S	temporarily freezes display from a LIST or LOCATE command after it finishes the line printing when the CONTROL/S is typed.
Digits 1-9	are used after a CONTROL/S to continue display, but only for a specified number of lines. For example, CONTROL/S followed by a "3" lists 3 more program lines. If you then type a "2", 2 more lines are listed, and so on.
CONTROL/Q	is used after a CONTROL/S or digits 1-9 to continue listing in a normal fashion. The listing begins with the program line which follows the last line displayed.

BASIC RESERVED WORDS

ABS	FN	NOT	SGN
AND	FOR	NULL	SIN
ASC	FRE	ON	SOUND
ATN	GOSUB	OR	SPC
CHR$	GOTO	OUTPUT	SQR
CLEAR	IF	PEEK	STEP
CLOAD	INP	PLOT	STOP
CLS	INPUT	POKE	STR$
COLOR	INSTR$	POS	TAB
CONT	INT	POT	TAN
COS	JOY	PRINT	THEN
CSAVE	LEFT$	READ	TO
DATA	LEN	REM	TONE
DEF	LET	RESTORE	USR
DIM	LIST	RETURN	VAL
DUMMY	LOG	REWIND	WAIT
END	MID$	RIGHT$	WINDOW
EXP	NEW	RND	
FIRE	NEXT	RUN	

and the arithmetic operators:

+, -, *, /, , =

MICRO *M* VIDEO™

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107
(313) 996-0626

Microsoft 8K Fast Graphics BASIC

User Guide Update

Microsoft 8K Fast Graphics BASIC is an upward compatible version of
Interact Level II BASIC that provides extended graphics capabilities.
It is recommended for development of·fast-action games, animations,
or special effects previously impossible within the BASIC language.

The program is called "8K BASIC" because is occupies 8K of your computer's
RAM. You must, of course, have at least 16K RAM on your machine to
use Fast Graphics BASIC. The remaining 8K of RAM is used for storing
your BASIC program and data and for controlling the screen display.
The available RAM for programming is the same as in Level II BASIC.

The benefits of this BASIC include:

1. Faster graphics displays

2. Simpler programming -- many FOR-NEXT loops for display
    generation can be eliminated entirely

3. Smaller programs -- programs can be written using fewer
    statements and therefore consume less RAM

4. Compatibility with Level II BASIC -- your present Level II
    programs can run "as is" in 8K Graphics BASIC

5. Compatibility with EZEDIT -- you can use your present EZEDIT
    program editor to modify programs written in Graphics BASIC
    or Level II BASIC.

6. New default color settings (black background, yellow
    statements, and white linenumbers) provide a display that
    is easier to read.

OPERATION

Load the run your Fast Graphics BASIC tape and programs in the same
manner as you do presently.

8K Graphics BASIC differs from Level II BASIC in two ways:

1. PEEKs and POKEs are automatically enabled when the program loads.
    You no longer need to enter the POKE initialization POKE 19215,25
    to use PEEK and POKE statements in your programs.

2. The PLOT statement has been extended to provide greater graphics
    capabilities, as described on the following page.

The Extended PLOT Statement

The PLOT statement now allows the use of 5 parameters, rather than the standard 3 allowed in Level II BASIC. The extended form of the PLOT command is:

PLOT x,y,c,xlen,ylen

The statement draws a rectangle in color "c" with the lower left corner at screen position "x,y", a horizontal length of "xlen" pixels, and a vertical height of "ylen" pixels. A single PLOT statement can produce a rectangle of any size, a single point, a horizontal line, or a vertical line on the screen. It produces graphics quickly by invoking a ROM subroutine, rather than the slower, point-by-point development of an image through a series of BASIC statements.

For example, try the following PLOT statement in direct mode:

PLOT 25,30,1,70,40

Isn't that fun??

Sample Program

The sample program on the following page illustrates several uses of the PLOT command to produce large color squares, grids, triangles, stripes, and random color patterns. Enter this program or a portion thereof into your computer to see some of the effects that can be achieved with your new Fast Graphics BASIC.

The PLOT statement can also be used to produce oversized lettering and fast graphics of varying shapes and sizes. Experiment with your own ideas and discover the joys of producing graphics through BASIC!

Sample Program

```
5 REM-DEMO PROGRAM FOR THE
6 REM-MICROSOFT 8K GRAPHICS BASIC
7 REM-      BY
8 REM-MICRO VIDEO CORP.
9 REM-ANN ARBOR, MI
10 CLS
20 COLOR 0,1,7,3
30 REM-TWO LARGE SQUARES
35 PLOT 7,7,3,96,66
40 PLOT 10,10,1,90,60
45 GOSUB 1000
50 REM-A GRID
60 FOR Y=10 TO 70 STEP 10
70 PLOT 10,Y,2,90,1
80 NEXT
90 FOR X=10 TO 110 STEP 10
100 PLOT X,10,2,1,60
110 NEXT
120 GOSUB 1000
130 REM-A TRIANGLE
140 L=1
150 FOR Y=70 TO 10 STEP -1
160 PLOT 10,Y,3,L,1
170 L=L+1:NEXT
180 GOSUB 1000
240 REM-SOME STRIPES
245 C=0
250 CLS
260 FOR Y=5 TO 72
262 C=C+1
270 PLOT 1,Y,C,117,1
280 NEXT
290 GOSUB 1000
300 REM-A STAR
310 CLS:COLOR 0,4,3,2
320 FOR C=1 TO 2

325 XS=59
326 XL=1
330 FOR Y=70 TO 30 STEP -1
340 XS=XS-1
350 XL=XL+2
360 PLOT XS,Y,C,XL,1
370 NEXT
380 XS=18
390 XL=83
400 FOR Y=60 TO 20 STEP -1
410 PLOT XS,Y,C,XL,1
420 XS=XS+1
430 XL=XL-2
440 NEXT
500 NEXT
510 GOSUB 1000
600 REM-RANDOM SQUARES
610 CLS
612 COLOR 0,5,6,7
615 FOR Q=1 TO 150
620 Y=5+INT(55*RND(1))
630 X=5+INT(80*RND(1))
640 C=1+INT(3*RND(1))
650 XL=2+INT(15*RND(1))
670 YL=1+INT(15*RND(1))
680 PLOT X,Y,C,XL,YL
690 NEXT
700 GOSUB 1000
999 GOTO10
1000 REM-A PAUSE LOOP
1010 FOR Q=1 TO 500
1020 NEXT
1030 RETURN
```

MICRO *MV* VIDEO™

Micro Video Disassembler

Program Instructions

The Micro Video Disassembler is a tool for the advanced programmer that allows you to process machine code at any location in RAM or ROM and display the equivalent assembly language mnemonics and operands on the TV screen. It assumes an advanced knowledge of programming and is not intended to teach machine code or assembly language programming.

LOADING THE PROGRAM: Insert the program cassette into the tape deck, depress the RESET button, depress the "L" key, rewind the tape if necessary, then depress the READ cassette button to load the program into the computer.

RUNNING THE PROGRAM: The computer asks you for the address of the first instruction to be disassembled. Enter that hexadecimal address, making certain it is the start of an "instruction boundary". Since 8080 instructions have variable lengths--1 to 3 bytes--if you start in mid-instruction, the disassembled mnemonics and operands will not reflect the actual code (and probably won't make much sense!)

For starting addresses to decode ROM subroutines, consult Micro Video's Guide to ROM Subroutines.

To disassemble a program, load the free-standing machine code into the computer before you load the Disassembler, using the RESET-"L" sequence. For accurate disassembly, you will need to know the starting address of your program.

Given the starting address of an instruction, the computer displays a full screen (page) of instructions and then pauses. Each line contains one instruction only. You can enter any of the following characters to control the display:

    P - To display another full "page" of instructions

    L - To display instructions one line at a time

    C - To display instructions continuously

    S - To stop a full page (P) or continuous (C) display

    E - To exit from display mode and give another address

NOTES:

1. Illegal op-codes are displayed as asterisks (*). If illegal op-codes appear, you may be attempting to disassemble stored data rather than instructions, or the starting address you gave may not be on an "instruction boundary".

2. The Disassembler loads into two blocks of RAM: 4C00-4C10 and 6000-6930 hexadecimal. It can disassemble itself.

3. All constants and addresses are displayed in yellow. The op-codes and register references appear in white.

4. The Disassembler can be used to "double-check" machine code written with the Micro Video MONITOR program. It is thus useful as a debugging tool for your own programs. Follow the instructions given in "RUNNING THE PROGRAM" to use the Disassembler for this purpose.

5. A list of suggested 8080 programming references is contained in the Guide to ROM Subroutines.

MICRO
VIDEO
MONITOR

MICRO VIDEO ™

The

MICRO VIDEO Monitor

User Guide

INTRODUCTION

The MICRO VIDEO Monitor is a machine-language program which is designed for use on the Interact Model One computer. Using commands entered at the keyboard, you can:

- Display the contents of RAM and ROM locations

- Substitute new values into RAM locations

- Execute programs with or without breakpoint control

- Examine and change selected registers

- Save any block of memory on cassette tape for later use

- Move the contents of memory forward or backward in RAM

- Fill a block of memory with a constant value

- Use partial screen display to facilitate graphic programming

- Compute the Sum and difference of two hexadecimal numbers to simplify absolute machine address programming

- Develop machine-language subprograms which can be combined with Level II BASIC programs and the USR function

The Monitor can be used as a tool to:

- Learn 8080 programming

- Develop faster game subprogram displays for use in a BASIC game environment

- Gain access to over 10K of available memory

- Convert published 8080 programs into an Interact environment

MONITOR DESCRIPTION

The Monitor is a machine-language program that is available on cassette tape and is loaded into the computer in the standard manner. Insert the program cassette tape into the tape deck, depress the RESET button, depress the "L" key, depress the REWIND button to rewind the tape and then depress the READ button to load the program into the computer.

The Monitor requires approximately 2K bytes of RAM memory and is loaded into the following two blocks:

	Locations in (Hexadecimal)		
Block 1	4C00	–	4C69
Block 2	6000	–	66FF

The Monitor thus resides in the same locations as does Level II BASIC. Therefore, the BASIC tape must be loaded <u>after</u> the Monitor tape in those programs which combine both BASIC and machine language programming.

MONITOR COMMANDS

Monitor commands consist of a single alphabetic character which specifies the command, followed by a list of numeric parameters. Numeric parameters are specified as hexadecimal numbers and are separated from each other by <u>commas</u>. Leading zeroes in hexadecimal numbers may be omitted, and <u>spaces</u> are ignored. The command is terminated by pressing the Carriage Return (CR) key.

The command is not executed until the (CR) key is pressed. Thus you may use the Backspace key to erase any typing errors and retype the remainder of the command correctly. If the Monitor discovers a syntax error in a command, a single question mark (?) appears on the screen following the command you entered. You are then prompted for the next command.

The command prompt is a greater-than-sign ($>$). It indicates that the Monitor is ready to accept your next command.

Notes:

1. When using the Monitor, all numeric values are expressed as hexadecimal (base 16) numbers, whereas BASIC uses all decimal (base 10) numbers. It is recommended that you have conversion tables, a calculator or be familiar with number base conversion techniques.

2. No positions of RAM, including the locations which contain the Monitor, are protected. Therefore, you are urged to use caution and to write copies of your program to cassette tape for backup purposes during program development.

3. When referencing the screen using ROM subroutines, the origin is at the upper left corner.

$$0,0 \quad\underline{\hspace{5cm}}\quad 112,0_{10}$$
$$70,0_{H}$$

$$0,4D_{H}$$
$$0,77_{10}$$
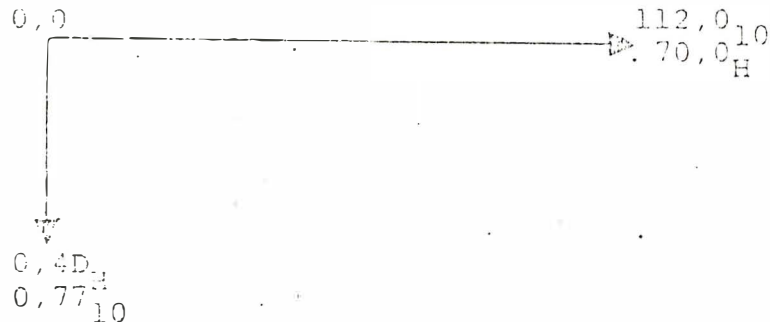
4. When loading tapes that combine BASIC user programs and machine language instructions, use the RESET, "L" sequence rather than the CLOAD command. The CLOAD command will not load machine-language instructions that may be on the cassette tape along with the BASIC program.

INDIVIDUAL MONITOR COMMANDS

DISPLAY Memory -- D ump

>D <startaddress>

where <startaddress> is the first (lowest) location in memory
to be displayed. Memory is displayed four locations per line
in a scrolling display. You may cause the display to pause by
depressing any key other than the (CR) key. Resume the display
by depressing another key. To stop the display and return to
Command Level, press the (CR) key.

Example:

To display the contents of ROM memory enter the command:

>D0

which produces the following display: *Contents of the addresses*

```
       ┌ 0000    3E 40 32 02 ┐      0000 = 3E
Hex   │  0004    28 C3 0C 00 │      0001 = 40
addresses 0008  F3 C3 04 08  │      0002 = 32
       │  000C    3E 38 32 00 │      0003 = 02
       └ 0010    30 AF 32 00 ┘      0004 = 28
                                      ↓ ect ↓
```

The first column of numbers are the hexadecimal addresses of
successive locations, in increments of four. The four columns
following the hex addresses display the contents of the four
successive locations beginning at each address.

```
      0 = 0  ┐
      1 = 1  │
      2 = 2  │
      3 = 3  │
      4 = 4  │
      5 = 5  │
      6 = 6  ├ 16 characters
      7 = 7  │
      8 = 8  │
      9 = 9  │
      A = 10 │
      B = 11 │
      C = 12 │
      D = 13 │
      E = 14 │
      F = 15 ┘
```

-4-

FILL Memory - F

>F <lowaddress>, <highaddress>, <constant>

where <lowaddress> is the first address of the block that is
to be filled, <highaddress> is the last address of the block,
and <constant> is the hexadecimal value to be stored in all
locations specified.

Example:

The TV screen is memory-mapped to locations 4000-49FF. There-
fore, to fill the screen with different patterns, try the follow-
ing FILL commands:

>F4000,49FF,0

>F4000,49FF,55

>F4000,4300,23

>F4300,4400,47

MOVE Memory - M

>M <lowaddress>, <highaddress>, <destination>

where <lowaddress> and <highaddress> define the beginning
and ending addresses of a block of memory whose contents are
to be moved to successive locations beginning at the <destina-
tion> address. The block may be moved forward or backward in
memory.

Example:

To display ROM memory in bit patterns on the screen, simply
MOVE the ROM memory to the screen with the command:

>M0,7FF,4000

PARTIAL Screen Display

>P <lines>

where <lines> is the hexadecimal number of text lines which
are allowed to scroll on the bottom portion of the screen. No
scrolling of information above the lines specified in the PAR-
TIAL command occurs with subsequent Monitor commands. By using
the PARTIAL command, you can more easily develop graphic displays
on the screen.

<lines> must have a value between 1 and B, hexadecimal, for
partial scrolling. If <lines> is greater than B, then a full
screen scroll takes place.

Example: To specify 4-line scrolling, enter the command:

>P4   *Same as Window in Basic*
         1 is full window down to one line visible
         B i. a full screen (no window)

EXAMINE/MODIFY Registers - X

>X <register>

where <register> specifies an 8080 register as follows:

A	Accumulator	*Exp. A = F0 F = F2*
B,C,D,E	Registers	*B = C9 C = 7F*
F	Flag Register	*D = 48 E = 41*
M	HL Register pair	*H = 5A L = 46*
P	Program counter	*M = 5A46*
S	Stack pointer	*P = D47A*
		S = 7F00

If no <register> is specified, the value of all registers are
displayed on the screen. If <register> is specified, the con-
tents of the registers, beginning with the specified register,
are displayed and may be selectively changed by responding to
queries.

- To change the value of a register, simply type in the new
  value. *Exp. A = F0 — C9 → changed to C9 → A = C9*

- To leave the value of the register unchanged, press the
  (CR) key.

- To exit from the query process, simply type the Backspace
  key.

The contents of the registers are not actually modified until a PROGRAM EXECUTE - G command is entered. The values are stored and displayed from a table located in RAM.

Example:

>XA

begins the display of registers starting with the accumulator and allows selective modification of each register individually.

HEXADECIMAL Sums and Differences - H

>H <number1>, <number 2>

where <number1> and <number 2> are two hexadecimal numbers whose sum and difference are to be computed and displayed on the screen. This command is useful to determine block lengths when the low and high address of the block are known.

Example:

>H4000,49FF

+39FF   -09FF
      ↖   ↖
     Sum   Difference

WRITE Tape - W

>W <lowaddress>, <highaddress>

where <lowaddress> and <highaddress> specify the block of memory that is to be written to cassette tape.

Examples:

>W 4FFD,5F7E   writes your BASIC and machine language program
                to tape

>W 4000,49FF   writes the screen to tape

LOAD Tape - L

>L

The LOAD command loads a tape into RAM according to the infor-
mation written on the cassette tape. After the tape has loaded,
control returns to the Monitor, providing it has not been over-
written as a result of the LOAD command.

REWIND Tape - R

>R

The REWIND command turns on the tape motor, allowing you to po-
sition the tape as desired by pressing the appropriate cassette
buttons. Depress any key on the keyboard to terminate the REWIND
command.

SUBSTITUTE Memory - S

>S <startaddress>

where <startaddress> is the first address whose contents are to
be displayed and optionally changed on an individual basis. The
SUBSTITUTE command displays memory, one location at a time, in a
forward or backward direction and allows you to enter a new value
or to leave the current value unchanged as follows:

   o To leave the value unchanged and display the next higher
     location, type a (CR). *Higher Location (CR)*

   o To leave the current value unchanged and display the previous
     location, type a Backspace key. *Lower Location (Back Space)*

   * To change a value and then display the next higher loca-
     tion, type in the new hexadecimal value and type a (CR).

   o To return to Command Level, type a period (.) followed by
     a (CR).

Example:

To illustrate the use of a PARTIAL screen and SUBSTITUTE command
to fill the screen RAM with bit patterns try a sequence like the
following:

>P4
>S4300
  FF
  11
  2
etc.

-8-

PROGRAM EXECUTE (GO) - G

> G <startaddress>, <breakpoint>

where <startaddress> specifies the address to which control is
to be given, and <breakpoint> optionally specifies the location
at which control of execution is to be returned to the Monitor.
Note that the Monitor places a three-byte call instruction at the
breakpoint location to recover control.

A breakpoint can also be generated with a Return instruction (C9)
in the user code, providing that everything is off the stack when
the Return instruction is encountered. All registers, with the
exception of the program counter, will be valid on return and can
be examined with the X command.

Examples:

> G5B00   Begin execution at location 5B00

> G5B00,5B32   Begin execution at 5B00 and return control
                   to the Monitor when the first attempt is
                   made to execute the instruction at 5B32

> G4C00   Begin execution of the Monitor

MEMORY MAP

Hexadecimal Addresses   Decimal Addresses

```
0   ┌────────────────────────────────┐  0
    │                 .              │
    │              ROM               │
7FF ├────────────────────────────────┤  2047
    │   Keyboard and Memory-         │
    │   Mapped I/O                   │
3FFF├────────────────────────────────┤  16383  ─ ─ ─ ─ ─ ─ ─ ─ ─ ─
4000│                                │  16384                        ▲
    │                                │
    │   TV Screen Display            │
    │                                │
49FF├────────────────────────────────┤  18943
4A00│                                │  18944
    │                                │
    │   BASIC Variables Space        │
    │                                │
4D21├────────────────────────────────┤  18745
4D22│                                │  19746
    │   BASIC Text Area (User        │
    │   program)                     │
    │                                │
5F7F├────────────────────────────────┤  24447
    │   Stack and ROM Variables      │
6000├────────────────────────────────┤  24576             16K RAM
    │                                │
    │                                │
    │                                │
    │                                │
    │   BASIC Interpreter            │
    │                                │
    │                                │
7FFF└────────────────────────────────┘  32767  ─ ─ ─ ─ ─ ─ ─ ─ ─ ─▼
```

8080 FREE STANDING PROGRAM

This program illustrates a 3-instruction program that invokes the RFILL subroutine that resides in ROM. RFILL will fill a rectangle on the screen very rapidly according to position, size, and color information that is contained in a table.

The program looks like this:

Location	Instruction	Hexadecimal
5E00	LXI B,5E10	01 10 5E
5E03	CALL RFILL	CD A2 05
5E06	RET	C9

and the table looks like this:

Location		Initial Contents	Interpretation
Decimal	Hexadecimal		
24080	5E10	4D	Vertical Size
24081	5E11	70	Horizontal Size
24082	5E12	02	Color
24083	5E13	0	X Coordinate
24084	5E14	0	Y Coordinate

where the X,Y coordinates specify the upper left corner of the rectangle. Remember that the origin of the screen is at the upper left corner as well.

Procedure

1. Enter the contents of these locations using the SUBSTITUTE command and initiate program execution by typing G5E00.

2. Change the color value in the table to color 3 and then give another G5E00 command.

3. Make squares of different sizes and screen locations by varying the vertical and horizontal sizes and X,Y coordinates. Be careful that you do not attempt to draw too large a square as doing so may erase program memory.

LEVEL II BASIC WITH MACHINE CODE SUBPROGRAM

1. Enter the 8080 Free Standing Program previously described in its original form.

2. Load the Level II BASIC Tape. After typing 'NEW', type in the following program. The USR(0) calls invoke the machine language code you entered previously. Type RUN.

3. Experiment with different color and size squares.

```
1 POKE 19215,25
2 POKE 19473,0
3 POKE 19474,94
10 CLS
100 VS=69
110 HS=104
120 VC=4
130 HC=4
140 FOR I=1 TO 8
150 POKE 24080,VS
160 POKE 24081,HS
170 POKE 24082,I
180 POKE 24083,VC
190 POKE 24084,HC
200 U=USR(0)
210 VS=VS-8
220 HS=HS-8
230 VC=VC+4
240 HC=HC+4
250 NEXT
260 FOR I=0 TO 3
270 POKE 24082,I
280 U=USR(0)
290 NEXT
300 GOTO 260
```

REFERENCES

8080/8085 Assembly Language Programming Manual. Santa Clara: Intel Corporation, 1977 (Order #9800301B)

Spracklen, Kathe, Z80 & 8080 Assembly Language Programming. Rochelle Park, New Jersey: Hayden, 1979. $8.95.

Graphics for programs are generated with the MONITOR in two ways. A block can be drawn at any location on the screen in any size and color by calling the ROM subroutine RFILL. This is useful for full or partial screen wipes or to draw a background rectangle very quickly.

The second method of generating graphics uses another ROM subroutine, RPLOT. RPLOT takes a picture or pattern you have stored in memory and outputs it to the screen.

RFILL

The ROM subroutine RFILL is located at 05A2H. When called, it produces a rectangle on the screen in the color and size you specify, at the location you specify.

To use RFILL, you must first set up a control table somewhere in memory that contains the parameters of the block to be output (size, color, coordinates). For example, to put an 8x8 pixel square in the upper lefthand corner of the screen, define the following table using the MONITOR SUBSTITUTE command:

Location	Value	Defines:
5810	08	vertical size in pixels
5811	08	horizontal size in pixels
5812	01	color (0-3)
5813	0A	vertical screen coordinate
5814	0A	horizontal screen coordinate

The starting memory location of this table is arbitrary. It may be placed anywhere within the memory you have defined for machine code storage with the BASIC POKE statements, but the table locations must be contiguous.

Then, to draw the rectangle on the screen, call RFILL and the table as shown below:

5800	01	10	58	CD
5804	A2	05	C3	00
5808	58	00	00	00
580C	00	00	00	00
5810	08	08	01	0A
5814	0A	00	00	00

calls RFILL to use table at 5810

RFILL control table

To execute this routine, type the MONITOR command G5800.

The ROM subroutine RPLOT is located at 497H. RPLOT allows you to take pictures stored in memory and draw them on the screen very quickly. To use RPLOT, you must define a table with the output parameters (as with RFILL) and specify the bit pattern of the pictures to be drawn.

Defining the RPLOT Control Table

Use the MONITOR SUBSTITUTE command to define the control table. As with the RFILL table, the location of the starting byte is arbitrary, but the table bytes must be contiguous.

To put an 8x8 pixel pattern in the lower righthand corner of the screen, define the table as follows:

Location	Value	Defines:
5810	20	LSB of bit pattern address
5811	58	MSB of bit pattern address
5812	08	vertical size in pixels
5813	08	horizontal size in pixels
5814	03	color (0-3)
5815	38	vertical screen coordinate
5816	58	horizontal screen coordinate

Defining the RPLOT Bit Pattern

The picture you want to appear on the screen must be stored in memory as a bit pattern before it can be accessed by RPLOT. The simplest method of defining the bit pattern is to draw the picture on a piece of quadrille graph paper. The following example defines the bit pattern to output a small diamond on the screen according to the parameters defined in the control table above.
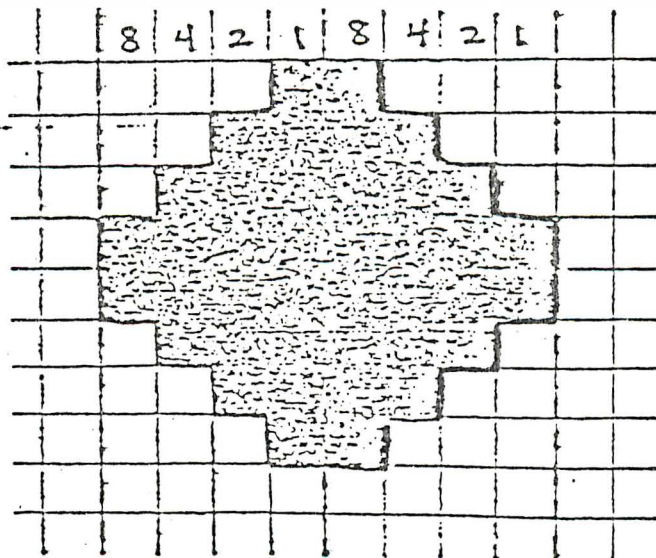
Starting at the leftmost edge of the picture, divide the picture vertically into half byte increments (4 pixels). Each group of eight pixels corresponds to a single byte. The first pixel in each half byte has a value of 8 if turned on. The second has a value of 4, the third a value of 2, and the fourth a value of 1. Add up the bits to be turned on for each half byte and convert to hexadecimal. Group the half bytes together to form full bytes and place that value in the appropriate memory location. Memory locations must be consecutive. For the diamond shape on the previous page, the bit pattern would be defined as follows:

```
5820   18
5821   3C
5822   7E
5823   FF
5824   FF
5825   7E
5826   3C
5827   18
```

Note that, in this example, the picture to be drawn is only one byte wide. You can, of course draw larger graphics in the same way. If your picture ends in "mid-byte", add zeros where no bits are turned on to form full bytes, as illustrated below.
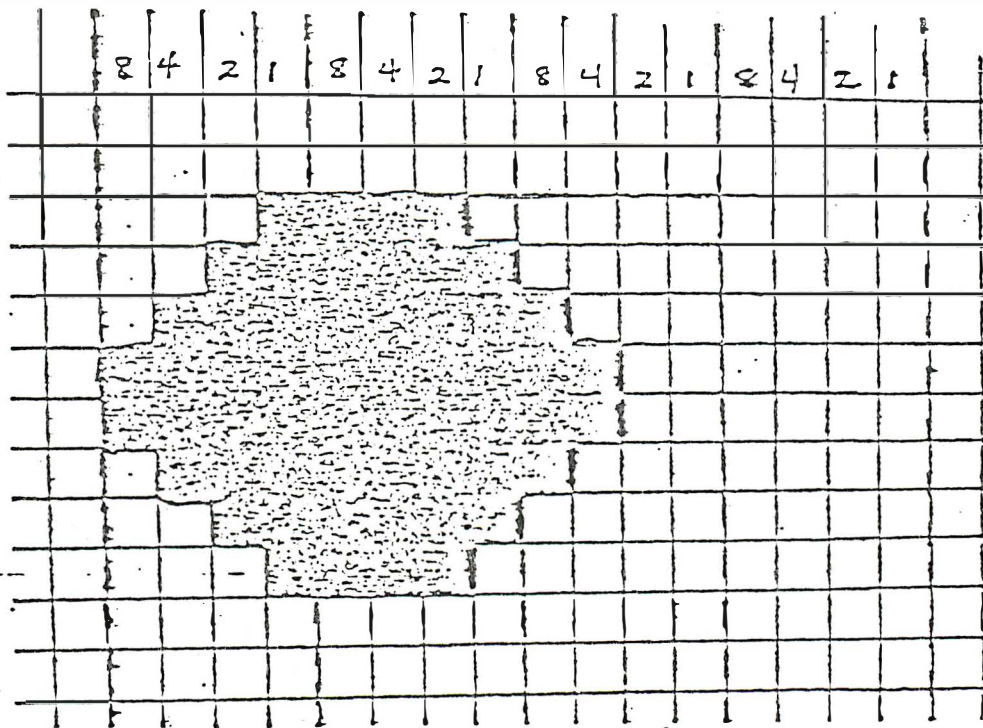
Because the above picture is 10 pixels wide, two full bytes are required to place it in memory. Zeros are used for those bits that are not turned on to form full bytes. The bit pattern for this picture would be:

```
5820    1E   00
5822    3F   00 ·
5824    7F   80
5826    FF · CO
5828    FF   CO
582A    7F   80.
582C    3F   00
582E    1E   00
```

To draw this picture on the screen with RPLOT, the value in location 5813 of
the control table (horizontal size in pixels) must be changed from 08 to 0A.
The rest of the table would remain the same.

Executing RPLOT to Draw the Picture on the Screen

To draw the diamond pattern in the first example on the screen, use the
following coding sequence:

```
5800    01   10   58   CD  ──────── call RPLOT and use table at 5810
5804    97   04   C3   00
5808    58   00   00   00
580C    00   00   00   00
5810    20   58   08   08  ──────── RPLOT control table
5814    03   38   58   00
581C    00   00   00.  00
5820    18   3C   7E   FF  ──────── bit pattern of picture to be
5824    FF   7E   3C   18            drawn with RPLOT
```

To execute this sequence you would type the MONITOR command `G5800`.

Graphics Motion

The following text and example describe the fastest way to move graphics
on the screen.

Located at 5CC4H is a routine used to move pictures stored in memory to
the screen. This routine is used in the BOMBS AWAY! program to draw the
plane, final explosion, and door and missile on top of the dam on the screen.
Before execution of this routine, a control table must be entered in memory.
For example, the table for drawing the airplane begins at 5C74H. This table
specifies:

```
5C74 ⎫ Address of first byte in top line of picture
5C75 ⎭
5C76   Vertical size in pixels
5C77   Horizontal size in pixels
5C78   Empty
5C79   Vertical screen location
5C7A   Horizontal screen location
5C7D   Must be 1
```

```
.10   3E  55  32  0E        57F0  01  01  01  CD          02  CO  32  CF
      43  01  01  04              F6  07  C9              58  3A  7A  5C
      CD  F6  07  3E        5800  01  08  58  CD          32  CE  58  C9
      00  32  0E  43              97  04  C9        58E0  3A  79  5C  3D
      3A  13  4A  D6              70  58  07  0A          32  79  5C  FE
      1F  32  13  4A              01  26  1C              00  CO  3E  01
      32  1E  4A  FE        5810  01  18  58  CD          32  79  5C  3A
      0E  C2  10  4A              97  04  C9              74  5C  C6  40
      C9                          24  58  1D  08          32  74  5C  FE
4A40  3E  E7  32  13              03  2F  73              01  CO  3E  0A
      4A  32  1E  4A        5820  C3  60  5E              32  74  5C  3E
      3E  01  32  87              CO  CO  CO  CO          00  32  CF  58
      4A  3E  22  32              EO  EO  EO  EO          C9
      8D  4A  C3  10              FO  FO  FO  FO    5910  21  14  40  22
      4A                          F8  F8  F8  F8          74  5C  21  23
4A60  CD  80  4A  3A              FC  FC  FC  FC          30  22  76  5C
      87  4A  3C  32              FE  FE  FE  FE          CD  CO  5C  C9
      87  4A  3A  8D              FF  FF  FF  FF    5920  2A  EF  5F  22
      4A  3D  32  8D              FF                      CA  58  7C  AD
      4A  C9                5848  01  02  FF              EB  CD  71  59
4A80  21  AB  5B  22        584C  C3  D0  59  00          6F  22  CC  58
      74  5C  21  01        5850  C3  80  57  00          CD  40  59  FE
      06  22  76  5C        5854  C3  CO  5D  00          11  D2  30  59
      21  22  1A  22              74  5E  14  18          32  79  5C  C9
      79  5C  CD  C4              01  38  3D        5940  E5  D5  2A  CA
      5C  C9                      1F  FF  7F  FF          58  EB  2A  CC
5780  21  18  56  22              FF  FF  FF  FF          58  7C  17  7B
      74  5C  21  07              FF  FF  7F  FF          17  AD  6F  7B
      09  22  76  5C              1F  FF  03  FF          17  7A  17  AC
      21  13  30  22              18  00  76  00          67  7A  17  17
      79  5C  CD  CO              AC  80  F3  04          E6  01  AB  5F
      5C  21  D4  55              AC  80  76  00          7D  17  AB  5F
      22  74  5C  21              18  00  00  00          7D  17  7C  17
      0B  0E  22  76              90  59  0D  05          AA  57  43  22
      5C  21  11  2E              03  3F  36  00          CC  58  EB  22
      22  79  5C  CD              1E  02  01  2F          CA  58  D1  E1
      CO  5C  21  54              1D  00  00  00          C9  21  00  00
      53  22  74  5C        5890  01  98  58  CD          D5  E5  06  08
      21  13  18  22              A2  05  C9  00          29  17  D2  81
      76  5C  21  0D              01  0C  00  4D          59  19  CE  00
      29  22  79  5C              13  00  00  00          05  C2  78  59
      CD  CO  5C  21        58A0  CD  FO  59  CD          C1  D1  C9
      AB  53  22  74              A0  59  C9  00    5990  20  F8  20  20
      5C  21  1B  21              C3  40  4A  CD          F8  20  20  20
      22  76  5C  21              D0  58  3A  CF          20  20  20  20
      09  25  22  79              58  FE  02  CC          20
      5C  CD  CO  5C              EO  58  CD  C4    59A0  21  01  5A  22
      21  20  53  22              5C  3A  7A  5C          74  5C  21  0B
      74  5C  21  23              3D  3D  32  7A          14  22  76  5C
      2B  22  76  5C              5C  FE  71  CO          3E  ED  32  7A
      21  05  20  22              C3  A0  58  00          5C  CD  20  59
      79  5C  CD  CO              00  00  00  00          3A  79  5C  C6
      5C  CD  FO  57              00  00  00  00          03  32  79  5C
      CD  10  59  C9        58D0  3A  F7  5F  FE          97  32  CF  58
```

```
         C9                    3A  6F  5C  32         5E40  CD  C4  5C  01
59D0  0E  01  11  42           70  5C  E5  2A               01  30  CD  F6
      22  CD  CA  7F           72  5C  EB  19               07  CD  60  4A
      01  00  08  CD           E3  19  D1  01               C9  00  00  00
      BB  7F  0E  01           00  00  C3  25         5E50  CD  C4  5C  01
      11  42  26  CD           5D  7B  0F  0F               01  30  CD  F6
      CA  7F  C9               0F  5F  E6  1F               07  C9  00  00
59F0  3E  10  32  74           47  3E  E0  A3         5E60  01  90  5E  CD
      5C  CD  C4  5C           4F  3E  FC  A2         5E64  A2  05  01  B8
      3E  01  32  74           0F  0F  5F  16         5E68  5E  CD  97  04
      5C  C9                   00  21  00  40         5E6C  01  98  5E  CD
5C74  01  5A  0B  14           19  09  C9  D5         5E70  A2  05  01  A0
      00  13  C9  00           C5  5E  23  56         5E74  5E  CD  A2  05
      00  01  00  00           23  4E  23  46         5E78  01  A8  5E  CD
5CC0  CD  F0  57  00           23  D5  5E  23         5E7C  A2  05  01  B0
      21  7B  5C  11           56  EB  D1  C5         5E80  5E  CD  A2  05
      74  5C  CD  93           D5  E5  1A  77         5E84  01  58  58  CD    5E8A
      5D  21  EF  5F           23  13  05  C2         5E88  97  04  (01) 80   POKE
      7E  BE  CA  D1           A6  5D  E1  01         5E8C  58  C3  97  04
      5C  EB  4E  23           20  00  09  E8         5E90  23  14  03  29
      46  C5  23  7E           E1  09  EB  C1         5E94  00  00  00  00
      32  71  5C  23           0D  C2  A3  5D         5E98  27  0A  02  25
      46  23  23  5E           C1  D1  C9                   14  00  00  00
      23  56  7A  E6     5DC0  21  0E  5B  22               1E  02  01  2F
      03  21  4F  5D           74  5C  21  0A               1D  00  00  00
      FE  00  CA  04           05  22  76  5C               02  0C  01  23
      5D  21  44  5D           21  19  14  22               01  0A  01  22
      FE  01  CA  04           79  5C  CD  50               14  00  00  00
      5D  21  36  5D           5E  21  10  5B               C4  5E  02  10
      FE  02  CA  04           22  74  5C  21               03  27  03  00
      5D  21  2A  5D           0A  06  22  76               61  86  F3  CF
      22  28  5D  C6           5C  21  19  12         5EC4  00  06  C0  00
      03  80  E6  FC           22  79  5C  CD               06  C0  1F  06
      0F  0F  32  70           50  5E  21  13               C0  1F  06  C0
      5C  32  6F  5C           5B  22  74  5C               1F  06  C0  1F
      D6  20  2F  3C           21  0A  04  22               06  C0  1F  06
      6F  26  00  22           76  5C  21  19               C0  1F  06  C0
      72  5C  CD  79           12  22  79  5C               1F  06  C0  1F
      5D  D1  01  00           CD  40  5E  21               06  C0  1F  1F
      00  1A  13  C3           15  5B  22  74               F0  FF  1F  F0
      2A  5D  0F  0F           5C  21  0A  04               FF  1F  F0  FF
      F5  E6  3F  4F           22  76  5C  21               1F  F0  FF  1F
      F1  E6  C0  C3           19  12  22  79               F0  FF  FF  FF
      4D  5D  07  07           5C  CD  40  5E               FE  FF  FF  FF
      07  07  F5  E6           21  17  5B  22               FF  FF  FF  FF
      0F  4F  F1  E6           74  5C  21  0A               FF  FF  FF  FF
      F0  C3  4D  5D           05  22  76  5C
      07  07  F5  E6           21  19  11  22
      03  4F  F1  E6           79  5C  CD  40
      FC  B0  41  77           5E  21  19  5B
      23  3A  70  5C           22  74  5C  21
      3D  32  70  5C           0A  05  22  76
      C2  25  5D  71           5C  21  19  10
      3A  71  5C  3D           22  79  5C  CD
      C8  32  71  5C           40  5E  C9  00
```

VECTOR GRAPHICS SUBROUTINES

User Guide

Vector Graphics is a set of three machine language subroutines designed specifically for use with the Interact. The subroutines are particularly useful for fast graphics displays—they allow you to draw very fast lines given the coordinates of the end-points of the line, or triangles in any orientation of any size, given the vertices. These routines can be called with the USR command from Microsoft 8K Fast Graphics BASIC (or the older, Level II BASIC) or from 8080 machine code entered with the Micro Video MONITOR. Although the average user will not need to examine the subroutines themselves, the more advanced programmer may be interested in disassembling this code using the Micro Video Disassembler.

LOADING THE SUBROUTINES

Your Vector Graphics tape contains two different files:

1. Graphics Subroutines machine code (1 copy)

2. BASIC Demonstration Program (2 copies)

To use these subroutines, follow this sequence of operations:

1. Load BASIC by depressing RESET, pressing "L", rewinding the tape if necessary, then depressing the READ cassette button to load the BASIC interpreter.

2. Load the Graphics Subroutines (first file on tape) using the RESET/"L"/Rewind/Read sequence.

3. Load or enter your BASIC program in the usual manner. Before keying in a new program, remember to type NEW. NEW does not affect the Graphics Subroutines. If you are loading the BASIC demonstration program included with the Graphics Subroutines, use the RESET/"L" sequence, but do not rewind your tape.

USING THE SUBROUTINES

The Graphics Subroutines reside in locations 5E00-5EF3 (hex), and a data block occupies locations 5EF4-5EFE. This RAM is at the end of the BASIC programming area, so make sure that your BASIC program does not become so long that it overwrites and destroys the subroutine code.

To call the subroutines from BASIC, you must set the address of the desired
subroutine in two locations by entering two POKE statements prior to the
A=USR(O) statement that calls the subroutine from the BASIC program. This
address is poked into locations 19474 and 19473.

The calling parameters for the subroutines are also passed to machine code
by POKE statements in BASIC. You can also set the parameters directly
at machine code level with the Micro Video MONITOR. The calling parameters
for the three Graphics Subroutines are given in both hexadecimal and decimal
values.

DLINE

The DLINE subroutine starts at hex address 5E6B. DLINE draws the best
approximation of a straight line between two points, given the screen coordinates
of those points. To access this subroutine, POKE the values 94 and 107
(decimal) into locations 19474 and 19473, respectively (as shown in line
30 of the following example). The calling parameters for DLINE are:

Parameter	Hex Location	Decimal Location	
COLOR	5EF4	24308	Selects color from current color set (0-3) to draw line.
X1	5EF5	24309	(X1,Y1) coordinate pair for line starting point
Y1	5EF6	24310	
X2	5EF7	24311	(X2,Y2) coordinate pair for second line endpoint
Y2	5EF8	24312	

A sample call to draw a line that slopes downward toward the right (from
(1,10) to (100,60)) in color 2 in the currently defined color set is as
follows:

```
10 CLS:COLOR 0,1,3,7                          Not required in 8K BASIC
20 POKE 19215,25                              Sets subroutine address
30 POKE 19474,94:POKE 19473,107
40 POKE 24308,2                               Sets color 2 for use (yellow)
50 POKE 24309,1:POKE 24310,10                 First endpoint (1,10) coordinates
60 POKE 24311,100:POKE 24312,60               Second endpoint (100,60) coordinates
70 A=USR(0)                                   Call DLINE to draw line
80 GOTO 10                                    Repeat above steps
```

TRIF

The TRIF subroutine starts at hex address 5E00. TRIF draws a completely
filled triangle on the screen, given the coordinates of the three *vertices*.
The TRIF subroutine is about 3 times slower than the TRIA subroutine,
although it ensures that the entire triangle is color-filled, regardless
of its orientation. To set the address of TRIF, POKE the decimal values
94 and 0 into locations 19474 and 19473, respectively.

The TRIF calling parameters are:

Parameter	Hex Location	Decimal Location	
COLOR	5EF4	24308	Selects color from current color set (0-3) to draw line
X1	5EF9	24313	First vertex coordinates
Y1	5EFA	24314	
X2	5EFB	24315	Second vertex coordinates
Y2	5EFC	24316	
X3	5EFD	24317	Third vertex coordinates
Y3	5EFE	24318	

A sample program to draw a triangle three times in three different colors (red, white, blue) on a black background is as follows:

```
10 CLS:COLOR 0,1,7,4          Not required in 8K BASIC
20 POKE 19215,25              Sets subroutine address
30 POKE 19474,94:POKE 19473,0
40 POKE 24313,10:POKE24314,15 Vertex at (10,15)
50 POKE24315,100:POKE24316,30 Vertex at (100,30)
60 POKE24317,80:POKE24318,60  Vertex at (80,60)
70 FOR C=1 TO 3               Color loop
80 POKE 24308,C               Set color
90 A=USR(0)                   Call subroutine to draw triangle
100 NEXT
110 CLS:GOTO 70               Clear screen and repeat
```

TRIA

The TRIA subroutine starts at hex address 5E28. TRIA produces a completely filled triangle at three times the speed of TRIF, providing the side opposite the first vertex given is either horizontal or vertical. If the side is not vertical or horizontal, then some of the internal points may not be plotted in the specified color. Set the address for the TRIA subroutine by poking the values 94 and 40 into decimal locations 19474 and 19473, respectively with POKE statements in BASIC.

The calling parameters for TRIA are identical to those for TRIF, given above.

The sample program on the following page illustrates the use of the TRIA subroutine to produce a set of "telescoping triangles".

```
10 REM TELESCOPING TRIANGLES
20 CLS:COLOR 0,7,1,4
30 POKE 19215,25
40 POKE 19474,94:POKE 19473,40
50 X1=55:Y1=0
60 X2=10:Y2=70
70 X3=110:Y3=70
80 FOR T=1 TO 8
90 POKE 24308,T
100 Y1=Y1+5
110 X2=X2+3:Y2=Y2-3
120 X3=X3-3:Y3=Y3-3
130 POKE24313,X1:POKE24314,Y1
140 POKE24315,X2:POKE24316,Y2
150 POKE24317,X3:POKE24318,Y3
160 A=USR(0)
170 NEXT
180 CLS:GOTO 50
```

NOTES

1. We recommend that you load and RUN the BASIC Demonstration Program
    on the tape, as you'll see the wide variety of effects that can be
    produced using the Vector Graphics Subroutines. A listing of that
    BASIC program is included with this guide.

2. When you CSAVE your programs that call the subroutines, the subroutines
    will not be copied to tape. You will need to reload them each time
    you wish to load and run your programs that use them.

3. The initializing POKE 19215,25 is required when using Level II BASIC
    to access the subroutines, but can be omitted if you are·using the
    new Microsoft 8K Fast Graphics BASIC.

4. If your program calls the subroutines more than once, you will find
    it easier to assign the POKE addresses to variables once at the beginning
    of the program and then use the variable names in the various POKE
    statements. This will reduce the likelihood of typing errors causing
    you to POKE the incorrect locations in memory, which can be fatal to
    your program! For example,

        X1=24309:Y1=24310...

        POKE X1,100:POKE Y1,35

    5. The screen origin for the Vector Graphics subroutines is at the
    upper left corner of the TV screen, rather than at the bottom left
    corner as it is in BASIC. The screen and sample relative positions
    of various (x,y) coordinate pairs are as follows:

X

Y

. (0,0)   (112,0) .

. (0,77)   (112,77) .
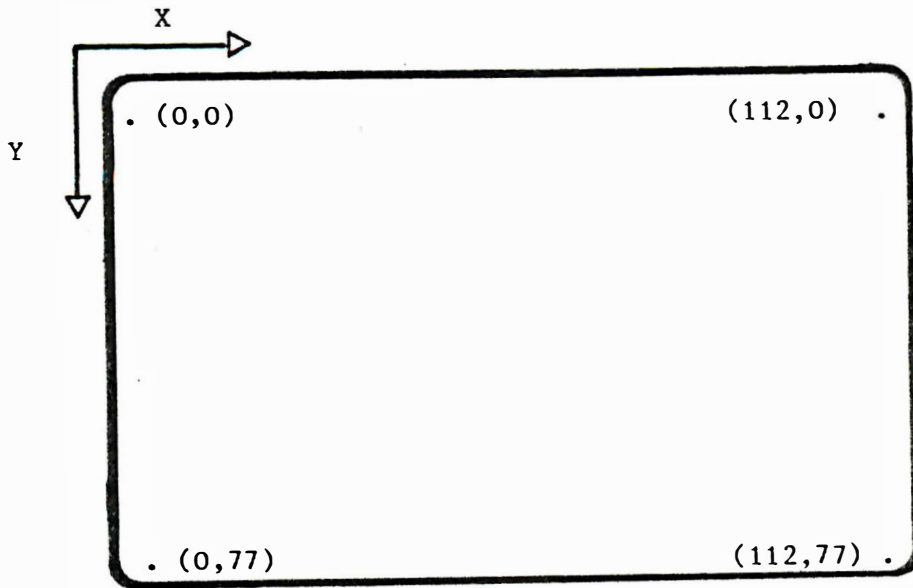
PROGRAM LISTING

The following program listing gives you all the BASIC code for the BASIC
demonstration program included with your Vector Graphics Subroutines.
If you want to see the subroutines themselves, use the Micro Video MONITOR
to display them on the screen. This listing and all the others in this
user guide were produced using an Interact equipped with the Micro Video
RS232 peripheral interface, RS232 BASIC, and a COMPRINT 912-S lineprinter.

```
5 REM VECTOR GRAPHICS DEMO PROGRAM
6 REM COPYRIGHT 1980, MICRO VIDEO CORP.
10 CLS:COLOR7,4,0,1
20 POKE19215,25:POKE19474,94
30 UA=19473:CO=24308:SC=24309:SL=24310:EC=24311
40 EL=24312:X1=24313:Y1=24314:X2=24315:Y2=24316
50 X3=24317:Y3=24318:TF=0:TG=40:DL=107
100 DATA9,6,0,0,0,0,3,4,3,6,4,6,5,4,6,2,6,0,5,0,4
105 DATA3,6,4,0,4,4,0,4,6
110 DATA14,0,2,0,1,2,0,4,0,6,1,6,2,4,3,3,3,4,3,6,4,6,5,4,6,2,6,0,5,0,4
130 DATA8,0.2,0,1,2,0,4,0,6,1,6,2,0,5,0,6,6,6
140 DATA4,1.1,3,0,3,6,1,6,5,6
145 RESTORE
150 FORI=1TO5:READN,X,Y:POKESC,X*10+26:POKESL,Y*10+8
155 POKEUA,DL:POKECO,I-4*INT(I/4)
160 FORJ=1TON:READX,Y:POKEEC,X*10+26:POKEEL,Y*10+8:A=USR(0)
165 POKESC,X*10+26:POKESL,Y*10+8
170 NEXT:P=1:GOSUB9000:POKEUA,TG:POKEX1,56:POKEY1,38
180 POKEX2,0:POKEY2,0
190 POKEX3,112:POKEY3,0:A=USR(0)
200 POKEX2,112:POKEY3,77:A=USR(0)
210 POKEY2,77:POKEX3,0:A=USR(0)
220 POKEX2,0:POKEY3,0:A=USR(0):NEXT
300 CLS:PRINT"    INTRODUCING":PRINT
310 PRINT"  VECTOR GRAPHICS":PRINT:PRINT
320 PRINT"  BY D.J.SCHWAB":PRINT:PRINT
330 P=7:GOSUB9000
340 CLS:PRINT"       WITH":PRINT" VECTOR GRAPHICS"
350 PRINT"    YOU CAN":PRINT"  DRAW LINES,"
360 PRINT" DRAW TRIANGLES.":PRINT:PRINT"   AND MORE!!!":PRINT:PRINT
370 P=7:GOSUB9000
1000 CLS:PRINT"   EXAMPLE 1":PRINT
1010 PRINT"  LINES CAN BE":PRINT"  DRAWN BETWEEN"
1020 PRINT" ANY TWO POINTS":PRINT"  ON THE SCREEN"
1030 PRINT" AND IN ANY COLOR":PRINT:PRINT:P=7:GOSUB9000
1040 CLS:POKEUA,DL:POKESC,56:POKESL,38
1050 FORI=1TO100:X=RND(1)*112:Y=RND(1)*77:C=RND(1)*3+1
1060 POKECO,C:POKEEC,X:POKEEL,Y:A=USR(0):NEXT
1070 P=10:GOSUB9000
1500 CLS:PRINT"   EXAMPLE 2":PRINT
1510 PRINT"  LINES CAN BE":PRINT"   USED TO"
1520 PRINT"   CONSTRUCT":PRINT"    FIGURES":PRINT:PRINT
1530 DIMXA(25),YA(25):PI=3.14159:R=20
1540 FORI=0TO24:XA(I)=R*COS(PI*I/12)
1550 YA(I)=.8*R*SIN(PI*I/12):NEXT
1560 P=1:GOSUB9000:CLS:POKECO,1:POKEUA,DL
1565 XC=56.5:YC=38.5
1570 FORI=1TO24:POKESC,XA(I-1)+XC:POKEEC,XA(I)+XC
1580 POKESL,YA(I-1)+YC:POKEEL,YA(I)+YC:A=USR(0):NEXT
1590 POKECO,3:FORJ=0TO20STEP4:XC=XA(J)+56.5:YC=YA(J)+38.5
1600 FORI=1TO24:POKESC,XA(I-1)+XC:POKEEC,XA(I)+XC
1610 POKESL,YA(I-1)+YC:POKEEL,YA(I)+YC:A=USR(0):NEXT
1620 NEXT:P=10:GOSUB9000
2000 CLS
2010 PRINT"   EXAMPLE 3":PRINT
2020 PRINT"  A SERIES OF":PRINT"   LINES CAN"
```

```
2030 PRINT" MAKE A TRIANGLE":PRINT:PRINT:PRINT
2040 P=7:GOSUB9000:POKECO,1:POKEUA,TG
2050 POKEX1,56:POKEY1,10
2060 POKEX2,13:POKEY2,70
2070 POKEX3,99:POKEY3,70
2080 CLS:A=USR(0):P=10:GOSUB9000
2500 CLS:PRINT"    EXAMPLE 4":PRINT
2510 PRINT"  SOME TRIANGLES":PRINT"   TAKE LONGER"
2520 PRINT"    TO FILL THAN":PRINT"      OTHERS"
2530 PRINT:PRINT:PRINT
2540 P=7:GOSUB9000:POKECO,1:POKEUA,TF
2550 POKEX1,13:POKEY1,70
2560 POKEX2,99:POKEY2,70
2570 POKEX3,56:POKEY3,10
2580 CLS:A=USR(0):P=10:GOSUB9000
3000 CLS
3010 PRINT"    EXAMPLE 5":PRINT
3020 PRINT"   SCREEN SWEEP":PRINT"  WITH ORIGIN AT"
3030 PRINT" CENTER OF SCREEN":PRINT"   USES FOUR"
3040 PRINT" QUICK TRIANGLES":PRINT:PRINT
3050 P=7:GOSUB9000:POKEUA,TG:FORC=1TO3:POKECO,C
3060 POKEX1,56:POKEY1,38:POKEX2,0:POKEY2,0
3070 POKEX3,112:POKEY3,0:A=USR(0)
3080 POKEX2,112:POKEY3,77:A=USR(0)
3090 POKEY2,77:POKEX3,0:A=USR(0)
3100 POKEX2,0:POKEY3,0:A=USR(0)
3110 NEXT:P=1:GOSUB9000
3500 CLS
3510 PRINT"    EXAMPLE 6":PRINT
3520 PRINT"   SCREEN SWEEP":PRINT"  WITH ORIGIN AT"
3530 PRINT" BOTTOM OF SCREEN":PRINT"   USES THREE"
3540 PRINT" QUICK TRIANGLES":PRINT:PRINT
3550 P=7:GOSUB9000:POKEUA,TG:FORC=1TO3:POKECO,C
3560 POKEX1,56:POKEY1,77:POKEX2,0:POKEY2,77
3570 POKEX3,0:POKEY3,0:A=USR(0)
3580 POKEY2,0:POKEX3,112:A=USR(0)
3590 POKEX2,112:POKEY3,77:A=USR(0)
3600 NEXT:P=1:GOSUB9000
4000 CLS
4010 PRINT"    EXAMPLE 7":PRINT
4020 PRINT"  A CIRCLE MADE":PRINT"   OF 12 SLOW"
4030 PRINT"    TRIANGLES":PRINT:PRINT:PRINT
4035 P=7:GOSUB9000:CLS
4040 XC=56.5:YC=38.5:POKECO,1:POKEUA,TF
4050 POKEX1,XC:POKEY1,YC
4060 FORI=2TO24STEP2:POKEX2,2*XA(I-2)+XC
4070 POKEY2,2*YA(I-2)+YC:POKEX3,2*XA(I)+XC
4080 POKEY3,2*YA(I)+YC:A=USR(0):NEXT
4090 P=10:GOSUB9000
4500 CLS
4510 PRINT"    EXAMPLE 8":PRINT
4520 PRINT"  A CIRCLE MADE":PRINT"   OF 24 SLOW"
4530 PRINT"    TRIANGLES":PRINT:PRINT:PRINT
4540 P=7:GOSUB9000:CLS
4550 FORI=1TO24:POKEX2,2*XA(I-1)+XC
4560 POKEY2,2*YA(I-1)+YC:POKEX3,2*XA(I)+XC
4570 POKEY3,2*YA(I)+YC:A=USR(0):NEXT
4580 P=10:GOSUB9000
```

```
5000 CLS
5010 PRINT"    LET YOUR":PRINT
5020 PRINT"  IMAGINATION":PRINT
5030 PRINT"   RUN WILD !":PRINT:PRINT
5040 P=7:GOSUB9000:CLS:POKEUA,TF:XC=56.5:YC=38.5:C=2
5050 FORI=1TO23STEP2:POKEX1,2*XA(I)+XC
5060 POKEY1,2*YA(I)+YC:POKEX2,XA(I-1)+XC
5070 POKEY2,YA(I-1)+YC:POKEX3,XA(I+1)+XC
5080 POKEY3,YA(I+1)+YC:POKECO,C:A=USR(0)
5090 POKEX1,XC:POKEY1,YC:POKECO,C-2:A=USR(0)
5100 C=C+1:IFC=4THENC=2
5110 NEXT:P=15:GOSUB9000
5500 CLS:C=0:POKEUA,DL:POKESC,56:POKESL,38:POKEEL,0
5510 FORI=0TO112:GOSUB6000:NEXT
5520 FORI=1TO77:GOSUB6010:NEXT
5530 FORI=111TO0STEP-1:GOSUB6000:NEXT
5540 FORI=76TO0STEP-1:GOSUB6010:NEXT
5545 X=RND(1)*112:POKESC,X:Y=RND(1)*77:POKESL,Y
5550 GOTO5510
6000 POKEEC,I:GOTO6020
6010 POKEEL,I
6020 POKECO,C:C=C+1:IFC=4THENC=0
6030 A=USR(0):RETURN
9000 FORK=0TO500*P
9010 NEXT:RETURN
```

MICRO *m* VIDEO™

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107
(313) 996-0626

DATALOG

Program Instructions

DATALOG is a BASIC program that lets you store an arbitrary number of
files containing string or numeric data on cassette tape. You can later
retrieve files either sequentially or by name and display the contents
of the file on the TV screen. DATALOG turns a cassette tape into a library.
You are the librarian, retrieving selected files or adding new ones onto
the end of the tape.

With DATALOG, the number and size of files are not limited by the free
RAM. A file can contain one or more screens of data, with automatic
paging in the file as required. Optional password protection may be
assigned to any screen of data in the file. This feature provides con-
fidentiality for some or all of the data in the individual files.

Additionally, DATALOG presents a technique for saving "string data" on
cassette tape with the BASIC CSAVE command. We have included a complete
program listing so that you can understand the techniques used in the
program and modify them to meet your particular requirements.

LOADING THE PROGRAM: Insert the Microsoft 8K BASIC (or Level II BASIC)
tape into the cassette deck. Depress the RESET button, press the "L"
key, rewind the tape if necessary, then depress the READ cassette button
to load the BASIC interpreter into the computer. When the BASIC tape
is loaded and the "OK" prompt appears, remove the BASIC cassette. Insert
the DATALOG program cassette into the tape deck, depress the RESET button,
and repeat the loading sequence as described above. (You can use the
CLOAD command in BASIC to load DATALOG; however, we recommend that in
general you load programs with the RESET-"L" procedure.) Type RUN and
press the "CR" key to execute the DATALOG program.

RUNNING THE PROGRAM: You control the program's operation by entering
responses as appropriate to questions that appear on the screen. No
entertainment controllers are used. The first question that appears
asks you if you want to read from a tape or write to one (R=READ, W=WRITE).
Press the "R" key if you want to read data already placed on a cassette
tape and display the information on the screen. Press "W" if you want
to enter information from the keyboard and write it to tape. At the
end of any read or write operation, this query screen reappears, allowing
you to perform read and write operations in any order convenient for
you.

After you have selected either the READ or WRITE operation, the program
asks if you need instructions for the operation you have selected (TAPE
INSTRUCTIONS? Y=YES, N=NO). We recommend that you respond "Y" to this
question until you are fully acquainted with the operation of the program,
as described below.

TAPE SET-UP

If you are using DATALOG for the first time, you have nothing stored
on tape, so insert a blank cassette into the tape drive in preparation
for writing data to tape. If you have previously stored information
on a tape in DATALOG files and wish to retrieve (R=READ) that information
or to put additional files on the end of that tape (W=WRITE), then insert
your DATALOG data tape into the cassette drive.

The "SELECT FUNCTION" screen reminds you to position your tape by depressing
the READ, FAST-FORWARD, or REWIND cassette keys as appropriate. If you
are writing onto a new tape, REWIND the tape, then depress the READ button
for a few seconds to position the tape beyond the non-magnetized tape
leader. If you wish to add new files to the end of the tape, use the
FAST-FORWARD cassette button to advance the tape quickly, then depress
the READ button and listen to the data tones through the TV speaker.
When you hear no further data read sounds, you have reached the end of
the recorded information. Note that the program is not reading the data
into the computer during this positioning operation.

Once the tape is positioned properly, press any key to stop the tape
motion. The computer then instructs you to depress the READ button on
the cassette drive (R=READ) or both the READ and WRITE cassette buttons
for the writing operation.

WRITING FILES

If you have elected to write data to tape, the computer asks "FILE MARKING?
Y=YES, N=NO".

File marking associates a name of up to 10 characters to a data file.
This allows you to selectively retrieve files from a single cassette
at a later time by having the computer search for a particular file.
Respond by pressing "Y" to name your file(s), or press "N" if you will
be content to retrieve the contents of the files through a sequential
search.

If you select file marking, the computer prompts you for the name you
want to associate with that file. Enter a file name of up to 10 characters
and press the "CR" key. The file name is then written to the tape.

The computer then prompts you to "ENTER DATA", followed by a blank screen.
Type in the textual or numeric data as desired. Note that the program
takes in each keystroke individually, so you may have to reduce your
typing speed accordingly. (You will hear a tone acknowledging entry
of each character. Press the "CR" key at the end of each line to enter
the line.) Individual lines are limited to 16 characters, and the computer
will not accept characters past the end of the line.

To enter a blank line, type in at least one space and press the "CR"
key. To terminate entry of a particular file, press the "CR" key a second
time. The computer will write whatever data precedes the terminating
carriage return to tape.

When the screen is full, the program automatically writes the data to
the cassette tape, asking you to stand by. This "automatic paging" feature
of the DATALOG program limits your files only by the length of the cassette
tape--not the screen size or available RAM.

Each screen can contain a maximum of 9 lines or 16 or fewer characters
each. The program reads and writes the data on a screen in a 180-character
length block.

After you have terminated file input by typing the extra "CR" and the
last screen (page) has been written to tape, the computer prompts you
for then next step to be performed ("SPACE BAR TO CONTINUE RECORDING,
CR KEY TO PLAY BACK"). Depress the space bar if you want to input another
file from the keyboard and write it to tape. To retrieve files from
tape at this point, press the "CR" key to initiate the READ operation.
You have approximately 5 seconds to respond to the query. If you do
not enter a response, the computer repeats the initial program screen
(R=READ, W=WRITE) and requests that you specify the operation to be performed.

READING FILES

If you elect to read data from the tape, the computer asks "FILE SEARCH?
Y=YES, N=NO". Press the "Y" key to retrieve and display the contents
of a file that has a specified name. Respond "N" to examine your files
in a sequential search without regard to any file names that may exist.

If you respond "Y", the computer asks for the name assigned to the file
you wish to retrieve and display. Enter the file name from the keyboard.
DATALOG then reads through the files on the tape, starting at the current
tape position, searching for the file with the name you specified. As
it searches the tape, the computer displays the names of other files
it encounters during the search. When the named file is found, DATALOG
sounds a siren and waits for the next instruction. This feature allows
your to perform an unattended search with an audible notification of
success as soon as the specified file is found and loaded.

If you respond "N" to the file search query, the files are searched sequentially
and the assigned file names, if any, are displayed as each file is read
into the computer. Press the space bar to display any file of interest
during the search. The computer automatically pages through the "screens"
of that file. During file display, you can press the space bar to hold
a particular screen image indefinitely. When you press the space bar
again, the computer continues its display and/or search operation.

Two other characters can affect the file reading operation. Press:

> * - to reenter the file search mode and specify a *particular*
>   file name to be found and loaded.

"CR" - to switch from file reading to file writing mode of operation.

PASSWORD PROTECTION

Any individual screen of information can be "protected" against unauthorized
viewing by entering a user-specified password on the first line of that
screen. A password line is distinguished from an ordinary line of text
by concluding it with an asterisk (*). Passwords may be up to 10 characters
in length, and no other text can be entered on the same line as the password.
Enter the password by pressing the "CR" key after you type the asterisk.
For example, if you name a file "RECIPE" and want to assign the password
"SECRET" to it, you would type the following as the first line of each
screen (to protect the entire file):

    SECRET* cr

Then, before a screen containing protected information will be displayed,
you will be required to enter the password associated with that screen.
To the prompt, enter the password (no asterisk), followed by "CR".
The computer gives you four chances to enter the correct password. If
you do not enter the password correctly after four tries, DATALOG "self-
destructs" to prohibit further tampering by the inexperienced or unauthorized
operator. Your files on tape will not be altered in any way, however.
We highly recommend that you keep some record of file names and associated
passwords to eliminate the frustration of forgetting a password and being
unable to access that data.

STOPPING AND RESTARTING DATALOG

At any time during program operation, you may halt execution by typing
a Control-C (pressing the Control and C keys simultaneously). You can
restart the program from the beginning by typing RUN. Data that has already
been written to tape will not be affected by halting the program in this
way. However, if you have any data that has not been written to tape
at the time you stop the program with a Control-C, that data will be
lost.

PROGRAM LISTING

The DATALOG program is listed in its entirety on the following pages.
The listing was produced using the Micro Video RS232 interface, RS232
BASIC, and a COMPRINT 912-S printer. For more information on these and
other accessories for your Interact computer, or for information on our
printer service, contact Micro Video, P.O. Box 7357, 204 E. Washington,
Ann Arbor, MI 48104.

```
@
1 REM DATALOG  MICRO VIDEO - ANN ARBOR
10 POKE19215,25:CLS:CLEAR(200):PRINTCHR$(8)
30 COLOR0,0,0,0:GOSUB1000
40 OUTPUT"DATALOG",37,55,0:OUTPUT"R=READ  W=WRITE",11,15,3
50 COLOR0,1,4,3:K$=INSTR$(1):GOTO5000
230 A$=J$
240 IFASC(A$)=3THEND(M)=3:CSAVE*D:M=0:RETURN
245 IFASC(A$)=4THEND(M)=4:CSAVE*D:M=0:RETURN
250 A$=A$+CHR$(13)
260 GOSUB360
270 IFC=13THEND(M)=C1:GOSUB700:RETURN
280 A$=MID$(A$,L+1,LEN(A$))
290 D(M)=C1:GOSUB700
310 IFLEN(A$)<>0THENGOSUB360
320 IFC=13THEND(M)=C1:GOSUB700:RETURN
330 A$=MID$(A$,L+1,LEN(A$))
340 IFLEN(A$)=0THEN RETURN
350 GOTO290
360 L=LEN(A$)
370 IFL>2THENL=3
380 C1=0
390 FORP=1TOL
400 C=ASC(MID$(A$,P,1))
410 C1=(C1*100)+C
420 IFC=13THENRETURN
430 NEXTP
460 RETURN
480 IFF<>0THEN510
482 IFN=5THEN500
485 OUTPUT"READING",36,60,3:OUTPUT"TAPE",45,52,1
487 IFSB=5THEN500
490 OUTPUT"PLEASE STAND-BY",11,25,2
500 N=5:F=1:IFM=0THENCLOAD*D
510 J$=CHR$(0)
520 C1=D(M):IFC1<5THENJ$=MID$(J$,2,LEN(J$)):RETURN
530 GOSUB570:GOSUB740:IFC1=13THENJ$=MID$(J$,2,LEN(J$)):GOSUB9000:RETURN
560 GOTO520
570 Z$=STR$(D(M))
580 FORU=2TOLEN(Z$)STEP2
590 C1=VAL(MID$(Z$,U,2))
600 IFC1=13THENRETURN
610 GOSUB670
620 NEXTU
630 RETURN
670 IFC1<4THENRETURN
680 J$=J$+CHR$(C1)
690 RETURN
700 M=M+1
720 IFM>ZTHENCSAVE*D:M=0
730 RETURN
740 M=M+1
750 IFM>ZTHENCLOAD*D:M=0
760 RETURN
1000 FORY=50TO60STEP5
1010 FORX=25TO85STEP5
1020 OUTPUTCHR$(1),X,Y,2
1030 NEXT:NEXT
```

```
1040 FORX=35TO75STEP5
1050 OUTPUTCHR$(1),X,53,3:OUTPUTCHR$(1),X,55,3:OUTPUTCHR$(1),X,57,3
1060 NEXT:RETURN
2050 CLS:OUTPUT"PRESS",41,55,1:OUTPUT"READ+WRITE",26,45,3
2060 OUTPUT"PRESS ANY KEY",17,15,2:A$=INSTR$(1)
2070 CLS:OUTPUT"FILE  MARKING ?",13,70,3
2080 OUTPUT"Y=YES.   N=NO",19,40,2:A$=INSTR$(1)
2090 IFA$="Y"THEN2170
2100 CLS:Z=60:V=70:DIMD(Z)
2105 OUTPUT"ENTER DATA",26,55,3:FORX=1TO200:NEXTX:CLS
2110 H=10:GOSUB6000
2115 IFB$=""THENGOSUB2145:GOTO8000
2120 J$=B$:B$=""
2125 GOSUB230:TONE100,25
2130 V=V-7
2135 IFV<10THENCLS:V=70:GOTO2110
2140 GOTO2110
2145 CLS:OUTPUT"RECORDING",30,60,3
2150 OUTPUT"PLEASE STAND BY",10,20,1
2152 IFX=5THENJ$=CHR$(4):X=0:GOSUB230:CLS:RETURN
2160 J$=CHR$(3):GOSUB230:CLS:RETURN
2170 CLS:OUTPUT"ENTER NAME",26,70,2:OUTPUT"OF FILE",35,62,1
2175 H=45:V=45:Z=60:DIMD(Z)
2180 GOSUB6000:J$=B$:GOSUB230:X=5
2190 GOSUB2145:CLEAR:GOTO2100
3000 CLS:OUTPUT"TAPE SET-UP",23,70,2:OUTPUT"_____",5,63,1
3010 OUTPUT"INSERT DATA TAPE",8,50,3:OUTPUT"SELECT FUNTION",14,40,1
3020 OUTPUT"WHEN COMPLETED",14,30,3:OUTPUT"PRESS ANY KEY",17,20,2
3030 REWIND:IFK$="W"THENGOTO2050
3040 CLS:OUTPUT"PRESS READ",26,50,3
3050 OUTPUT"PRESS",41,25,2:OUTPUT"ANY KEY",35,16,2
3070 A$=INSTR$(1)
3080 CLS:OUTPUT"FILE SEARCH",25,70,2:OUTPUT"Y=YES   N=NO",22,45,3
3090 A$=INSTR$(1)
3095 IFA$="Y"THEN3200
3100 CLS:CLEAR:Z=60:DIMD(Z)
3110 CLS:V=70
3115 GOSUB480:IFC1=4THENOUTPUT"FILE NAME",29,45,2
3116 IFC1<5THENT=100:GOSUB7000:GOSUB3127:GOTO3140
3117 L=LEN(J$):L=L*6
3118 H=(112-L)/2:IFN=5THENCLS
3120 OUTPUTJ$,H,V,3
3122 V=V-7:N=0
3125 IFV<10THENT=100:GOSUB7000:GOSUB3127:CLS:GOTO3110
3126 GOTO3115
3127 IFN<255THENA$=INSTR$(1)
3128 RETURN
3140 CLS:OUTPUT"END OF FILE",23,70,3:OUTPUT"_____",5,67,2
3150 OUTPUT"CR KEY TO RECORD",8,57,1
3155 OUTPUT"SPACE-BAR",29,45,3:OUTPUT"FOR NEXT ENTRY",14,37,3
3160 OUTPUT"PRESS * KEY",26,24,2:OUTPUT"FOR FILE SEARCH",11,17,2
3165 T=550:GOSUB7000:IFN=255THEN10
3175 IFN=254THEN3200
3180 IFN=251THENCLEAR:K$="W":GOTO5000
3190 CLEAR:GOTO3100
3200 CLS:CLEAR:SB=5:H=45:V=40:Z=60:DIMD(60)
3210 OUTPUT"ENTER FILE NAME",11,70,2
3215 GOSUB6000
```

```
3217  OUTPUT"FILE",14,24,2:OUTPUT"NAME",14,16,1
3218  T$="":GOSUB3250
3220  A=PEEK(14336):IFA=253THENCLS:M=0:F=1:N=5:SB=0:GOTO3110
3221  GOSUB480
3222  T$=J$
3225  IFJ$=B$THENGOSUB3250:SOUND0,24844:A$=INSTR$(1):GOTO3100
3230  GOSUB480
3235  IFC1=4THENTONE100,50:GOSUB3250
3240  M=0:F=0:GOTO3220
3250  FORX=42TO102STEP4
3252  OUTPUTCHR$(1),X,22,3:OUTPUTCHR$(1),X,18,3
3253  NEXTX
3255  L=LEN(T$):L=L*6
3256  H=((66-L)/2)+42
3257  OUTPUTT$,H,20,0
3260  RETURN
5000  CLS:OUTPUT"TAPE",44,70,1:OUTPUT"INSTRUCTIONS",20,62,1
5020  OUTPUT"Y=YES    N=NO",20,40,3
5030  A$=INSTR$(1):IFA$="Y"THEN3000
5040  IFK$="W"THEN2050
5050  GOTO3040
6000  A$=INSTR$(1)
6010  IFA$=CHR$(13)ORA$=CHR$(8)THEN6100
6025  IFH>100THENTONE250,20:GOTO6000
6030  OUTPUTA$,H,V,3:B$=B$+A$:H=H+6:GOTO6000
6100  IFA$=CHR$(13)THENRETURN
6110  IFLEN(B$)=0THEN6000
6120  H=H-6
6130  OUTPUTCHR$(1),H,V,0
6140  B$=LEFT$(B$,LEN(B$)-1)
6150  GOTO6000
7000  TONE100,100
7010  FORA=1TOT
7020  N=PEEK(14336)
7030  IFN<255THENFORA=1TO100:NEXTA:RETURN
7040  NEXTA:RETURN
8000  CLS:OUTPUT"SPACE-BAR",29,70,3:OUTPUT"TO CONTINUE",23,57,1
8010  OUTPUT"RECORDING",29,49,1:OUTPUT"CR KEY",38,30,3
8020  OUTPUT"TO PLAYBACK",23,20,2
8030  CLEAR:T=400:GOSUB7000:IFN=251THEN5000
8040  IFN<255THENCLEAR:GOTO2070
8050  GOTO10
9000  T=0:IFSB=5THENRETURN
9005  A$=RIGHT$(J$,1):IFA$="*"THENJ$=LEFT$(J$,LEN(J$)-1):GOTO9010
9007  RETURN
9010  CLS:OUTPUT"RESTRICTED FILE",11,70,3
9020  OUTPUT"ENTER",41,55,2:OUTPUT"PROPER CODE",23,47,2
9030  B$="":H=40:V=35:GOSUB6000
9040  IFB$=J$THENN=5:F=1:SS=0:GOTO3110
9050  T=T+1:IFT>4THEN10
9060  GOTO9010
```

MICRO ◼ VIDEO

305 North First St.
P.O. Box 7357
Ann Arbor, MI 48107
(313) 996-0626

AL

The Editor, Assembler, and Monitor

Welcome to assembly language programming on your Interact!

As you know, your computer operates with an Intel 8080A microprocessor.
With AL, you can use standard 8080 mnemonic instructions to program your
computer, gaining the speed of direct machine access without sacrificing
the simplicity of a structured language. Assembly language provides an
easy, efficient means of programming in machine language, using a standard,
readable notation that does not require you to know or specify absolute
hexadecimal addresses. In fact, AL is "multi-lingual" – you can program
in decimal, hex, octal, binary, and ASCII.

AL's Editor, Assembler, and Monitor form a powerful programming team in
a single program. The Editor lets you enter and modify your assembly language
instructions. Then, the Assembler processes the instructions to build
a symbol table and convert the source code into machine-readable (object)
form. The Monitor allows you to manipulate the object code directly, and
make special patches or tests. You can read and write tapes with AL, as
well as dump your symbol table, source code, even the contents of memory
to your printer. Best of all, AL occupies only 4K of RAM.

NOTE: AL is intended for advanced programmers. This document is not
intended to teach you how to program in 8080 assembly language, but rather
to explain the operation of this particular assembler and editor. We have
assumed that you have a working knowledge of 8080 instruction mnemonics.
If you do not, you will need a manual that explains the concepts and
instruction codes in depth. Two suggested reference documents are Kathe
Spracklen's Z80/8080 Assembly Language Programming and Intel Corporation's
Intel 8080 Assembly Language Programming. The Spracklen book is available
from Micro Video; you can order the Intel manual directly from Intel Cor-
poration.

Producing a program with AL involves 4 fundamental steps.

1) Building and editing a source file with the Editor.

2) Performing a first assembly pass on the source code to produce a symbol
    table.

3) Performing a second assembly pass to convert the source code to object
    (hexadecimal) form.

4) Making corrections to the source file or merging multiple source files
    into a single unit with the Monitor.

Copyright 1983, Micro Video Corporation

Load AL by pressing the RESET button, pressing the 'L' key, rewinding the tape if necessary, then depressing the READ cassette button.

When the program has loaded, AL signifies his readiness to accept commands by displaying the "greater than" (" > ") symbol. Whenever you see this prompt, you are at AL's command level.

>I

When the readiness prompt reappears, the command has been executed. If, at any time, you give AL an invalid command or a command in an improper format, he will respond with a question mark ("?").

>Z
?
>IT
?
>

In the examples in this document, information you type is presented in reverse type. The computer's responses are in **bold** type.

If a command is to be followed by a parameter, you must put a space between the command and the parameter.

AL recognizes two special characters with regard to editing and command operation. These characters are the carriage return and backspace. The effects of these characters on function operation are as follows:

(cr) — Carriage Return. This key is used to execute all functions and to store lines of code in memory during program entry or editing. In all examples in this document, the carriage return is represented as **(cr)**.

(bs) — Backspace. You can use the backspace key to correct errors in a line before you have pressed (cr). You can also use it in certain functions to erase characters in a line. In all examples in this document, the backspace key is represented as **(bs)**.

MEMORY MAP

AL's own object code resides at locations 4C00H to 4CBBH and 6000H to 6FFFH. These locations should therefore be off limits for your text and object code.

The following table shows the default locations for your text and symbol table.

	16K RAM	32K RAM
text area	7000H–7FFFH	7000H–0B6FFH
symbol table area	5B80H–5F80H	0B700H–0BFFH

If you want to increase or decrease the size of your text area use the TEXT command.

Example:

    TEXT 7200H, 7EFFH

This changes the storage buffer for your source code from the default location to the area between and including 7200H through 7EFFH. This feature is useful when you need a larger object code area than default conditions provide.

Similarly, you may also manipulate the symbol table area using SYMT.

Example:

    SYMT 5C00H, 5F80H

Here your symbol table area starts at 5C00H and is 380H bytes long.

If you reset these areas, the new settings remain in effect even if you restart AL operation with RESET-R.

The **NEW** command clears the text buffer so that you can begin with a clean slate. Normally, you will use this command when AL is first loaded as a precaution to insure that the buffer pointers are correctly set.

FREE tells you the total number of bytes available for source code. Each character uses up one byte, so, essentially, this command lets you know how many more characters can be input. Normally, FREE returns 46FFH in a 32K machine (0B6FFH minus 7000H) or 0FFFH in a 16K machine (7FFFH minus 7000H) when the text buffer is empty and the default parameters are in effect.

The Editor is a powerful 8080 program editor that makes it easy for you to prepare your source code file(s) for assembly. You enter the lines containing 8080 mnemonic instructions to be processed and converted to hexadecimal form. The Editor automatically numbers the lines you type. It has commands that let you display all or part of your source code; print it out on the screen or a printer; change the contents of entire lines; delete or add lines with automatic resequencing; relocate lines; edit within a line using special control characters; replace character strings; and store and retrieve source code with cassettes.

Following is a brief walk-through of the Editor operation.

ENTERING YOUR SOURCE CODE

To begin entering lines of code, use the I (Input) command. Type:

> `I(CR)`

All of AL's commands are executed by pressing the (cr) (carriage return) key. You also use (cr) to conclude each line of source code you enter and store it in memory.

AL responds to the Input command with a line number. Begin typing your lines of source code, ending each with (cr). You do not have to number your lines; each is numbered automatically at the left side of your screen. To end the source code entry, type (cr) only after the next line number is displayed.

Consider the following program entry sequence:


```
>   I(cr.)
1   ;SAMPLE SOURCE CODE
2   ORG 5000H
3   BEGIN: XRA A
4   CLS  SET 573H
5   CHRCOL SET 62EH
6   (cr)
>
```


AL returns to command level after entry of 5 lines, because (cr) was pressed in response to the line 6 prompt.

To see the code you've entered, including assigned line numbers, use the
L (List) command.

```
> L(cr)
1  ; SAMPLE SOURCE CODE
2  ORG 5000H
3  BEGIN: XRA A
4  CLS SET 573H
5  CHRCOL SET 62EH
```

You can begin listing from any point in your source code by including
the desired line number with the L command (remember to put a space after
the L).

```
> L 4(cr)
4  CLS SET 573H
5  CHRCOL SET 62EH
```

You can also specify a label name instead of a number for listing.

```
> L BEGIN(cr)
3  BEGIN: XRA A
4  CLS SET 573H
5  CHRCOL SET 62EH
```

The above command starts the listing at line 3, since that's where the
label BEGIN occurs.

To examine a range of lines, specify the beginning and ending lines in
the range with the L function, separating them with a comma (,).

```
> L 2,4(cr)
2  ORG 5000H
3  BEGIN: XRA A
4  CLS SET 573H
```

You can also replace the first parameter of the above format with a label
name, though the second must be a number. This rule is universal with
respect to any two-parameter command in AL.

```
> L BEGIN,4(cr)
3  BEGIN: XRA A
4  CLS SET 573H
```

Note that you can temporarily halt the listing at any time by typing any
key. Press any key except (cr) to continue. (cr) terminates the listing.

SOURCE CODE FORMAT

You begin a line with any character other than the following:

```
COLON (:)
SEMICOLON (;)
LINEFEED
SPACE
CARRIAGE RETURN
```

Each line is considered to consist of a series of variable length fields containing labels, 8080 mnemonics, operators, and/or data.

FIELD FORMAT

Fields are variable in length and are classified by their serial position in the line, their contents, or the contents of preceding fields. Fields are classified as containing one of the following; a label, a mnemonic, a pseudo-operator, an operand, an address, a name, a number (address, instruction, or data), or a comment. Several characters act as delimiters of fields:

    SPACE   – general field delimiter
    COMMA (,)   – register operand field
    SEMICOLON (;)   – comment field
    COLON (:)   – label field

LABELS

Labels are used to identify the beginning of a set of instructions for easy access from other parts of the program. Any line of code may be assigned a label. A label may be up to six characters in length and must be delimited by a colon (:). If a label is assigned to a line, it must appear in the first, or leftmost, field.

Labels are subject to the following conventions: ·

1) The first character must be a letter of the alphabet (A–Z).

2) A label may contain any combination of six letters and numbers, but must not contain other symbols. For example,

        A12345:
        EXAMPL:
        C23:

    are legal labels, while the following are not:

        72ABC:
        A+B:
        ?ZRPM:

3) A label may only be assigned once in any set of source code instructions.

NAMES

Once a series of letters and numbers has been defined as a label, the label is considered to be a name. The name may then be used as an address-defining reference to complete CALL, JMP or other memory addressing instructions appearing in other parts of the source code. A name may also be used with the SET pseudo-operator to define a constant value for use as a "data" operand.

HEXADECIMAL NUMBERS

AL accepts hexadecimal values as operands. However, since the letters
A-F are used in hex notation, any hexadecimal number that begins with
one of these letters must be preceded by a zero. Otherwise, the hex value
will be assigned as an undefined label. All values given in hex notation
must end with the letter H. For example, the hex value F72E must be repre-
sented to AL as 0F72EH.

OTHER NUMBER SYSTEMS

AL also accepts numbers represented in four other bases besides hexadec-
imal - decimal, binary (which must be terminated by a B), octal (terminated
by a Q), and base four (terminated by an R). Below are a few numbers and
how they would be represented to AL in each of the five bases:

hex	decimal	octal	base 4	binary
0BAH	176	272Q	2322R	1011 1010B
0AH	10	12Q	22R	1010B
0FFH	255	277Q	3333R	1111 1111B

SYMBOLS

You can use a symbolic name as an operand to represent as address, data,
or other value, as long as the name satisfies the criteria previously
defined for naming labels.

ASCII CHARACTERS

You can use a single ASCII character enclosed in single quotation marks
(e.g., "B") as an operand. Any ASCII character normally processed by AL
may be used, with the exception of the delimiting characters SPACE, COLON
(:), SEMICOLON (;), COMMA (,), CARRIAGE RETURN and LINEFEED. You can,
however, specify these delimiting characters as operands, provided you
express the ASCII value in hexadecimal notation, as given below:

SPACE	20H
COLON (:)	3AH
SEMICOLON (;)	3BH
(cr)	0DH
LINEFEED	0AH

For all other characters, AL will automatically convert the given ASCII
character representations to their hex values.

EXAMPLES OF MNEMONIC FORMAT

The example below merely illustrates some of the possible mnemonic formats
that can be used with AL. The example is designed to illustrate formats
only, not actually to do anything.

```
LABEL:   CALL CHRCOL
         MOV A,B
         LXI H,56F2H
         JMP BEGIN
BEGIN:   CALL 0573H
         MVI C,OFFH
         'S'
         JMP 4C06H
         MVI A,'R'
         LXI H,'AB'
```

PSEUDO-OPERATORS

In addition to the standard 8080 mnemonics, AL recognizes six powerful
pseudo-operators (or, "pseudo-ops"). Pseudo-ops are special instruction
mnemonics that, while not a part of the standard 8080 instruction set,
can be used to direct the operation of AL. These pseudo-ops are discussed
below.

ORG

The ORG (ORiGinate) pseudo-op is always used at the beginning of the source
code to tell AL where to set the pseudo-program counter when it begins
to assemble a program. It also can be used to specify an area of memory
for the operation of a particular part of the program.

The ORG instruction must appear in the first field on the line, and is
followed by a valid mnemonic operand that specifies the starting address
as two bytes of object code. For example,

```
          ORG 4C00H
```

directs the assembler to begin producing object code and store the first
byte produced at location 00 of "page" 4C (hex).

Note that if you store your program source file in several sections to
later be merged into a single program, a starting address must be speci-
fied with ORG at the beginning of the source for each section.

DS

The DS (Define Storage) pseudo-op allows you to set aside an area in memory
for variable storage. When AL processes a "DS" mnemonic, it adds the value
of the operand associated with the mnemonic to the current value of the
pseudo-program counter. This effectively causes the pseudo-counter to
skip over a block of addresses.

Example: DS 0100H

DB

The DB (Defined Byte) pseudo-op lets you define a constant or string.
The constant is defined by following the "DB" mnemonic with a hexadecimal

value or a mnemonic operand. The DB mnemonic inself does not generate
any object code. However, the defined byte specified by the operand is
generated by AL during the assembly process.

For example,

    DB 5FH

generates a byte of code with the value 5FH. The following example would
instruct AL to produce a byte of object code containing the ASCII binary
pattern for the letter 'R'.

    DB 'R'

You can also define a string with DB. The example below defines a string
called MESSAGE followed by a null.

    DB 'MESSAGE',0

The only characters not allowed in strings are semicolons and apostrophes.

DW

The DW (Define Word) pseudo-op is an extension of the "DB" concept. In
this case, a "word" is defined to mean a "double byte". The DW operator
is followed by a two-byte operand. For example,

    DW 3672H

The assembler produces object code in the format LSB/MSB. Therefore, in
the above example, the assembler would produce a 72 and then generate
the value 36 as object code.

SET

The SET pseudo-operator can be used to assign a value to a name. The format
for using the SET pseudo-op is slightly different from standard mnemonic
format in that the pseudo-operator mnemonic does not come first in the
line. Instead, the symbolic name to be set to a value heads the line.
That symbolic name is followed by the mnemonic SET and a mnemonic operand
representing a value that the name is to assume. For example,

    CHRCOL SET 0636H

This instruction sets the value of CHRCOL to 0636H. Note that there is
no COLON (:) following the symbolic name.

END

The END pseudo-op may be used in a source file, but is optional. Normally,
assembly stops when the last line of the source code is assembled, but
an END statement will cause assembly to cease immediately, no matter where
it occurs in the source program.

BRK

This pseudo-op assembles a break point in the object code so that when the program is run for testing purposes, it may be interrupted. BRK is useful for program development, but should be removed from the source code file before final assembly.

EDITING YOUR SOURCE FILE

AL allows you to edit the individual lines of code in your source file and to add or delete lines as necessary.

ADDING LINES

You can add lines to your source code by using the I (Input) command described earlier. Simply type I(cr), and AL will prompt you with the next line number, letting you add text to the end of your existing code.

You can also insert new lines between existing lines of file using the I function. To insert lines, you specify where in your code the new lines are to be added when the number following the I command tells AL to add new lines immediately before that line. You can add as many lines as you wish with this function. Exit the insert mode by typing (cr) right after the next line number appears; AL then add the new lines and renumbers the source file to incorporate them.

For example, say that you have entered a block of code, but believe you have omitted some instructions. First, display the area in question:

```
>  L 10,14(cr)
10  START: CALL CLS
11  LXI B,OFFFFH
12  CALL SETSCR
13  MVI C,1
14  CALL CHRCOL
```

To add two new lines between line 11 and 12 use the I command with a parameter.

```
>  I 12(cr)
12  CALL DELAY(cr)
13  MVI C,3(cr)
14  (cr)
```

Now, if you display lines 10 through 16, you'll see that the new lines have been added and the remaining lines in the file renumbered.

```
>  L 10,16
10  START: CALL CLS
11  LXI B,OFFFFH
12  CALL DELAY
13  MVI C,3
14  CALL SETSCR
15  MVI C,1
16  CALL CHRCOL
```

DELETING LINES

You can remove a single line or range of lines from your source file using the DEL (Delete) command. AL automatically renumbers your source file after deleting the specified line(s). You must give a starting line number or label and a range of lines if you want to eliminate more than just a single line. Remember, if you wish to delete <u>all</u> lines of source code from your file, use the NEW command.

For example, to remove the single line 10 from the previous example, type:

> `DEL 10(cr)`

To erase lines 10 through 12 (inclusive) from your file, enter:

> `DEL 10,12(cr)`

Use the L function after erasing the lines to get an update on the revised line numbers.

```
1  ORG 5000H
2  CALL SETSCR
3  LDA 5800H
4  INR A
5  STA 5800H
```

RELOCATING LINES

You can relocate a line or group of lines in your source file with the M (Move) command. The command takes three parameters. The first is the starting line number. It is followed by the last line number to be moved, and the third parameter indicates the line number in front of which the block is to be relocated.

Let's say that in the above example, you wish to move lines 3 and 4 before line 2. You would use the M (MOVE) command as follows:

> `M 3,4,2(cr)`
> `L(cr)`
```
1  ORG 5000H
2  LDA 5800H
3  INR A
4  CALL SETSCR
5  STA 5800H
```

MODIFYING LINES OF CODE

You have three options for altering a line of code — the C (Change) command to completely modify the contents of a specific line or lines, the MOD (Modify) command to edit a line without having to retype it completely, or the REP (REPlace) command to change a character string in one or many lines.

C (Change)

This command requires you to retype the entire line you want to change. Use this command when the corrections to a line are extensive enough to warrant total retyping. For minor corrections, such an inserting or deleting characters, you will find the MOD command move convenient and useful.

To change the contents of a single line, type the C command, followed by the line number of the line to be altered. For example, if line 3 contains the comment

        3   ;ILLUSTRATING THE USE OF

and you want to change it completely, you might type

        >   C 3(cr)
        C   ;USING THE(cr)
        4   (cr)

If you are changing only one line, make sure to press (cr) again immediately after concluding the changed line with (cr), to return to command level. The line you specify with the C command tells AL to begin changing lines at that number. If you do not exit to command level after changing the desired line, you may inadvertantly change lines you did not mean to!

You can also specify a range of lines to change. The command shown below lets you change only lines 4 through 7.

        >   C 4,7(cr)

Note that when you specify a line to be changed, it is <u>automatically deleted</u> and the code you type in takes its place. In other words, the above example is equivalent to:

        >   DEL 4,7(cr)
        >   I 4(cr)

In general, it's advisable to use caution in using the C command to change large groups of lines. You'll probably find the DEL and I commands a more reliable means of altering ranges of lines.

MOD (Modify)

With the MOD command, you have a convenient means to perform minor editing on the lines in your source file. This function lets you search a line for any given character, then delete it or add characters following it. Type the MOD command followed by the line you want to change. When you press (cr), AL waits for you to enter your "search character." AL then prints the line up through the first occurrence of that character and stops to let you add, delete or change characters. You can even respecify the "search character" in mid-line to perform multiple changes in the line with a single pass.

Certain characters have special meaning within this command. These characters let you "move across" the line for editing. These characters and their meanings are:

∧ To respecify the search character during an editing pass.

< To find the next occurrence in the line of the specified search character.

> To append the remainder of the original line to the revised line.

(cr) To delete the remainder of the original line, leaving only the revised portion. (cr) is also used to execute the ∧ , > , and < special characters in the MODIFY function.

To illustrate the many ways these editing characters can be employed, let's work with a single, one-line example:

        1   ;THIS IS AN EXAMPLE

To modify the line so that it reads "THIS IS A SIMPLE EXAMPLE", you might use the MOD command as follows:

        >   MOD 1(cr)
        N
            THIS IS AN (bs) SIMPLE >(cr)
            EXAMPLE
        >   L 1(cr)
        1   ;THIS IS A SIMPLE EXAMPLE

In the process above, modification of line 1 is specified. The character 'N' is specified as the search character, and AL prints the line up through the first occurrence of 'N'. The (bs) deletes the 'N' from the line, then the word to be inserted, SIMPLE, is typed. The " > " symbol following SIMPLE appends the remainder of the original line (EXAMPLE) to the revised line.

When you type (cr), AL prints the remainder of the line and returns to command level.

To change the line to read "THIS IS SIMPLE" requires that you search for and delete the "A", respecify the search character to "E", then use the (cr) to delete the word EXAMPLE from the line.

        >   MOD 1(cr)
        A
        ;THIS IS A (bs)(bs) ∧ (cr)
        E
        SIMPLE(cr)
        >   L 1(cr)
        ;THIS IS SIMPLE

To add the word VERY between IS and SIMPLE in the line, you will have
to use the "✗" character to locate the first, then second, occurrence
of the letter "S".

```
>  MOD 1(cr)
S
;THIS <(cr)
IS VERY >(cr)
SIMPLE
>  L 1(cr)
1  ;THIS IS VERY SIMPLE
```

To add characters to the beginning of the line, you must search for the
first character in the line, delete it, add characters or spaces as
desired, then retype the letter you erased and append the rest of the
original line.

While it may seem easier just to retype the line (and, in some cases it
may well be!), we urge you to become familiar with using these editing
characters with the MOD command. Once you are comfortable with editing
in this way, you'll find that this command lets you make modifications
in your source code rapidly and easily.

The MOD command can also be used with several lines. The following command
allows you to modify lines 3 through 5.

```
>  MOD 3,8(cr)
```

REP (REPlace)

The REP command lets you replace a character string in your source code
with a different character string. The general format is the command word
REP, followed by the strings to be replaced and the replacement string,
separated with a comman and enclosed in quotes. You can also append line
numbers to the command to define where replacement is to occur. If no
line numbers are given, replacement occurs through the entire source file.
Consider the following source code:

```
1  VAR SET 5800H
2  LDA VAR
3  INR A
4  STA VAR
5  ;VARIABLE STORAGE
```

The label VAR occurs in lines 1,2, and 4. If we wanted to change the label
VAR to LAB, we would use the following command:

```
>  REP 'VAR','LAB'
>  L(cr)
1  LAB SET 5800H
2  LDA LAB
3  INR A
4  STA LAB
5  ;LABIABLE STORAGE
```

As you can see, every occurrence of VAR was replaced with the string LAB. Since we probably didn't want to change line 5, a better solution in this case would be:

> REP 'VAR','LAB',1,4

This tells us to replace VAR with LAB in lines 1 through 4 only.

SOURCE CODE ERROR

If you attempt to add in more source code than you have allotted for it, AL gives you the source code buffer full message:

    0?

STORING AND RETRIEVING YOUR SOURCE CODE

At various points during source code preparation, you will want to save the information you've entered on cassette tape. If the program you are writing is long, you may wish to save it in sections on separate tapes for easy revision if required after an assembly pass. You will also generally want to have a back-up copy of your source file on tape before beginning the assembly process, so that your hard work is not lost if something goes awry!

SAVING YOUR SOURCE CODE ON TAPE

To save your current source file on tape, use the R (Rewind) and SAVE commands.

> R   REWIND. This command turns the tape motor on so you can position your tape accurately for reading and writing. Use the F-FWD, READ, or REWIND cassette buttons to advance or rewind the tape. The sounds you hear when you use the READ button are not the sounds of a program loading, so AL and your source file are unaffected. You can also depress the READ and WRITE cassette buttons with this function to erase the tape.

> SAVE   When your data tape is properly positioned, depress the READ and WRITE cassette buttons. Then type SAVE(cr), and AL will write your source code to tape. The " > " prompt appears when the writing operation is completed.

RETRIEVING YOUR STORED SOURCE CODE

If you are continuing program entry from a previous session, or if *an* error has been discovered in the assembly process, you can reload a source file stored on tape earlier with the LOAD command.

Use the R (REWIND) command as described above to position the tape for reading. When it is properly positioned, depress the READ cassette button and type LOAD(cr). The " > " prompt reappears when the *reading operation* is completed. You can then edit or add to your file using the other AL commands.

Note that when you use LOAD, it add the source code you're loading onto
the end of any source code currently in AL. This can be useful when you
want to edit/assemble two files at once. However, if you only want to
work on one file at a time, use the NEW option before loading a tape.

PRODUCING PRINTED LISTINGS OF YOUR SOURCE CODE

AL has several commands that let you access a printer through your RS232
peripheral interface. These commands let you set the baud rate and line
length appropriate for your particular printer and produce listings of
your source code file.

SETTING THE BAUD RATE

To set the baud rate required by your printer, use the B (Baud) command.
to execute this command, type "B", followed by a four-digit, hexadecimal
number equivalent to the desired baud rate. The default AL baud rate is
1200 bps. For printers requiring a different baud rate setting, select
from the table below. If your printer operates at 600 bps, you would type
the folloiwng baud rate command:

> B 00AH(cr)

The baud rate has been set when the " > " prompt appears.

```
┌─────────────────────────────────────────────┐
│              BAUD RATE SETTINGS               │
│                    Bnnnn                      │
│                                               │
│        Baud Rate              nnnn =          │
│                                               │
│          110                  03F6            │
│          134.5                033D            │
│          150                  02E8            │
│          300                  0174            │
│          600                  00BA            │
│          1200                 005D            │
│          2400                 002F            │
│          4800                 0017            │
│          9600                 000C            │
└─────────────────────────────────────────────┘
```

LISTING THE SOURCE FILE

The parameters for producing a source file listing are identical to the LIST command except that the command used is LL.

The following command lists your entire program:

> `LL(cr)`

To list a range of lines, specify the lines that begin and end the range when you invoke the LL command.

> `LL 15,40(cr)`

To list a single line, specify the line number with LL:

> `LL 23(cr)`

AL THE ASSEMBLER - GENERATING THE OBJECT CODE

When your source code has been prepared, you are ready to begin the assembly process. Remember that your source file should begin with an ORG mnemonic to tell AL where to generate the object code.

GENERATING THE SYMBOL TABLE

AL has a multi-pass assembler with forward referencing capabilities. This means that you can reference labels in the source code before the labels have been defined. Because AL has this feature, it must perform a preliminary pass on the source file to determine which labels are used and to assign addresses to the labels. In this pass, called Pass 1, AL builds the symbol table. This is the first step in the assembly process; it must be executed before any subsequent passes can be done.

Pass 2 actually creates the object code in RAM.

The command A (Assemble) executes pass 1 and pass 2 in sequence. If any errors are found in pass 1, pass 2 cannot be executed.

You can also execute each pass separately. The command A1 initializes and builds the symbol only, and A2 creates the object code. When executing the passes separately, be sure to do A1 first, since the object code is dependent on a complete symbol table.

There is an additional assembly pass which can be used to add new labels to an existing symbol table. This pass is done with the command AO. It is useful for building a symbol table from more than one source file without destroying the references created by the initial A1.

ERRORS IN ASSEMBLY

The assembly pass will abort on the first occurrence of an error in the source code. You must correct the error before you can proceed with assembly of the file.

All error messages are given in the format:

        E: nn LLL

where:

        LLL is the line number in which the error was found.

        nn is the error type. The error numbers and their meanings are
            given on the following page.

ERROR CODE	MEANING
00	uneven or missing quotation marks
01	extra term on line
02	symbol table full
03	redefined label
04	missing or invalid mnemonic
05	number error
06	missing register
07	invalid operator

DISPLAYING THE SYMBOL TABLE

You may display the contents of the symbol table on the screen using the SD command, or dump it to the printer using the command LSD.

SAVING AND RETRIEVING OBJECT CODE

Once your program is correctly assembled, you will want to save it on tape. Use the "R" key to turn the tape motor on. Position the tape and press (cr) to turn the motor off. Depress the READ and WRITE cassette buttons and use the following format to save your code.

W addr1,addr2

Where addr1 is the starting address of your object code and addr2 is the ending address.

The example below saves the object code starting at address 5000H to 528FH.

> `W 5000H,528FH`

Note that the code below will also save the same area of memory since 20,480 equals 5000H.

> `W 20480,528FH`

The first parameter may also be a label name if properly defined in the symbol table.

> `W LABEL,528FH`

Reload object code tapes with the READ command. Note that if your object code uses some area of RAM also occupied by AL, AL will be destroyed and must be reloaded.

AL THE MONITOR - EXAMINING AND PATCHING OBJECT CODE

AL also has several functions normally found in a monitor program. These functions allow you to manipulate the object directly, which helps in the testing phase of your program.

DISPLAYING MEMORY

The D (Display) command lets you examine actual memory locations. It outputs their addresses and values in hexadecimal. Type 'D', then the starting address that you want to begin displaying.

To display part of the contents of ROM, enter the command:

> `D 200H(cr)`

This produces the following display:

```
0200    54    41    50    45
0204    00    FF    03    3A
0208    14    52    20    54
020C    4F    20    52    45
0210    53    54    41    52
```

You can halt the display temporarily by hitting any key but (cr), then restart with any key but (cr). (cr) terminates the display.

The first column of numbers shown are the hexadecimal addresses of successive locations, in increments of four. The four columns following the hex address display the contents of the four successive locations, beginning at the address in the first column.

You can also use a label name instead of an address on the Display command.

You can also display the contents of the various registers with the DR command.

To dump memory to the printer instead of the screen, use the LD command.

CHANGING MEMORY LOCATIONS

The S command displays memory, one location at a time, in a forward or backward direction. To use the command, type S followed by the starting address you wish to change. Using S, you can enter a new value or leave the current value unchanged as follows:

- To leave the value unchanged and display the next higher location, type a carriage return (cr).

- To leave the current value unchanged and display the previous location, press the Backspace (bs) key.

o   To change a value and then display the next higher location, type
    in the new hexadecimal value and press (cr).

o   To return to AL's command level, type a period (.), then press (cr).

Numbers may be entered in any of the acceptable numeric formats, including
characters or strings in single quotes.

SETTING A WINDOW

You may change the screen display format using the WIN (WINdow) command.
Type WIN, followed by the number of lines to be allocated for scrolling
on the bottom portion of the screen. No scrolling will occur above the
point specified with the WIN command, even with subsequent AL commands.
The WIN command lets you develop graphic screen displays with AL's Monitor
more easily. The window is reset to its default parameters when RESET
is pressed.

RUNNING YOUR OBJECT CODE

The G (Go) command executes your object code beginning at the address
you specify. The example below starts running the program at 5210H.

>   `G 5210H(cr)`

A label may be used in place of the address.

When the program is interrupted by a break point, the contents of the
registers may be examined using the DR command. This feature tells you
the current status of each register, individual flag settings, the program
counter, and stack pointer.

To continue program execution after a break, just type CONT and program
execution will pick up where it left off, at the next instruction following
BRK.

Any of the individual registers, A,F,B,C,D,E,H, or L or the register pairs
BC, DE, or HL can be assigned a value using the SET command.

Example:

>   `SET D,'A'(cr)`

This assigns the value 41H (ASCII for "A") to register D.

The assigned value can also be a label name if the symbol table is intact.
For instance, the following is legal as long as the label DATA exists
in the symbol table:

>   `SET HL,DATA(cr)`

-21-

SAMPLE PROGRAM

```
          ORG 4COOH                    EXAM1:    CALL PUTC
RCHRAD SET 5FE1H                                 ;OUTPUT CHAR
SETSCR SET 575H                                  LXI, B,OCFFH
CHRCOL SET 62EH                                  ;SET DELAY VALUE
PUTC SET 43EH                                    CALL DELAY
DELAY SET 7F6H                                   MVI C,3
PUTSCR SET 54FH                                  CALL CHRCOL
CLS SET 573H                                     ;SET UP ERASE COLOR
                                                 MVI C,1
INT:      LXI SP,5FCOH                           ;POINT TO FIRST CHAR
          ;SET STACK POINTER                     CALL PUTC
          JMP BEGIN                              ;ERASE CHAR
          ORG 5OOOH                              MOVE A,D
                                                 ;MOV X COORD TO ACC
BEGIN:    CALL CLS                               ADI 2
          ;CLEAR SCREEN                          ;ADVANCE X COORD
          LXI B,OFFFFH                           CPI 112
          ;SET DELAY VALUE                       ;END OF SCREEN?
          CALL DELAY                             RP
          MVI C,3                                ;IF SO,EXIT
          CALL SETSCR                            MOV D,A
          ;SET SCREEN COLOR                      ;PUT BACK X COORD
          MVI C,1                                MVI C,2
          CALL CHRCOL                            CALL CHRCOL
          ;SET STRING COLOR                      ;RESET COLOR
          LXI D,819H                             MVI C,1
          ;X,4 SCREEN COORDINATES                ;POINT TO 1ST CHAR
          LXI B,STRTAB                           JMP EXAM1
          ;ASCII TABLE                           ;DO IT AGAIN
          CALL PUTSTR
          ;DISPLAY STRING              CHRTAB:   DB 7,7
          CALL ANIM                              ;CHAR HGT/WDTH IN BITS
          ;MOVE CHARACTER                        DB 1OH,18H
          JMP BEGIN                              DB OFCH,OFEH,OFCH
          ;REPEAT                                DB 18H,1OH
                                      STRTAB:    DB 2OH,2OH,2OH
ANIM:     LXI H,CHRTAB                           DB 'DEMONSTRATION'
          ;GET ADDR OF USER CHRTAB               DB O
          SHLD RCHRAD
          MVI C,2
          CALL CHRCOL
          ;SET CHAR COLOR
          MVI C,1
          ;POINT TO FIRST CHAR
          MVI D,O
          MVI E,O
          ;X,Y SCREEN COORDS
```

RS232 BASIC -- LINEPRINTER COMMANDS
=====================================

RS232 BASIC has the same features and language capabilities as standard
Microsoft Level II BASIC. In addition, it contains two commands necessary
to access a lineprinter directly from BASIC or from a program:
LLIST and LPRINT.

LLIST

The LLIST command produces a lineprinter listing of the BASIC program
currently in memory. LLIST can be used in direct mode only. To produce
a listing, type LLIST (the command will appear on the TV screen). The
program will list out to the lineprinter.

LPRINT

The LPRINT command outputs information to a lineprinter. It can be used
in direct mode, but is more commonly embedded in a program to produce
reports, analyses, summaries, etc. In this case, the LPRINT command is
not executed until the program is run. The LPRINT command has the general
form:

                    LPRINT "string"
                        or
                    LPRINT variable

Examples of the use of these two commands are shown on the following page.
The program listing and resultant output were produced using an RS232-
equipped Interact and a COMPRINT lineprinter.

Combined BASIC and Machine Code Programming -- the USR Function

The MICRO VIDEO MONITOR gives you the facility to write programs with better
animation that execute faster than programs written in BASIC. However, you
may not want to write an entire program in machine code but still want the
improved graphics and speed the MONITOR offers. You can combine BASIC and
machine code by entering the machine code with the MONITOR and accessing
it from BASIC with the USR function.

Before entering the USR function, you must define the location (LSB and MSB)
of the machine code in memory, using two POKE statements. The value put into
the first POKE location defines the LSB (starting location of the machine code
routine); the value put into the second POKE location defines the MSB. To
determine the value to be entered in each POKE statement, convert the starting
and ending bytes of the routine from hexadecimal to decimal.

                    POKE 30499,L
                    POKE 30500,M
                    USR

RS232 BASIC -- Sample Program

To produce a program listing, type LLIST (command appears on TV screen):

```
10 LPRINT"      TEST PROGRAM"
20 CLEAR(200)
30 A$=""
40 FORCH=32TO95
50 A$=A$+CHR$(CH)
60 LPRINTA$
70 NEXT
```

Then, execute the program by typing RUN (also appears on TV screen):

```
        TEST PROGRAM
!
!"
!"#
!"#$
!"#$%
!"#$%&
!"#$%&'
!"#$%&'(
!"#$%&'()
!"#$%&'()*
!"#$%&'()*+
!"#$%&'()*+,
!"#$%&'()*+,-
!"#$%&'()*+,-.
!"#$%&'()*+,-./
!"#$%&'()*+,-./0
!"#$%&'()*+,-./01
!"#$%&'()*+,-./012
!"#$%&'()*+,-./0123
!"#$%&'()*+,-./01234
!"#$%&'()*+,-./012345
!"#$%&'()*+,-./0123456
!"#$%&'()*+,-./01234567
!"#$%&'()*+,-./012345678
!"#$%&'()*+,-./0123456789
!"#$%&'()*+,-./0123456789:
!"#$%&'()*+,-./0123456789:;
!"#$%&'()*+,-./0123456789:;<
!"#$%&'()*+,-./0123456789:;<=
!"#$%&'()*+,-./0123456789:;<=>
!"#$%&'()*+,-./0123456789:;<=>?
!"#$%&'()*+,-./0123456789:;<=>?@
!"#$%&'()*+,-./0123456789:;<=>?@A
!"#$%&'()*+,-./0123456789:;<=>?@AB
!"#$%&'()*+,-./0123456789:;<=>?@ABC
!"#$%&'()*+,-./0123456789:;<=>?@ABCD
!"#$%&'()*+,-./0123456789:;<=>?@ABCDE
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEF
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFG
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGH
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
```

.
.

MICRO *MV* VIDEO ™

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107
(313) 996-0626

RS232 BASIC -- I/O PARAMETER CONTROL
=======================================

BAUD RATE

Your RS232 interface can be used with Micro Video's RS232 BASIC to drive any
RS232-compatible lineprinter, at standard or non-standard baud rates between
110 and 19200 bps. The default baud rate initialized with RS232 BASIC is
1200 bps. This is the most commonly used baud rate, but there may be occasions
when you will need to select a different rate. To set the baud rate between
600 and 19200, use the following POKE initialization:

```
    POKE 25098,A            ; BAUD LATCH AT .C000H
```

where: A = 111860/desired baud rate

The table below identifies the value of A for standard baud rates.

A=	BAUD RATE
186	600
93	1200
47	2400
23	4800
12	9600
6	19200

To set a baud rate lower than 600 bps, a two-byte divisor, initialized by
two separate POKE statements, must be used.

```
    POKE 25098,A
    POKE 25099,B
```

The table below defines the values of A and B for standard baud rates lower
than 600 bps.

A=	B=	BAUD RATE
116	1	300
232	2	150
246	3	110

Note: Some devices do not operate at exactly standard (600, 1200, etc.) baud
       rates. If you have difficulty using a device with your Interact and
       RS232 BASIC, it is far more likely that the device's required baud rate
       is slightly off standard than that there is a problem with your computer
       or interface. Try decreasing or increasing the baud rate setting slightly
       for correct operation of the peripheral.

DATA FORMAT

The default data format setting in RS232 BASIC is 8-bit word length with 1 Stop bit, odd parity. Although this is a standard format used by most printers, some devices require non-standard data format. To set the individual bits for a non-default format, select as appropriate from the following statements.

Start with:

A=∅

To change word length:

```
A=A OR ∅              ; FOR 5-BIT LENGTH
A=A OR 1              ; FOR 6-BIT LENGTH
A=A OR 2              ; FOR 7-BIT LENGTH
A=A OR 3              ; FOR 8-BIT LENGTH (DEFAULT)
```

To set number of Stop bits:

```
A=A AND 251           ; FOR 1 STOP BIT (DEFAULT)
A=A OR 4              ; FOR 2 STOP BITS
```

To set parity on or off:

```
A=A AND 247           ; FOR NO PARITY CHECKING
A=A OR 8              ; FOR PARITY CHECKING (DEFAULT)
```

To set even or odd parity:

```
A=A AND 239           ; FOR ODD PARITY IF PARITY SET ON (DEFAULT)
A=A AND 16            ; FOR EVEN PARITY IF PARITY SET ON
```

Then, to initialize the non-default format:

```
POKE 25100,A          ; LINE CONTROL REGISTER AT .C004H
```

You can reset to the default data format with the statement:

```
POKE 25100,11
```

This is, of course, not the most efficient way to define the format parameters. This explanation is intended to explain the construction of the value to be POKEd into location 25100. To determine the value of A for your particular device, you can enter these statements in direct mode and then use the statement

PRINT A

to obtain the value of A. Then, for future use, you can poke that value directly into location 25100.

LINE FEED CONTROL

You can also control automatic line feeding through RS232 BASIC. The line
feed flag is initialized at 0 (no line feed). To set automatic line feed
upon encounter of a carriage return, use the following POKE instruction:

POKE 25097,10

PRINT CHARACTER DELAY CONTROL

For successful operation of teletypes as printers and some printers that
run at very high or low baud rates, you may need to adjust the speed at
which BASIC sends characters to the printer. Symptoms of this condition
may include characters being dropped out of lines or lines not printing
at all. To delay printing of characters for your particular device, use
the following POKE instruction and a value within the range indicated below:

POKE 19279,4 (default)

POKE 19279,0 (4800 bps)

POKE 19279,18 (110 bps--TTY*)

You may have to experiment with the POKE value, adjusting it up or down
to meet the requirements of your printer.

NOTE: The I/O control parameters for non-default printer requirements must
        be initialized before you can access the printer. You can enter
        the required POKE instructions in direct mode, or you can set them
        up in a small initializing program. The RS232 port is not initialized
        after you have entered the commands or run the initializing routine
        until BASIC responds "OK".

* To use a Model 33 Teletype as a printer with your Interact will require
that the RS232 board be modified for 20Ω access, as shown in the wiring
diagram on the following page.

20 MA Loop Modification for Use of Model 33 Teletype as a Printer

RS232 Board
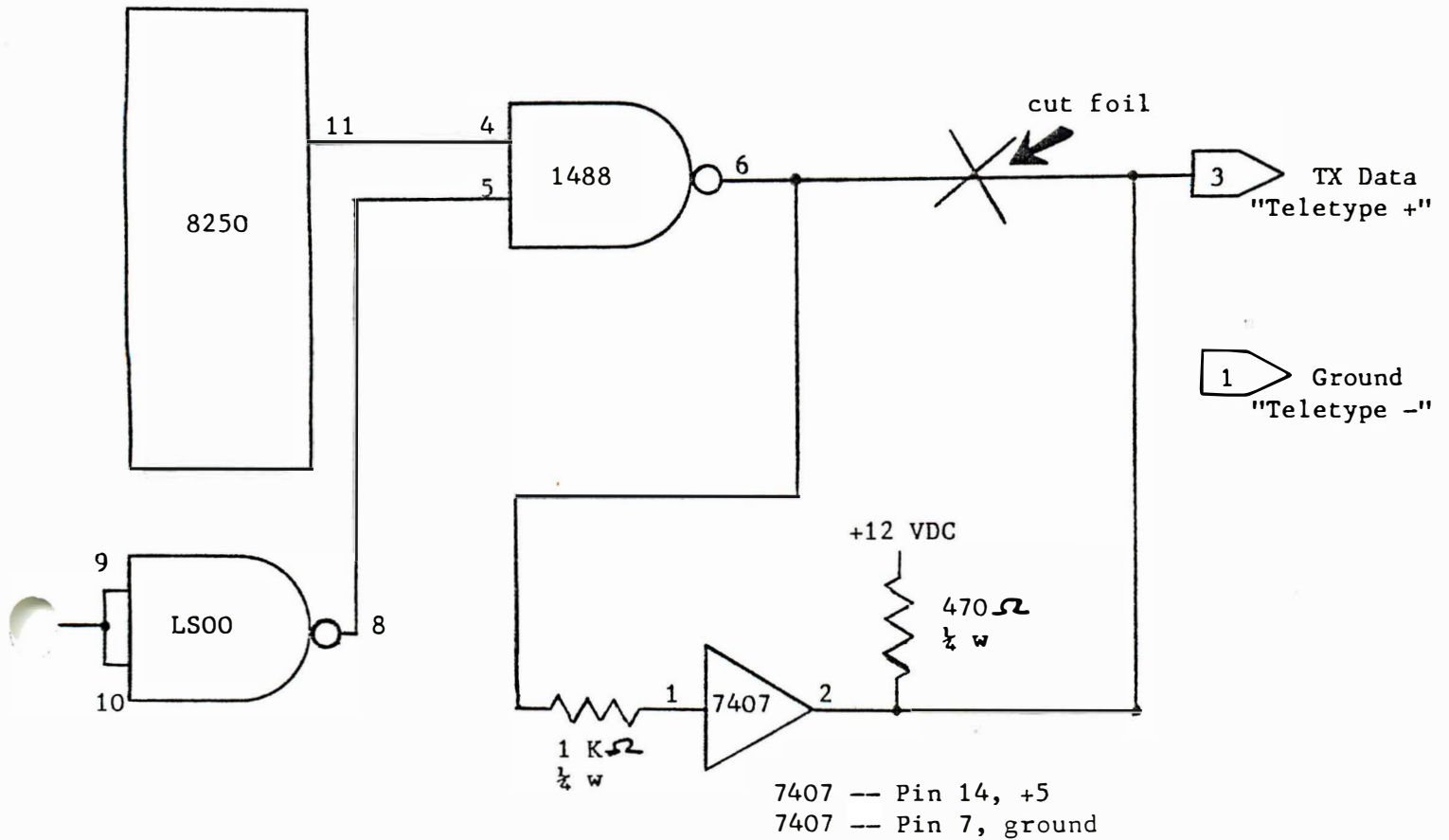
+12 VDC available at pin 28 of the 8080A chip or pin 14 of the 1488 chip.

DISCLAIMER: This modification has been successfully made to Interact computers for teletype access. Please do not attempt to install this modification without proper electronics training and skills. Micro Video assumes no responsibility for damage to the RS232 board or Interact computer due to user error in making this or other modifications.

MICRO ⚡ VIDEO ™

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107
(313) 996-0626

RS232 PORT -- MONITOR ACCESS
=============================

(for advanced programmers)

Your RS232-equipped Interact can also drive a printer or modem via the
MONITOR. By substituting values into various memory locations, you can
access and control any or all registers to transmit and receive data. Use
the SUBSTITUTE command in the MONITOR to set values of register memory
locations as necessary to set baud rate, parity, word length.

The table on the next page summarizes the accessible 8250 registers. The
pages following the table give additional information about using the table
data.

More detailed information about the 8250 accessible registers can be obtained
by ordering the following document from National Semiconductor Corporation:

        INS8250 Asynchronous Communications Element
        Publication # 426305492-001A

        National Semiconductor Corporation
        2900 Semiconductor Drive
        Santa Clara, CA 95051

8250 ACCESSIBLE REGISTERS

Register Address / Bit No.	C000 DLAB=0 Receiver Buffer Register (Read Only)	C000 DLAB=0 Transmitter Holding Register (Write Only)	C003 Line Control Register	C004 MODEM Control Register	C005 Line Status Register	C006 MODEM Status Register	C000 DLAB=1 Divisor Latch (LS)	C001 DLAB=1 Divisor Latch (MS)
0	Data Bit 0*	Data Bit 0	Word Length Select Bit 0	Data Terminal Ready	Data Ready	Delta Clear to Send	Bit 0	Bit 8
1	Data Bit 1	Data Bit 1	Word Length Select Bit 1	Request to Send	Overrun Error	Delta Data Set Ready	Bit 1	Bit 9
2	Data Bit 2	Data Bit 2	Number of Stop Bits	Out 1	Parity Error	Trailing Edge Ring Indicator	Bit 2	Bit 10
3	Data Bit 3	Data Bit 3	Parity Enable	Out 2	Framing Error	Delta Receive Line Signal Detect	Bit 3	Bit 11
4	Data Bit 4	Data Bit 4	Parity Select	Loop	Break Interrupt	Clear to Send	Bit 4	Bit 12
5	Data Bit 5	Data Bit 5	Stick Parity	0	Transmitter Holding Register Empty	Data Set Ready	Bit 5	Bit 13
6	Data Bit 6	Data Bit 6	Set Break	0	Transmitter Shift Register Empty	Ring Indicator	Bit 6	Bit 14
7	Data Bit 7	Data Bit 7	Divisor Latch Access Bit (DLAB)	0	0	Received Line Signal Detect	Bit 7	Bit 15

* Bit 0 is the least significant bit and is the first bit serially transmitted or received.

LINE CONTROL REGISTER (C003)

The Line Control Register byte specifies the format used for transmitting data
to a printer and transmitting and receiving via a modem.

WORD LENGTH -- Bits 0 and 1

Bits 0 and 1 of this register specify the word length to be used. Encode these
bits to set word length as follows:

LENGTH	BIT 1	BIT 0
5 bits	0	0
6 bits	0	1
7 bits	1	0
8 bits	1	1

STOP BITS -- Bit 2

Bit 2 specifies the number of Stop bits generated or received for each serial
character transmitted. Set this bit according to the selected word length as
follows:

WORD LENGTH	SET	STOP BITS
5,6,7,8 bits	0	1
5 bits	1	1-1/2*
6,7,8 bits	1	2

* used very infrequently, only for specialized applications

PARITY -- Bits 3, 4, 5

Bits 3, 4, and 5 enable and define the parity to be used.

Bit 3 enables parity generation and checking between last word bit and Stop
     bit of serial data. This bit produces an even or odd number of 1's
     when data word bits and the parity bit are summed.

> 0 = parity off
> 1 = parity on

Bit 4 sets even or odd parity when parity is enabled with bit 3. It transmits
     and checks for either an even or odd number of 1's in the data word and
     parity bits.

> 0 = odd parity
> 1 = even parity

Bit 5 is the Stick Parity bit. When parity is enabled with bit 3 and bit 5
     is set to 1, the parity bit set by bit 4 is transmitted and received
     as the opposite of the actual parity setting. Bit 5 is virtually never
     used.

BREAK CONTROL -- Bit 6

This bit sets a break control on serial output and is used for communication
with a timesharing system. When bit 6 is set to 1, the serial output is
placed "on hold", regardless of other transmitter activity, until bit 6 is
reset to 0.

DLAB -- Bit 7

Bit 7 is the Divisor Latch Access Bit (DLAB). It must be set to 1 in order
to access the Baud Rate Generator for defining the baud rate for transmitting
and receiving data.

Bit 7	Access to
1	Divisor latches of Baud Rate Generator
0	Receiver Buffer Transmitter Holding Register

BAUD RATE CONTROL

With the MONITOR, the baud rate to be used is set during the initialize routine,
as illustrated in the sample program on the following page. The divisor latches
(DLAB) must be loaded during initialization to insure desired baud rate. To
load DLAB, bit 7 must be set to 1 (see Data Format). Reset bit 7 to 0 to send
or receive characters. The table below gives the values to be loaded for the
various standard baud rates.

BAUD RATES USING 1.11860 MHz CLOCK

BAUD RATE	INITIALIZATION VALUE
110	1014 03F6
150	744 02E8
300	372 0174
600	186 BA
1200	93 5D
1800	62 3E
2000	56 38
2400	47 2F
3600	31 1F
4800	23 17
7200	16 10
9600	12 0C
19200	6 06

SAMPLE INITIALIZATION AND SEND/RECEIVE ROUTINES
==

MONITOR Code	Assembly Routine	Remarks

```
                            ORG     4D00H
         ;
         ;              ROUTINE TO INITIALIZE PORT
         ;

4D00    3A 03C0    INIT:    LDA     0C003H       ; DO DIVISOR LATCH ENABLE
4D03    F6 80               ORI     80H          ; SET BIT
4D05    32 03C0             STA     0C003H       ; PUT TO HARDWARE
4D08    3E 5D               MVI     A,93         ; CLOCK/93=1200 BAUD
4D0A    32 00C0             STA     0C000H       ; DIVISOR LATCH LSB
4D0D    AF                  XRA     A            ; CLEAR A
4D0E    32 01C0             STA     0C001H       ; DIVISOR LATCH MSB
4D11    32 04C0             STA     0C004H       ; 0 INTO .C004H FOR PORT 0
                                                 ; 40H FOR PORT 1
4D14    3A 03C0             LDA     0C003H       ; DISABLE DIVISOR & ENABLE XMIT
4D17    E6 7F   F6 03       ANI     7FH    '2I  03H
4D19    32 03C0             STA     0C003H
4D1C    C9                  RET
         ;
         ;   LO:            ROUTINE TO SEND THE CHARACTER IN THE 'C' REGISTER
         ;                  TO THE PRINTER OR MODEM.
         ;                  ROUTINE DESTROYS THE 'A' REGISTER & FLAGS (PSW).
         ;
4D1D    3A 06C0    LO:      LDA     0C006H       ; LOOK FOR 'CTS'
4D20    E6 10               ANI     10H
4D22    CA 1D4D             JZ      LO           ; KEEP LOOPING TILL DEVICE READY
4D25    3A 05C0             LDA     0C005H       ; XMITTER REGISTER EMPTY?
4D28    E6 20               ANI     20H
4D2A    CA 1D4D             JZ      LO           ; KEEP LOOPING TILL EMPTY
4D2D    79                  MOV     A,C          ; GET CHARACTER INTO 'A'
4D2E    32 00C0             STA     0C000H       ; TORPEDO AWAY!!
4D30    C9                  RET
         ;
         ;   CI:            ROUTINE TO INPUT A CHARACTER FROM THE PORT.
         ;                  CHARACTER IS RETURNED IN THE 'A' REGISTER.
         ;                  THE 'A' REGISTER & FLAGS (PSW) ARE DESTROYED.
         ;
4D31    3A 05C0    CI:      LDA     0C005H       ; GET STATUS BYTE
4D34    E6 01               ANI     1
4D36    CA 314D             JZ      CI           ; LOOP TILL READY
4D39    3A 00C0             LDA     0C000H       ; GET CHARACTER
4D3C    C9                  RET                  ; & GO HOME!
```

RS232 PIN CONFIGURATIONS

Printer Configuration	Function	Modem Configuration	
1	Protective Ground	1	
3	Transmitted Data	2	
2	Received Data	3	
4	Request To Send	4	
5	Clear To Send	5	
6	Data Set Ready	6	
7	Signal Ground (Common Return)	7	
8	Received Line Signal Detector	8	
9	Reserved for Data Set Testing	9	
10	Reserved for Data Set Testing	10	
11	Unassigned	11	
12	Sec. Received Line Signal Detector	12	
13	Sec. Clear To Send	13	
14	Secondary Transmitted Data	14	
15	Transmission Signal Element Timing	15	(DCE Source)
16	Secondary Received Data	16	
17	Receiver Signal Element Timing	17	(DCE Source)
18	Unassigned	18	
19	Secondary Request To Send	19	
20	Data Terminal Ready	20	
21	Signal Quality Detector	21	
22	Ring Indicator	22	
23	Data Signal Rate Selector	23	(DTE/DCE Source)
24	Transmit Signal Element Timing	24	(DTE Source)
25	Unassigned	25	

Notes:

Your RS232 interface is wired for operation of a printer, as shown above.
Wiring is the same for a printer and modem, except that pins 2 and 3
are reversed. To use your interface with a modem, adapt a connector
cable in which pin connections 2 and 3 are switched.

Some printers may require a connector cable in which pins 4 and 5 and
pins 6 and 20 are also reversed. If necessary for successful operation
of your particular printer, adapt a cable in which these connections
are switched.

Not all the pins listed above are used in the Micro Video RS232 interface.
The eleven pins used are pins numbered 1, 2, 3, 4, 5, 6, 7, 8, 18, 20,
and 25. Pins 18 and 25 are reserved and used in your interface for the
secondary port.

MICRO ⚡ VIDEO

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107
(313) 996-0626

RS232 PERIPHERAL INTERFACE

Installation Instructions

TOOLS REQUIRED: Phillips screwdriver, epoxy (or other glue suitable
                 for plastics)

It is recommended that this hardware be installed in a static-free
environment and that you touch something metal to ground yourself before
handling the interface to avoid inadvertantly damaging any of the chips.

1. Read all instructions through carefully before proceeding with the
    installation.

2. Unplug the computer from the wall socket and TV switchbox. Place
    the computer upside down on a clean, preferably padded, surface, and
    remove the six screws that hold the top of the case to the bottom.
    Turn computer rightside up and gently raise the cover 2-3 inches.

3. Reach into the unit and unfasten the three sets of cables at the pin
    connectors on top of the metal box which encloses the processor.
    Make sure to note the positions of the cables for correct reassembly.
    Remove the top of the case completely.

4. Loosen the large metal plate on top of the processor housing by
    unscrewing the four screws holding it in place.

5. Unscrew the six small screws that hold the small metal connector panel
    on top of the large metal plate. Rotate this panel gently and pass it
    through the hole in the metal plate. The plate can then be completely
    removed from the housing, making the PC board easily accessible.

6. Remove the screws on the RS232 board connector cable to separate the
    grey plastic piece from the connector plug. Using epoxy or other
    glue suitable for plastic, glue the grey plastic piece to the outside
    of the case, lining it up with the existing cut-out in the side of the
    case. Allow glue to dry before reattaching connector plug.

7. Gently tilt the connector panel back to expose the 8080 chip. Use
    caution in moving the panel so that you do not break any wires or
    capacitor pins.

8. Remove the CPU (marked 8080) chip from its socket.

9. Plug 8080 chip into the empty socket on the right side of the RS232
    board (as shown below).
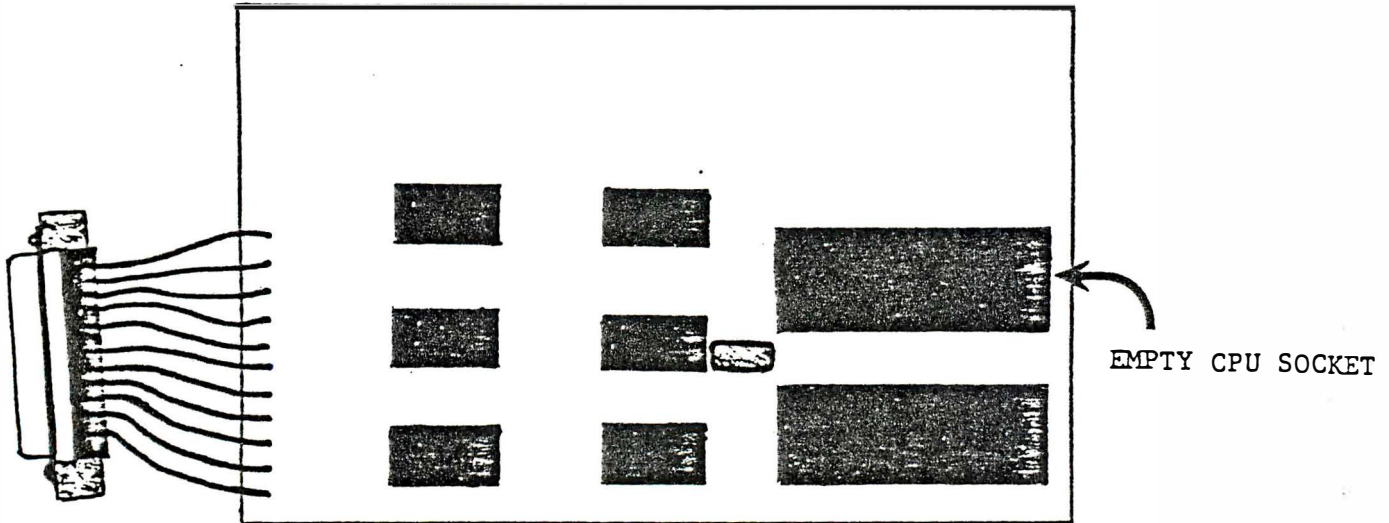
ROUGH LAYOUT OF RS232 BOARD

EMPTY CPU SOCKET

The board is now ready for installation into the computer.

10. Slide entire board in under the connector panel and line up the pins
     on the bottom of the board with the holes in the empty CPU socket.

11. When correctly aligned, press down firmly to seat the board in the
     socket.

12. When board is securely seated in the CPU socket, draw the connector
     cable through the hole in the left side of the electronics housing.
     Then, reattach the connector plug to the outside of the grey plastic
     piece you previously glued to the side of the case.

12. Reassemble your unit.

13. Enjoy the expanded capabilities your Interact now has!!

NOTE: To use your computer with a printer or modem, you will need a
       ribbon cable to attach to the device and the connector plug.
       These cables are readily available at any electronics supplier.

```
6A15 RET
6A16 PUSH PSW            (SPC 2)
6A17 PUSH B
6A18 MVI  C,20    (SPACE)
6A1A CALL 69EC    (SENDIT)
6A1D CALL 69E1    (DELAY)
6A20 CALL 69EC    (SENDIT)
6A23 CALL 69E1    (DELAY)
6A26 POP  B
6A27 POP  PSW
6A28 RET
6A29 PUSH PSW            (SPC 3)
6A2A PUSH B
6A2B MVI  C,20    (SPACE)
6A2D CALL 69EC    (SENDIT)
6A30 CALL 69E1    (DELAY)
6A33 CALL 69EC    (SENDIT)
6A36 CALL 69E1    (DELAY)
6A39 CALL 69EC    (SENDIT)
6A3C CALL 69E1    (DELAY)
6A3F POP  B
6A40 POP  PSW
6A41 RET
6A42 PUSH PSW            (SPC1)
6A43 PUSH B
6A44 MVI  C,20   (SPACE)
6A46 CALL 69EC   (SENDIT)
6A49 CALL 69E1   (DELAY)
6A4C POP  B
6A4D POP  PSW
6A4E RET
6A4F INR  B
6A50 INR  B
6A51 INR  B
6A52 INR  B
6A53 INR  B
6A54 INR  B
6A55 INR  B
6A56 INR  B
6A57 INR  B
6A58 INR  B
6A59 INR  B
6A5A INR  B
6A5B INR  B
6A5C INR  B
6A5D INR  B
6A5E INR  B
6A5F INR  B
6A60 INR  B
6A61 INR  B
6A62 INR  B
6A63 INR  B
6A64 INR  B
6A65 NOP
6A66 INR  B
6A67 NOP
6A68 INR  B
6A69 INR  B
6A6A INR  B
6A6B INR  B
6A6C INR  B
6A6D NOP
6A6E INR  B
6A6F NOP
6A70 INR  B
```

SEND 2 SPACES ?

SEND 3 SPACES ?

SEND 1 SPACE ?

```
6990 PUSH H              EXCHES [illegible] (CR/LF)
6991 PUSH  D
6992 PUSH  B
6993 PUSH  PSW
6994 CALL  69C1   ( INIT PORT )
6997 LXI   H,692C  LOAD HL              2C INTO H   69 INTO L
699A MOV   A,D      D -> A
699B SUB   M       SUBTRACT CONTENTS OF 692C FROM ACC  (17H) (23 DEC)     23+12 = 35 = ( ?
699C CPI   0C      COMPARE RESULT TO 0CH = 12 DEC                         23+13 = 41 = LRHT?
699E CZ    6A42  ( SPC1 )
69A1 CPI   12      COMPARE RESULT TO 12H = 13 DEC                         24+23 = 47 = (1)
69A3 CZ    6A16  ( SPC2 )
69A6 CPI   18      COMPARE RESULT TO 18H = 24 DEC
69A8 CZ    6A29  ( SPC3 )
69AB MOV   A,M
69AC SUB   D
69AD CNC   6A01    ( CR/LF )
69B0 CALL  69EC   ( SEND IT )
69B3 CALL  69E1   ( DELAY )
69B6 MOV   M,D
69B7 POP   PSW
69B8 POP   B
69B9 POP   D
69BA POP   H
69BB CALL  055B   ( OUT8A ) - ENTERS PUTSCR BUT NOT AT BEGINNING
69BE JMP   61D4                                          ( INIT )   ( INITIALIZE PORT )
69C1 PUSH  PSW
69C2 MVI   A,83    MOVE 83H INTO ACC      10000011      3 BIT WORD, NO STOP BIT, PARITY OFF,
69C4 STA   C003    PUT TO HARDWARE                      NO BREAK, SET DLAB TO 1 (SET DIVISOR LATCH ACCESS
69C7 LDA   692A    LOAD ACC WITH 17H      00010111      DIVISOR LATCH LSB 4800 BAUD ( CHANGED FROM INPUT
69CA STA   C000    DIVISOR LATCH LSB
69CD LDA   692B    LOAD ACC WITH 0H
69D0 STA   C001    DIVISOR LATCH MSB
69D3 XRA   A       ZERO THE ACCUMULATOR
69D4 STA   C004    0 INTO C004 - PORT 0
69D7 LDA   C003    BRING 83 BACK TO ACC
69DA ANI   7F      AND IT WITH 7F         10000011 - RESET DLAB TO ZERO   ENABLE XMIT
69DC STA   C003    ENABLE XMIT        AND  01111111
69DF POP   PSW                             11111111
69E0 RET           ALL DONE
69E1 PUSH  PSW                           ( DELAY ) ROUTINE
69E2 PUSH  B
69E3 LXI   B,02B0                  B0 (176) INTO B  02 (2) INTO C          689
69E6 CALL  07F6   ( DELAY )                                                13.4
69E9 POP   B                                                              2752
69EA POP   PSW                          2 x 256 = 512                     2064
69EB RET                                   176                            68 3      -> 9 NSEC ?
69EC LDA   C006    ( MODEM STATUS REGISTER ) - LOAD CONTENTS OF MODEM STATUS REGISTER
69EF ANI   10      COMPARE WITH 10H        - LOOK FOR CTS ( CLEAR TO SEND )
69F1 JZ    69EC
69F4 LDA   C005    ( LINE STATUS REGISTER ) - KEEP LOOPING UNTIL PRINTER READY
69F7 ANI   20                                 TRANSMIT REGISTER EMPTY?
69F9 JZ    69EC                            KEEP LOOPING UNTIL REGISTER EMPTY
69FC MOV   A,C     MOVE CHARACTER INTO ACC
69FD STA   C000    SEND IT OUT - STORE IT IN XMIT HOLDING REGISTER
6A00 RET
6A01 PUSH  PSW                          ( CR/LF )
6A02 PUSH  B
6A03 MVI   C,0D    ( CARRIAGE RETURN )
6A05 CALL  69EC   ( SEND IT )
6A08 CALL  69E1   ( DELAY )                         - SEND CR/LF
6A0B MVI   C,0A    ( LINE FEED )
6A0D CALL  69EC   ( SEND IT )
6A10 CALL  69E1   ( DELAY )
6A13 POP   B           6A14  POP PSW
```

```
6925 STC          37  55   7
6926 NOP          0   0
6927 NOP          0   0
6928 NOP          0   0
6929 NOP          0   0
692A RAL          17  23
692B NOP          0   0
692C RAL          17  23
692D NOP          0   0
692E NOP          0   0
692F NOP          0   0
6930 *            20               20
6931 MOV   D,D    52  82    R
6932 MOV   D,E    53  83    S
6933 DCR   L      20  45    -
6934 STA   3233   32  50    2  33  51  3  32  50  2
6937 *            20
6938 NOP          00
6939 MOV   B,D    42  66    B
693A MOV   B,C    41  65    A
693B MOV   D,L    55  85    U
693C MOV   B,H    44  68    D
693D *            20
693E MOV   D,D    52  82    R
693F MOV   B,C    41  65    A
6940 MOV   D,H    54  84    T
6941 MOV   B,L    45  69    E
6942 CMC          3F  63    ?
6943 *            ??  /
6944 NOP          00
6945 LXI   B,6930         DATA
6948 LXI   D,1A42         PUTSCR
694B CALL  054F
694E CALL  61AB
6951 LXI   D,0842
6954 LXI   B,6255
6957 CALL  054F    PUTSCR
695A RET
695B LXI   D,0C42
695E LXI   B,6939
6961 CALL  054F    PUTSCR
6964 CALL  61AB
6967 CALL  61DD
696A CPI   0D    — CARR RET RETURN IS  0D ???
696C RZ
696D PUSH  H
696E LXI   H,692B  (????)
6971 CALL  697F  (CALL 6144)
6974 MOV   M,A
6975 DCX   H
6976 CALL  697C  (CALL 61DD)
6979 MOV   M,A
697A POP   H
697B RET
697C CALL  61DD
697F CALL  6144
6982 RLC
6983 RLC
6984 RLC
6985 RLC
6986 MOV   B,A
6987 CALL  61DD
698A CALL  6144
698D ADD   B
698F RET       698F NOP
```

LOOKS LIKE PRINTE[R]
ROUT[
HE[

RS-232

6930 INITIAL MESSAGE ADDRESS (IN BC) (RS232?)
1A42 SCREEN LOCATION FOR UPPER CORNER OF FIRST CHAR
CALL PUTSCR — DISPLAY CHARACTER STRING

842 SCREEN LOCATION FOR UPPER CORNER
6255 INITIAL MESSAGE ADDRESS (DISASSEMBLER)
CALL PUTSCR

C42 SCREEN LOCATION FOR UPPER CORNER
6939 INITIAL MESSAGE ADDRESS (BAUD RATE?)
CALL PUTSCR

MICRO ⓜ VIDEO

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107
(313) 996-0626

AIRCRAFT LANDER

Program Instructions   (for 1 player)

AIRCRAFT LANDER is a BASIC program which simulates the landing of
an aircraft on a 60' by 3000' runway. The screen display realistically
presents the pilot's view during approach and landing and the plane's
instrument panel. The landing can be performed with AUTO PILOT mode,
which demonstrates a computer-controlled perfect landing, or in MANUAL
PILOT, in which you pilot the plane using the left entertainment controller.
The instrument display gives you compass headings, recommended headings,
and dive/climb changes. A post-landing evaluation is presented after
touchdown on the runway.

We recommend that you fly in AUTO PILOT mode for the first landing
so that you see the development of visuals, the layout of your instrument
panel, and the recommended values of flight variables during a successful
landing. Sound effects that simulate the engine noise vary with engine
speed.

LOADING THE PROGRAM: Insert the Level II BASIC tape or the Microsoft
8K Graphics BASIC tape into the cassette deck. Depress the RESET
button, press the "L" key, rewind the tape if necessary, then depress
the READ cassette button to load the BASIC interpreter into the computer.
When BASIC is loaded and the "OK" prompt appears on the screen, remove
the BASIC cassette, insert the AIRCRAFT LANDER cassette into the tape
drive. Depress the RESET button, press the "L" key, rewind the tape
if necessary, then depress the READ cassette button to load the program
into the computer under BASIC control. When the "OK" prompt reappears,
type RUN and press the "CR" key to execute the program.

GAME PLAY: The computer asks whether you want to fly using AUTO PILOT
(press the "1" key) or MANUAL PILOT (press "0"). Type the appropriate
value, then press the "CR" key to begin the simulation.

In AUTO PILOT mode you can sit back and enjoy the flight. Landing
is more difficult than it looks, so AUTO PILOT lets you watch the
process unfold. A landing takes several minutes to negotiate (just
like the real thing!), so don't be in too much of a hurry with this.
Pay particular attention to the instrument panel and adjustments that
are made during landing. When you're flying with MANUAL PILOT, you
may feel that things are moving too fast—especially if you get off
course!

MANUAL PILOT OPERATION

In MANUAL PILOT mode, you control the aircraft landing via the left entertainment controller. The controller functions are:

1. Potentiometer (pot) know controls the engine speed (RPM) and hence, the speed of the airplane.

2. The joystick lever simulates the "joystick" in an actual plane:

   a. Push the lever away from you (up) to cause the plane to dive.

   b. Pull the lever toward you (down) to cause the plane to climb.

   c. Move the lever to the left or right for course heading corrections.

   d. The diagonal lever positions are used to bank the plane, which causes the horizon to tilt and the picture perspective to change accordingly.

3. The fire button is used to obtain a report evaluating the landing, and to control restart of the landing simulator.

SCREEN DISPLAY

The screen display has three parts:

1. Blue sky,

2. Green field with runway outline and cross-hair to indicate center of vision, and

3. The cockpit instrument display.

SAMPLE PROGRAM SCREEN

```
                        BLUE SKY

                                    ┌─── RUNWAY OUTLINE
  GREEN FIELD

                                    ┌─── CENTER OF VISION


        0 + * * * +   ◄─────── GUIDANCE INDICATOR

        HEIGHT     RANGE
         150       4500

  MPH     RPM      V∧
  140     2500     720 V  ◄────── VERTICAL VELOCITY
                                  IN FEET/MIN
```
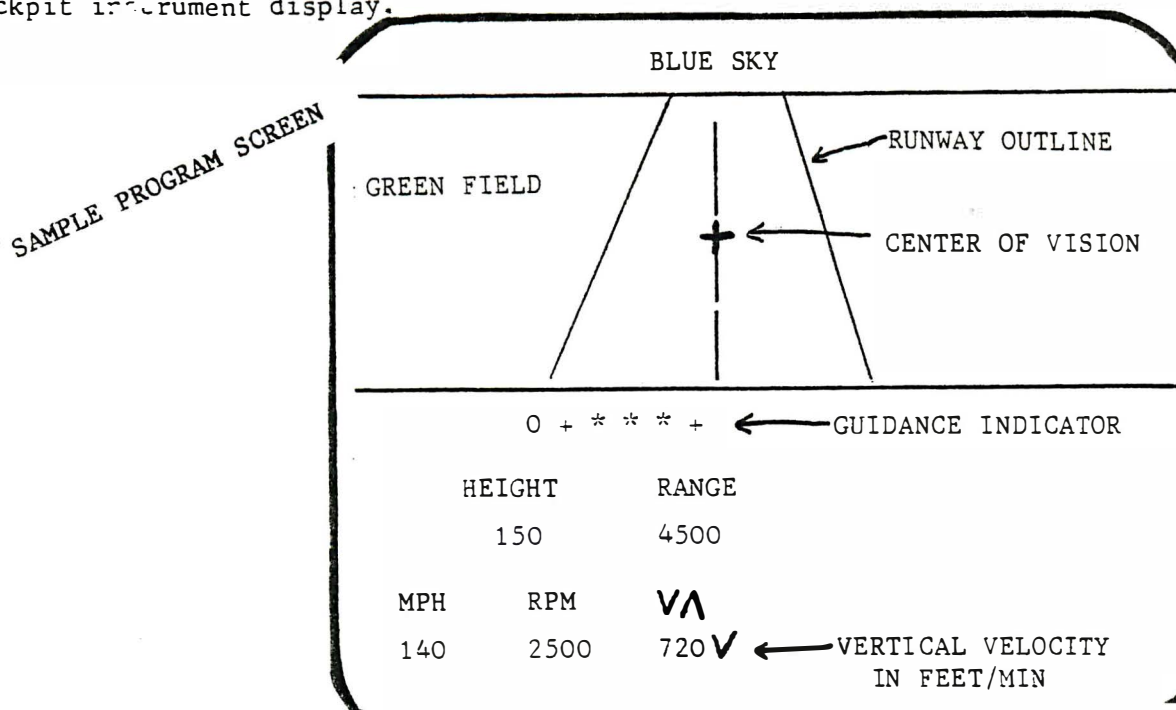
The Instrument Panel is relatively self-explanatory, except for the Guidance Indicator System (the top line of data on the "panel"). It consists of three parts:

1. A numeric compass heading, where zero (0) indicates "on course",

2. Left-right pointing arrows which indicate required heading changes,

3. Up-down pointing arrows which indicate required climb/dive changes.

The number of arrows (1, 2, or 3) indicate the relative amount of required change.

The Vertical Velocity Indicator in the bottom right corner of the panel indicates vertical velocity of your aircraft in feet per minute. The arrow symbol following the velocity value indicates direction, up or down.

THE SIMULATION: The simulation begins with your airplane 30,000 feet from the end of the runway. Values of speed, height, angle attack, and ground distance east or west from the runwayd are randomly determined. The program loop reads the settings on the entertainment controller (joystick lever and pot) and adjusts new values of plane position, speed, angle, and visual display. Other facts you should know include:

1. If speed drops to 100 mph, an audible tone warns that a stall is imminent. To prevent the stall, drop the aircraft nose by 15 degrees.

2. The runway is plotted as a white line. At distances greater than 10,000 feet, you'll see just the single line of the runway centerline. At closer distances you'll see the edges of the runway and the centerline.

3. When the aircraft is banked, the horizon line is displayed as a blue line on the green field.

4. Touchdown occurs when the height is less than 8 feet. Use the pot knob to decelerate and reduce speed to zero before running off the end of the runway. Landing is evaluated by both point of landing and impact force, with appropriate comments appearing on the screen.

IMPLEMENTATION: The program is written in BASIC and can be examined and/or modified. The computations, however, are mathematically quite sophisticated and shouldn't be modified without a good deal of prior study and examination.

Have a nice flight, and thank you for flying Micro Video!

MICRO ⚡ VIDEO

AIRCRAFT LANDER

Program Listing

Following is a complete listing of the AIRCRAFT LANDER program. This
listing was produced using the Micro Video RS232 interface, RS232 BASIC,
and a COMPRINT 912-S printer. The benefits of having listings of programs
as complicated and as large as AIRCRAFT LANDER are evident when you compare
this listing against the results of a LIST command on the TV screen.

You can obtain similar listings of your own programs by purchasing our
interface hardware and software. Alternatively, you may choose the Micro
Video Printer Service to obtain occasional listings of your programs
without having to invest in your own printer.

See our catalogues for information on both the RS232 interface and the
Printer Service (available exclusively from Micro Video).

```
100 CLEAR80:CLS:COLOR0,4,2,7:PRINTCHR$(8):A$=CHR$(1):I=30000:Z=4
102 PRINT"MANUAL PILOT = 0":PRINT"  AUTO PILOT = 1":PRINT:INPUT"0 OR 1";OC:CLS
104 OUTPUT"HEIGHT    RANGE",14,25,3:OUTPUT"MPH RPM   V↑",14,13,3
106 GOSUB110:H=2000+A:K=.01745:S=.001*A:GOSUB110:V=250+A:G=30*A
108 CT=1:M=150:IV=100:POKE19215,25:POKE19473,0:POKE13474,93:GOTO200
110 A=INT(100*RND(1)):IFRND(1)>.5THENA=-A
112 A$=A$+A$+A$+A$:RETURN
120 IFA1<33THENA3=A3+33*A4-A1*A4:A1=33
122 FORY=A1TOA2:A5=Y:IFA5>71THENRETURN
124 X=A3+A5*A4-A1*A4:IFX>0ANDX<112THENPLOTX,Y,3
126 NEXT:RETURN
200 REM"LOOP"
202 FORX=23808TO23818:READY:POKEX,Y:NEXT
204 DATA1,6,93,195,162,5,45,112,2,0,0:RESTORE
208 P=ATN(SQR(H)/6467):HO=INT(56-M*TAN(P+O)):IFI*I-H*H-G*G<0GOTO321
210 IFHO<33THENPOKE23816,1:L=USR(0):GOTO266
212 IFHO>75THENL=USR(0):GOTO230
214 IFT=0THENPOKE23814,HO-32:POKE23817,77-HO:L=USR(0)
216 IFT=0THENPOKE23814,77-HO:POKE23816,1:POKE23817,0:L=USR(0):GOTO230
222 L=USR(0):A1=HO+44*SIN(T):A2=56-44*CT:A3=56+44*CT:A4=TAN(T)
224 FORX=A2TOA3:Y=A1+(A2-X)*A4:IFY<76ANDY>32THENPLOTX,Y,1
226 NEXT
```

```
230 GI=SQR(I*I-H*H):N=SQR(GI*GI-G*G):B=ATN(H/GI):C=ATN(G/N)
232 PX=TAN(C+S):PY=-TAN(B+O)-PX*ST:PX=PX*CT:IFI<10000GOTO240
234 RX=ATN(56/M):IFG<0THENRX=-RX
236 RX=RX-S:IG=G/SIN(RX):BE=ATN(H/IG):EX=TAN(RX+S)
238 PE=-TAN(BE+O)-EX*ST:EX=EX*CT:SX=(PX-EX)/(PY-PE):GOTO260
240 GQ=G+50:QG=G-50:CL=ATN(QG/N):CR=ATN(GQ/N):PL=TAN(CL+S):PR=TAN(CR+S)
242 GL=SQR(N*N+QG*QG):BL=ATN(H/GL):YL=-TAN(BL+O)-PL*ST:PL=PL*CT
244 GR=SQR(N*N+GQ*GQ):BR=ATN(H/GR):YR=-TAN(BR+O)-PR*ST:PR=PR*CT:GOTO248
246 IV=IV-1:IFIV<1GOTO330
248 W=N-IV*30:IFW<30GOTO246
250 U=W*W:IG=SQR(U+G*G):LG=SQR(U+QG*QG):RG=SQR(U+GQ*GQ):BE=ATN(H/IG)
252 RX=ATN(G/W):RL=ATN(QG/W):RR=ATN(GQ/W):LB=ATN(H/LG):RB=ATN(H/RG)
254 EX=TAN(RX+S):PE=-TAN(BE+O)-EX*ST:EX=EX*CT:SX=(PX-EX)/(PY-PE)
256 EL=TAN(RL+S):LP=-TAN(LB+O)-EL*ST:EL=EL*CT:SL=(PL-EL)/(YL-LP)
258 ER=TAN(RR+S):RP=-TAN(RB+O)-ER*ST:ER=ER*CT:SR=(PR-ER)/(YR-RP)
260 A1=56+PE*M:A2=56+PY*M:A3=56+EX*M:A4=SX:GOSUB120:IFI>10000GOTO266
262 A1=56+LP*M:A2=56+YL*M:A3=56+EL*M:A4=SL:GOSUB120
264 A1=56+RP*M:A2=56+YR*M:A3=56+ER*M:A4=SR:GOSUB120
266 OUTPUTCHR$(43),54,58,0:B$=" + ":D$=" + ":BD=B/K+O/K:CD=C/K
268 IFI<10000THENZ=1:IFI>5000THENZ=2:BD=BE/K+O/K:CD=RX/K
270 C$="VVV":IFBD<1THENC$="V V":IFBD<.5THENC$=" V ":IFBD<.2THENC$="***"
272 IFBD<-.2THENC$=" ↑ ":IFBD<-.5THENC$="↑ ↑":IFBD<-1THENC$="↑↑↑"
274 IFABS(CD)<=.5GOTO280
276 IFCD>.5THENB$=" >":IFCD>1THENB$=" >>":IFCD>1.5THENB$=">>>"
278 IFCD<-.5THEND$="<  ":IFCD<-1THEND$="<< ":IFCD<-1.5THEND$="<<<"
280 U=-INT(S/K):SI=1:IFABS(U)=>2THENSI=2
282 IFU<0THENU=360+U:GOTO282
284 IFU=0THENY$="  0":GOTO288
286 Y$=STR$(U):IFU<100THENY$=" "+Y$:IFU<10THENY$=" "+Y$
288 FORY=7TO31STEP12:OUTPUTA$,8,Y,0:NEXT:X$="     ":IFCD<0THENX$=Y$:Y$=""
290 OUTPUTX$+B$+C$+D$+Y$,8,31,3:F=INT(V*SIN(O)):K$="V":IFF>0THENK$="↑"
291 A=INT(.682*V):IFA<100ANDH>8THENTONE100,500:O=O-.3
292 OUTPUTSTR$(H),8,19,3:OUTPUTSTR$(I),68,19,3:OUTPUTSTR$(A),8,7,3
293 R=INT(500+34*POT(0)):SOUND2*INT(3-R/3375),2*INT(7.5-R/750)
294 OUTPUTSTR$(R),32,7,3:OUTPUTSTR$(60*ABS(F))+K$,68,7,3:X=5000
295 V=V+Z*((10*R-V*V)/X-50*SIN(O)):IFOC=0GOTO300
296 O=-B-K:S=-C:IFI<2*XANDI>XTHENO=-BE:S=-RX
298 S=S-G/I:GOTO312
300 ONJOY(0)GOTO302,303,310,305,306,307,310,304,308,309
301 GOTO310
302 S=S+SI*K:GOTO310
303 S=S-SI*K:GOTO310
304 O=O+K:GOTO310
305 O=O-K:GOTO310
306 O=O-K:T=T+5*K:GOTO310
307 O=O-K:T=T-5*K:GOTO310
308 O=O+K:T=T+5*K:GOTO310
309 O=O+K:T=T-5*K
310 IFT<0THENT=T+K
311 IFT>0THENT=T-K
312 S=S+T/5:G=INT(G+Z*V*SIN(S)):IFABS(T)<KTHENT=0:ST=0:CT=1
313 IFT<>0THENST=SIN(T):CT=ABS(COS(T))
314 I=INT(I-Z*V*COS(O)*COS(S)):H=INT(H+Z*F):IFH>8GOTO200
320 IFO=0GOTO332
```

```
321 IFABS(G)>50THENCLS:PRINT"YOU LANDED OFF":PRINT"THE RUNWAY!":GOTO327
322 IFF<-20THENCLS:PRINT"YOU LANDED":PRINT"TOO HARD!":GOTO327
323 IFI>3000THENCLS:PRINT"YOU LANDED SHORT!":GOTO327
324 G$=STR$(ABS(G))+"FT":V$="AT"+STR$(A)+"MPH":I$=STR$(I)
325 F$="NICE LANDING!":IFF<-10THENF$="A BIT HEAVY!"
326 H$=STR$(F)+"FT/SEC":GOTO332
327 PRINT:PRINT"CRUNCH!!!!":COLOR0,0,0,0:FORX=1TO20:NEXT
328 COLOR7,7,7,7:FORX=1TO20:NEXT:COLOR0,4,2,7
329 FORX=1TO6:SOUND1,514:FORY=1TO10:NEXT:SOUND1,515:FORY=1TO10:NEXTY,X
330 GOTO346
332 H=7:O=0:T=0:OC=0:V=V-20:IFV<=0GOTO342
339 IFABS(G)>50THENCLS:PRINT"YOU RAN OFF SIDE":PRINT"OF RUNWAY":GOTO327
340 IFI<1THENCLS:PRINT"YOU RAN OFF END":PRINT"OF RUNWAY":GOTO327
341 GOTO200
342 FORY=7TO31STEP6:OUTPUTA$,8,Y,0:NEXT:OUTPUT"PRESS HIT BUTTON",8,25,3
243 OUTPUT"FOR COMMENTS",8,19,3:IFFIRE(0)GOTO342
344 CLS:PRINT"YOU LANDED";G$:PRINT"OFF CENTERLINE":PRINTV$:PRINTI$+"FT FROM END"

345 PRINT"VERTICAL SPEED":PRINTH$:PRINTF$
346 FORX=1TO5000:NEXT:CLS:SOUND7,4096.
347 PRINT:PRINT"PRESS HIT BUTTON":PRINT"TO PLAY AGAIN":IFFIRE(0)GOTO347
348 RUN
```

ARTILLERY COMMAND

A BASIC Game Program for 2 Players

You and your opponent have fixed artillery positions on either side of
an impassable mountain. The object of the game is to determine the firing
angle and muzzle velocity necessary to destroy the opponent's bunker.
With each shot you narrow down your range, but watch out, because your
opponent is firing back! Various options allow you to control the difficulty
of the game. It's an addicting strategy game for two players!

8K Fast Graphics

LOADING THE PROGRAM: The Artillery Command program requires Microsoft
8K BASIC. Insert the Microsoft 8K BASIC cassette into the cassette deck.
Depress the RESET button, press the "L" key, rewind the tape if necessary,
then depress the READ cassette button to load the BASIC interpreter into
the computer. When the BASIC "OK" prompt appears, remove the BASIC interpreter
cassette, insert the Artillery Command program cassette, depress the RESET
button, press the "L" key, rewind the tape if necessary, then depress the
READ cassette button to load the Artillery Command program into the computer.
When the BASIC "OK" prompt reappears, type RUN and press the "CR" key to
execute the program.

GAME OPTIONS: After you type RUN, the program asks you a series of question
which define the parameters of the game. First, it asks for the left and
right player names ("LEFT PLAYER?", "RIGHT PLAYER?"). Type in the left
and right player names as prompted, ending by pressing the "CR" key. The
computer then asks you how hard you want the game to be ("1=HARD, 2=EASY").
This difficulty option determines how accurate your shot must be to destroy
the opponent's bunker. The EASY option gives you a range of four pixels;
HARD narrows the target to a 2-pixel range. The computer then asks what
wind factor, if any, you want included. You must select a wind factor.
The options are as follows:

    1 - No wind
    2 - Wind is at a constant speed and direction and does
        not change until a player scores.
    3 - Wind changes randomly, but remains constant 80% of the
        time.
    4 - Wind can vary in speed and direction with every shot.
        This is the most difficult wind factor.

Wind speed and direction is shown in the upper left corner of the screen.
Wind speed is given in miles per hour. The wind direction is indicated
by the direction of the arrow. The wind factor can cause a shot to go
long or short, so remember to adjust your angle and velocity accordingly.

Copyright 1980, Micro Video Corporation

GAME PLAY: After all game options have been selected, two artillery bunkers
appear on the screen and a random mountainous terrain is drawn between
them. The computer randomly selects which player goes first and indicates
its selection by placing a white up-arrow ($\wedge$) beneath that player's bunker.
It then asks for firing information. It first asks for the firing angle
("ANGLE?"). Enter the angle in degrees by typing a number between 0 (horizontal)
and 90 (straight up) and pressing the "CR" key. The computer then asks
"VELOCITY?". Enter a number between 200 and 2000 to specify the muzzle
velocity in feet per second. After you enter the velocity, the shot is
fired and the trajectory is traced on the screen. If you fire too short,
your shot will hit the mountain and blast away the part where it hits.
If your shot is too long and goes past your opponent's bunker, a piece
of ground will be destroyed. If your angle and velocity cause the missile
to go far off the edge of the screen or if your shot is too weak, the missile
will explode harmlessly in mid-air. When one player hits the other player's
bunker, the bunker explodes, and a new round begins. The loser of the
previous round gets the first shot in the next round. After each round,
the score is displayed. The winner is the first person to destroy the
opponent's bunker five times.

NEW GAME: When the game ends, the computer asks if you want to play again.
Type "Y" to play another game, "N" to stop. If you type "Y", the computer
asks if you want to change any of the options. To play with the same difficulty
and wind options, type "N" to the "CHANGE?" query. To redefine any of
the options, type "Y" and answer the queries as in the beginning of the
game.

PROGRAM LISTING: A complete program listing for the Artillery Command
program is provided on the following pages. This listing was produced
using a computer equipped with the Micro Video RS232 interface, RS232 BASIC,
and a COMPRINT 912-S lineprinter.

With this listing, you can easily make changes to the program as desired.
A couple of suggestions for changes:

1) Z is the difficulty factor. It is one-half the number of pixels
   spread in the target area. It is now set for 2 or 4 pixels, depending
   on whether you select the HARD or EASY option in the game. Change
   Z to 1 to make the game more difficult; increase Z to make it easier.

2) If you select WIND option 3, there is a 20% chance that the wind
   speed and direction will change. To make the wind change more often
   with WIND-3, change the 80 in line 1480 to a number lower than 80.
   The percentage of change will equal the difference between the number
   you replace 80 with and 100.

```
1 REM*ARTILLERY        COMMAND
2 REM COPYRIGHT  1980 MICRO-VIDEO        CORPORATION
10 CLEAR:CLS:COLOR0,7,7,7:WINDOW36
20 OUTPUT"ARTILLERY",32,45,3
30 PRINT"      COMMAND":PRINT:PRINT:PRINT
40 FORX=1TO6:READA,B:TONEA,B:NEXT
50 DATA110,56,80,77,63,97,51,106,63,97,51,240
60 SOUND1,1570:COLOR1,7,7,7:FORX=1TO400:NEXT:SOUND7,4036
70 DIMA(100)
100 CLS:WINDOW77:COLOR6,2,1,7
140 PRINT"TWO GUNNERS WILL TAKE TURNS TRYINGTO HIT THE OTHER"
150 PRINT"FIRST TO SCORE 5 HITS WINS":FORX=1TO900:NEXTX:CLS:GOTO160
155 CLS:PRINT"  CHANGE-Y OR N":PRINT:PRINT:PRINT:N$=INSTR$(1):IFN$="N"GOTO225
160 PRINT:PRINT"LEFT PLAYER":INPUTA$
165 IFLEN(A$)=0THENA$="LEFT"
170 CLS:PRINT:PRINT"RIGHT PLAYER":INPUTB$
175 IFLEN(B$)=0THENB$="RIGHT"
180 CLS:PRINT"     HARD=1            EASY=2":PRINT:PRINT
190 INPUTZ:IFZ<1ORZ>2THEN180
195 Z=Z*2
200 CLS:PRINT" NONE-1    WIND-3 WIND-2    WIND-4":PRINT:PRINT
220 INPUTWW:IFWW>0ANDWW<5THEN225
221 GOTO200
225 A(0)=0:A(1)=0
230 WINDOW11:COLOR6,2,1,7:CLS
240 D=41*RND(1)+8:E=25*RND(1)+17
250 F=43*RND(1)+64:G=25*RND(1)+17
260 FORI=-2TO2:PLOT(D+I),E,3:PLOT(D+I),(E+1),3
270 PLOT(F+I),G,3:PLOT(F+ITAB(,(G+1),3:NEXTI
280 M=40*RND(1)+36
290 IFM<E THEN M=E+2
300 IFM<G THEN M=G+2
310 PLOT1,12,1,D+5,E-12
330 PLOTF-5,12,1,118-F,G-12
359 K=((M-E)/(51-D)):L=E-1
370 FORI=D+6TO56:L=L+K
380 PLOTI,12,1,1,L-11
390 NEXTI
400 K=((M-G)/(F-61)):L=M-1
419 FORI=56TOF-6:L=L-K
420 PLOTI,12,1,1,L-11
430 NEXTI
440 P=100*RND(1)+1:IFP<50THENP=0:GOTO460
450 P=1
460 IFWW><1THEN GOSUB1470
465 GOSUB2000
470 INPUT"ANGLE";A
480 IFA<0THENGOTO470
490 IFA>90THEN GOTO470
500 INPUT"VELOCITY";V
510 IFV<100GOTO500
520 IFV>2000GOTO500
524 IFWW=1THENGOTO530
525 OUTPUTY$,15,67,0:OUTPUTWS,20,67,0
530 V=V*.056
540 A=A*.0174533
550 W=A*A*A
```

```
                  ...  ...   ..A*A/120-...   ...
570 A=1.5708-A
580 W=A*A*A
590 C=A-W/6+W*A*A/120-W*W*A/5040
600 Q=V*C:R=V*S
610 K=D:L=E:IFP=1THENQ=-Q:K=F:L=G:WF=-WF
620 T=.05:B=0
630 T=T+.1:X=K+Q*T*(2.3+WF)
640 IF(X<0)OR(X>112)THEN680
650 Y=(R-16*T)*T+L:IFY>77THENTONE8,25:GOTO630
652 IFY<0THEN680
654 B=B+2:A(B)=X:A(B+1)=Y
660 IFPOINT(X,Y)THEN680
670 PLOTX,Y,2:TONE88-Y,35:GOTO630
680 IFP=0THEN710
690 IFABS(D-X)<ZTHEN760
695 FORTT=1TO10:SOUND1,1570:NEXTTT:SOUND7,4096
700 P=0:GOTO730
710 IFABS(F-X)<ZTHEN760
715 FORTT=1TO10:SOUND1,1570:NEXTTT:SOUND7,4096
720 P=1
730 FORI=2TOBSTEP2
740 PLOTA(I),A(I+1),0:NEXTI
750 GOTO460
760 A(P)=A(P)+1
770 IFP=0THENXX=F-4:YY=G+7
780 IFP=1THENXX=D-4:YY=E+7
790 GOSUB2500
820 IFA(P)>4THEN970
830 CLS:IFWW=.5THENWW=2
835 TONE131.96:TONE80,316
840 OUTPUTA$,10,67,3:OUTPUT"SCORE",10,57,3:OUTPUTA(0),40,57,3
850 OUTPUTB$,10,37,3:OUTPUT"SCORE",10,27,3:OUTPUTA(1),40,27,3
860 FORTT=1TO1500:NEXT:GOTO230
970 CLS:IFA(1)>4THENGOTO880
875 OUTPUTA$,20,57,3:OUTPUT"IS THE WINNER",10,37,3:GOTO885
880 OUTPUTB$,20,57,3:OUTPUT"IS THE WINNER",10,37,3
885 FORX=1TO300:SOUND0,24844:NEXT:SOUND7,4096:WINDOW77
890 INPUT"PLAY AGAIN-Y OR N";C$
900 IFC$="Y"THENCLS:GOTO155
999 END
1470 IFWW=.5THENGOTO1550
1475 IFWW=2THENWW=.5
1480 IFWW=0THENCF=100*RND(1)+1:IFCF<80THENGOTO1550
1490 IFWW=3THENWW=0
1500 DF=100*RND(1)+1:IFDF>50THENDF=1:GOTO1520
1510 DF=0
1520 WS=INT(10*RND(1)+1)
1530 IFDF=0THENY$="<":GOTO1550
1540 Y$=">"
1550 OUTPUTY$,15,67,3
1560 OUTPUTWS,20,67,3
1570 WF=WS/10:IFDF=0THENWF=-WF
1580 RETURN
2000 IFP=0THENOUTPUT"↑",D-2,E-1,3:OUTPUT"↑",F-2,G-1,1
2010 IFP=1THEN OUTPUT"↑",D-2,E-1,1:OUTPUT"↑",F-2,G-1,3
2020 RETURN
2500 SOUND1,1570
```

```
2505 EX=1
2510 COLOR7,2,1,7
2520 POKE24545,00:POKE24546,93
2525 GOSUB3000
2530 OUTPUTCHR$(1),XX,YY,2
2532 COLOR6,2,1,7:GOSUB3000:COLOR7,2,1,7
2535 GOSUB3000
2540 COLOR6,2,1,7
2550 OUTPUTCHR$(2),XX,YY,2
2555 GOSUB3000
2560 OUTPUTCHR$(3),XX,YY,2
2565 COLOR7,2,1,7
2570 OUTPUTCHR$(4),XX,YY,0
2575 GOSUB3000
2576 COLOR6,2,1,7
2580 OUTPUTCHR$(5),XX,YY,0
2585 GOSUB3000
2590 OUTPUTCHR$(6),XX,YY,0
2595 GOSUB3000
2596 IFEX<2THENEX=EX+1:GOTO2560
2600 OUTPUTCHR$(7),XX,YY,0
2602 GOSUB3000
2605 GOSUB3000
2610 SOUND7,4096:COLOR6,2,1,7
2615 FORX=1TO500:NEXT
2620 RETURN
3000 FORX=1TO100:NEXT:RETURN
```

MICRO ◪ VIDEO

ATTRO-LOGIC

Program Guide
============================

for 1 or 2 players

Only two placement rules govern the play of this simple logic game, but sharp strategy determines who comes out ahead.

LOADING THE PROGRAM: Insert the program cassette into the tape deck, depress the RESET button, depress the "L" key, rewind the tape if necessary, then depress the READ button to load the program into the computer.

GAME OBJECT: The object of ATTRO-LOGIC is to place pieces (symbols) on the screen playing board according to how their features (attributes) differ, so that you outscore your opponent. You score points for placing pieces horizontally (to the left or right of a square containing another piece) or vertically (above or below a square containing a piece). The symbol attributes are color and shape, size or direction.

GAME SELECTION: When the program is loaded, the computer asks for the number of players ("# PLAYERS? 1=ONE PLAYER, 2=TWO PLAYERS). Press the "1" key to play the computer and plug in the left joystick. Press the "2" key for two players and plug in both joysticks. The computer then asks you to enter the player name(s). Type your name(s) and press the CR key to continue. You are then asked to choose the type of symbols to be used in the game. The symbols used can differ in color and one other attribute-- shape, direction, or size. Type 1 to play with colored symbols with different shapes. Type 2 to play with colored arrows pointing different directions. Type 3 to play with colored symbols. of varied sizes.

GAME PLAY: When game selection is complete, the playing board and symbols appear on the screen, and the computer randomly selects which player goes first. Use the joystick lever to position the flashing cursor over the symbol you want to pick us. Press the hit button to pick up the piece. Then, move the cursor to the playing board using the joystick lever and position it over the square in which you want to place the piece. Press the hit button to place the symbol on the board and complete your turn. The first symbol may be placed anywhere on the board and scores no points. Place additional pieces on the board to score points according to the following rules:

    HORIZONTAL PLACEMENT -- To place a symbol in the square to the left or right of a square containing a symbol, the two symbols must differ in only one attribute--color OR shape, size, direction. You score 1 point for horizontal placement.

    VERTICAL PLACEMENT -- To place a symbol above or below a square containing a symbol, the two symbols must differ in both attributes. They must be

both a different color AND a different shape, size or direction. You
score 2 points for vertical placement.

To place a symbol in a square bounded by two or more squares containing
symbols, placement rules must be met for all bordering squares. If you
try to place a symbol that does not meet placement rules in a square,
a buzzer sounds, and you lose your turn.

The game ends when all pieces have been played or when no more pieces can
legally be played. The computer tallies the score and prints the winner.

NEW GAME: When the game is over, the computer asks "SAME GAME? Y=YES,
N=NO". Press the Y key or the joystick hit button to play the game again
using the same attributes. Press the N key to redefine the number of
players or the attribute used. The starting player alternates in
sequential games that are not redefined. If game selection is reset, the
computer once again randomly chooses the starting player.

BOMBS AWAY!

Program Instructions
=========================

LOADING THE PROGRAM: Insert the Level II BASIC program cassette into the
cassette deck. Depress the "L" key, rewind the tape if necessary, then
depress the READ cassette button to load the BASIC interpreter into the
computer. When the BASIC interpreter is loaded and the "OK" prompt appears,
depress the RESET button, insert the BOMBS AWAY! program cassette into
the tape deck, press the "L" key, rewind the tape if necessary, then
depress the READ cassette button to load the BOMBS AWAY! program into the
computer. Do NOT use the BASIC CLOAD command to load BOMBS AWAY! or the
machine language portion of the program will not be loaded. Plug in the
left entertainment controller. When the BASIC "OK" prompt reappears, type
RUN and press the "CR" key to execute the program.

GAME PLAY: In BOMBS AWAY! you are a fighter pilot trying to damage a
reservoir dam sufficiently to break through the dam wall and flood the
electric plant in the valley below. Your plane flies across the screen at
random altitudes determined by the computer. As the plane crosses the
screen, press the joystick hit button to release a bomb. You have eight
tries to knock a hole in the dam wall. If you succeed, the dam collapses,
and water rushes down into the valley, flooding and destroying the electric
plant. If you are not successful, your plane is shot down and demolished.

NEW GAME: At the end of each game, a new game starts automatically.

CSAVE "Bombs"
W 4A00, 4BFF
W 4C6A, 5FFE

BOMBS AWAY! Programming Tutorial

In addition to the entertainment value of the BOMBS AWAY! program, it is also a helpful tool for learning about effectively combining the MICRO VIDEO MONITOR and Level II BASIC to produce games or other programs with better, faster animation and execution.

We have provided a complete listing of the Level II BASIC portion of the BOMBS AWAY! program and a partial listing of the machine code portion. You can list the Level II code by typing the LIST command at the BASIC "OK" prompt. To list the machine code not included in the attached listing, load the MONITOR tape. Then, use the DISPLAY command to list the desired routine(s).

If you want to make changes to the program, you can use the EZEDIT program tape to change BASIC code and the MONITOR SUBSTITUTE command to alter the machine code.

From BASIC to Machine Code and Back Again

The BOMBS AWAY! program is driven from BASIC, but uses a number of machine code routines for graphic display and movement. The machine code routines are accessed by means of the BASIC USR function. Before invoking USR, you must define the location in memory of the routine to be called. This is done by specifying the starting address (LSB and MSB) of the desired routine with two POKE statements:

    POKE 19473, lsb

    POKE 19474, msb

Convert the starting address from hexadecimal to decimal to get the values of "lsb" and "msb" to be POKEd into locations 19473 and 19474. These POKE statements must be reentered for each separate routine called.

As you can see in the BASIC program listing included, our programmer chose to define J as 19473 and J+1 as 19474, to avoid having to retype these numbers each time the USR function is invoked and to conserve memory. He suggests that, in writing your own programs that use both BASIC and machine code, you try to keep all your machine language routines within the same 256-byte block so that the value of J+1 can remain constant.

To return to BASIC upon completion of a routine called with USR, put a C9 instruction at the end of that routine.

The following pages contain two listings. The first is a listing of the BASIC logic for the BOMBS AWAY! program. You can use the LIST command in BASIC to reproduce this information on your TV screen. The second page presents REMARKS for the program; each REMARK line corresponds to the BASIC code line with that same line number. These comments are included as explanation of the BASIC logic only and are not an actual part of the BASIC portion of the program. That is, you will not see these comments if you list the BASIC code with the LIST command in BASIC.

```
REM BOMBS AWAY
OK
LIST

 1 POKE19215,25:J=19473:M=22541:N=24545:T=M+11:U=M+194:E=23673:F=22664
 3 CLS:COLOR0,7,1,4:SOUND7,4096:POKEJ+1,88:POKEF+74,95
 5 POKEJ,32:A=USR(0):OUTPUT"/",59,18,2:OUTPUT"/",59,15,2:GOTO100
 7 SOUND2,60:FORA=1TO20:NEXT:SOUND7,4096
 8 G=PEEK(E+1)-245:Y=69-PEEK(E)
10 FORX=115TO0STEP-2
15 IFPEEK(U)=2THENPOKEJ,171:A=USR(0)
20 IFX+G<23THEN32
25 OUTPUTCHR$(1),X+G,Y,1:OUTPUTCHR$(1),X+G,Y,0
26 X=X-1:Y=Y-2
27 IFPOINT(X+G,Y)=1ORPOINT(X+G-2,Y)=1THEN42
28 IFPOINT(X+G-3,Y-1)=2THEN40
30 NEXT
32 IFY>40THENPOKEJ,160:A=USR(0):TONE999,5:RETURN
34 IFPOINT(X+G-1,Y)=3THENX=X+3:GOTO40
35 IFPOINT(X+G-2,Y)=3THENX=X+4:GOTO40
40 GOTO45
42 POKEJ,160:A=USR(0):TONE999,5:RETURN
45 POKEJ,76:A=USR(0)
46 POKEJ,160:A=USR(0)
48 POKEJ,0:V=77-Y
49 POKEM,V-2
50 POKEM+1,X+G-3
51 POKEM-1,0:POKEM-5,96:A=USR(0):POKEM-1,1:POKEM-5,112:A=USR(0)
52 FORA=1TO40:NEXT:POKEM-1,0:A=USR(0)
54 FORA=10TO40:IFPOINT(20,A)=0THEN58
55 NEXT:RETURN
58 POKEJ,144:POKEF+17,20:POKEF+20,0
60 FORB=39TO46:POKEF+19,B:A=USR(0):FORR=1TO40:NEXT:NEXT
61 POKEF+17,12:POKEF+20,19:FORB=34TO77:POKEF+19,B:A=USR(0):NEXT
62 POKEJ,16:SOUND1,2:FORB=19TO115STEP2:POKET+6,B:A=USR(0):NEXT:RUN
100 POKEN,72:POKEN+1,86:POKEJ,160:A=USR(0):W=0
101 IFPEEK(U)=2THENGOSUB7
102 IFPEEK(E+1)=237THENGOSUB110
103 POKEJ,171:A=USR(0):GOTO101
110 IFW=8THEN115
111 W=W+1:RETURN
115 SOUND3,340:POKEJ,84:A=USR(0):SOUND7,4096:POKEF+74,0
117 POKEJ,163:A=USR(0):POKEE,17
120 POKEJ,171:A=USR(0):IFPEEK(E+1)=177THEN125
121 FORR=1TO4:NEXT:GOTO120
125 SOUND2,60:POKEJ,168:A=USR(0):SOUND7,4096
130 POKEJ,80:COLOR0,0,0,0:FORA=1TO20:NEXT:COLOR7,7,7,7
131 FORA=1TO20:NEXT:COLOR0,7,1,4
132 FORB=1TO6:SOUND1,514:FORA=1TO10:NEXT:SOUND1,515:A=USR(0):NEXT
134 FORA=1TO1000:NEXT:RUN
OK
```

<u>REMARKS Listing for BOMBS AWAY! BASIC Code</u>

```
1 ENABLING POKE. SET MACHINE LANG. VARIABLES
3 CLEAR SCREEN. SET COLORS. SET MSB WITH J+1
5 PRODUCE WATER, DAM, PLANT GRAPHICS. GOTO MAIN LOOP.
7 MAKE BOMB SOUND. START BOMB ROUTINE.
8 SET(G,Y) BOMB LOCATION COORDINATES.
10 START MAIN BOMB LOOP.
15 CHECK IF PLANE OFF TOP OF SCREEN. IF SO, DON'T CALL PLANE.
20 CHECK LEFT BOMB LIMIT
25 OUTPUT BOMB WITH SPECIAL CHR$ CHARACTER.
26 SET BOMB ANGLE.
27 CHECK PROTECTIVE WHITE SHIELD OR TOP OF DAM.
28 CHECK FOR PROPOER HIT AREA.
32 CHECK IF LEFT LIMIT IS AIR. RESET PLANE.
34 CHECK IF BOMB REACHES WATER.
35 DITTO
42 RESET PLANE. EMIT MISS TONE.
45 HIT SOUND
46 RESET PLANE
48 (THROUGH 52) SETS EXPLOSION COORDINATES AND INVOKES IT.
54 GOTO 58 IF DAM IS BUSTED.
58 (THROUGH 61) DROPS WATER LEVEL AND WIPES OUT DAM.
62 PRODUCE WATER SOUND AND WAVE MOTION.
100 CREATE SPECIAL CHR$1) AND RESET PLANE.
101 CHECK FIRE BUTTON AND GOTO BOMB LOOP IF DEPRESSED.
102 IF PLANE AT RIGHT EDGE, INCREMENT NUMBER OF PASSES(W).
103 PRODUCE THE PLANE.
110 CHECK IF LAST (3'TH) PASS.
115 DOOR SOUND. OPEN DOOR, LIFT MISSLE, DISABLE HIT BUTTON.
117 RESETS PLANE WITHOUT CALLING RANDOM ALTITUDE.
120 FINAL PLANE FLYBY. STOP PLANE BEFORE HIT.
125 FIRE THE MISSLE.
130 (THROUGH 132) PRODUCE FINAL EXPLOSION AND FLASH 6 TIMES.
134 PAUSE, THEN START GAME OVER AGAIN.
OK
```

BOMBS AWAY! MONITOR Code

The following pages contain a partial "hex dump" of the machine code used in the BOMBS AWAY! program. To see the contents of locations not included in this listing, use the MONITOR DISPLAY command.

Following is a list of the major routines used in BOMBS AWAY! The starting address for each routine is given, followed by a brief description of the function of that routine.

Address	Function
5E60	Draws water (RFILL)
5E66	Draws waves on water (RPLOT)
5E6C	Draws dam
5E72	Draws protective front on dam
5E84	Draws line on top of dam
5E84	Draws electric plant
5E8A	Draws electric pole
58A0	Sets plane to start of screen; sets random altitude of plane
58AB	Checks fire button and moves plane up 2 pixels if depressed; puts out plane and moves it left 2 pixels; checks for left limit of screen
5800	Puts out dam explosion
5810	Puts out edge of flood wave
5780	Puts out final plane explosion
5848	Special CHR$(1) bomb pixels
5890	Puts out line in background color for dam wipe (RFILL)
5DC0	Opens door on top of dam and lifts missile
4A40	Shoots missile at plane
59D0	Dam explosion sound (calls SOUND routine in BASIC interpreter)
5CC4	Routine to move picture of airplane to screen; also used to open door on dam and lift missile, and to produce final explosion (pictures put in memory with RPLOTs)
5920	Random number generator; 5934H contains maximum number to be generated; stores random number in 5C79 (vertical screen coordinate of plane in control table)

To see the plane, dam door, and missile graphics stored in memory, use the MONITOR command MOVE 5800:6000:4000.

To see the explosion graphics, MOVE 5000:5800:4000.

This table must start at 5C74, because the MOVE routine looks there for it.

The routine to put the picture on the screen is called by another routine, located at 58AB, which controls movement of the picture. This Move routine is called from a GOTO loop in the BASIC portion of the program (line #103). The Move routine changes either the horizontal location in the control table only or both the horizontal and vertical locations. The routine checks to see if the hit button has been depressed; if it has, both the horizontal and vertical location bytes are reduced by two pixels. If the fire button has not been depressed, only the horizontal location byte is reduced by two pixels. After changing the screen coordinate(s) in the control table, the Move routine calls the routine at 5CC4 and redraws the plane according to the new coordinate specifications. In this way, graphics motion is accomplished.

Notes

1) In parameter control tables for program graphics, use a vertical coordinate below 50H (4A00) to display graphic on screen. Change the vertical coordinate to above 50H to store the picture permanently in memory.

2) Vertical screen coordinates are opposite in BASIC and machine code. In BASIC, the top of the screen is 77, in machine code 0 is the top. The horizontal coordinates are the same.

Suggested Programming Exercises.

1) Change the number of tries you have to knock down the dam.

2) Make the airplane fly more slowly.

3) Change the altitude at which plane enters screen from random, computer-generated to joystick pot knob controlled.

4) Change the angle at which the bomb falls.

5) Add other pictures to screen graphics.

BOWLING

A BASIC Game Program for 2 Players

Can't get a free lane at your local bowling alley? Now you can bowl at
home on your TV screen. Strikes, spares, splits, gutter balls--the Micro
Video BOWLING simulation program has them all! Automatic scoring uses
standard bowling rules. And, like the real thing, even a "bad leave" is
makeable, provided your aim is sharp.

LOADING THE PROGRAM: Insert the Microsoft 8K (or Level II) BASIC cassette
into the cassette deck. Depress the RESET button, press the "L" key, rewind
the tape if necessary, then depress the READ cassette button to load the
BASIC interpreter into the computer. When the BASIC "OK" prompt appears,
remove the BASIC interpreter cassette, insert the BOWLING cassette, depress
the RESET button, press the "L" key, rewind the tape if necessary, then
depress the READ cassette button to load the BOWLING program into the computer.
When the BASIC "OK" prompt reappears, type RUN and press the "CR" key to
execute the program.

GAME PLAY: Plug in both entertainment controllers. After you type "RUN",
a "bowling lane" appears on the screen, flanked on the top and bottom by
two sets of yellow lines that mark the gutters. The set of 10 pins appears
at the right end of the "lane". Left player always goes first. Each player
has up to two balls per frame with which to knock down all 10 pins. A
black "ball" appears on the left end of the alley and moves randomly up
and down. When you feel the ball is in the right position to get the maximum
score, depress the fire button on your controller to release the ball.
The ball "rolls" down the lane and hits the pins. If you release the ball
too close to the gutter, the ball will roll into the gutter and you get
no score for that ball. After the left player has thrown both balls (one
if you get a strike), it is the right player's turn. If you mark with
a strike or spare on the 10th frame, you get bonus balls (two or one, respectively)
to complete the scoring.

SCORING: The game is scored according to standard bowling rules. The
left player's score is shown on the top right side of the screen. The
right player's score is shown at the bottom right side of the screen.
The score shown is a running score that changes at the end of each frame.
The frames are displayed on the top left and bottom left side of the screen
as they are bowled. The frames are numbered and marked as they are bowled,
using the following symbols:

    X   Strike
    /   Spare
    -    Open frame

Top score, as in regular bowling, is 300. Frame numbers appear before the frame is scored and can be used to determine whose turn it is.

NEW GAME: When the last ball is thrown, the game ends. The word "OVER" appears on the left side of the lane, and the BASIC "OK" prompt appears. Type RUN and press the "CR" key to play another game.

PROGRAM LISTING: A complete listing of the BOWLING program is included on the following pages. It was produced with an Interact equipped with the Micro Video RS232 peripheral interface, RS232 BASIC, and a COMPRINT 912-S lineprinter. This listing has been included to illustrate programming techniques used and so that you can alter the program as desired.

```
1 CLEAR10
2 Z$=CHR$(1):PRINTCHR$(8)
5 DIML$(12),R$(12),L(12),R(12),P(10)
9 F1=1
10 CLS:COLOR2,3,0,7:F=1:C=3:L=6:R=6
50 FORI=12TO112:PLOTI,57,1:PLOTI,51,1:PLOTI,25,1:PLOTI,19,1:NEXT
80 GOSUB206
91 C=3:GOSUB1000
99 IFF>10THEN108
100 OUTPUTF,F1,70,3
108 GOSUB4000
109 BAL=0:IFN=2THEN149
110 IFN=0THENGOSUB210:N=1:GOTO80
111 IFF=13THEN130
115 IFF=10ANDL$(F)="/"THENN=0:F=11:GOTO110
120 IFF=10ANDL$(F)="X"THENN=0:F=11:GOTO110
125 IFF=12ANDL$(F-1)="X"THENN=0:GOTO110
130 IFF>10THENF=10:N=1
148 GOSUB4000
149 BAL=0:N=1:IFF<11THENOUTPUTF,F1,16,2
150 IFN=1THENGOSUB210:N=0
151 IFF=13THEN170
155 IFF=10ANDR$(F)="/"THENN=2:F=11:GOTO80
160 IFF=10ANDR$(F)="X"THENN=2:F=11:GOTO80
165 IFF=12ANDR$(F-1)="X"THENN=2:GOTO80
170 IFF>10THENF=10
180 F1=F1+6:F=F+1:OUTPUTF,F1,78,2
200 IFF=11THENOUTPUT"OVER",12,43,3:END
205 GOTO80
206 FORI=1TO10:P(I)=0:NEXTI:RETURN
210 SOUND7,4096:B=INT(25*RND(1))+28
211 V=1
215 OUTPUTZ$,6,B,2
220 IFFIRE(N)=0THEN300
225 B=B+V
226 Q=INT(27*RND(1))+22
227 IFB=QTHENOUTPUTZ$,6,B-V,0:V=V*-1:GOTO215
230 IFB=53ORB=28THENOUTPUTZ$,6,B-V,0:V=V*-1:GOTO215
235 OUTPUTZ$,6,B-V,0:GOTO215
300 SOUND5,510:OUTPUTZ$,I-4,B,0
302 FORI=6TO108STEP2:OUTPUTZ$,I-2,B,0
311 IFB>50THENB=56
312 IFB<30THENB=24
313 OUTPUTZ$,I,B,2:NEXTI
401 SOUND7,4096:SOUND3,24
410 OUTPUTZ$,108,B,0
420 IFB<>24ANDB<>56THEN440
```

```
430 IFBAL<>1THENGOSUB206
435 GOTO2065
440 IFBAL=1THENGOSUB525:GOSUB1000:GOTO5000
445 GOSUB2000:SOUND7,4096:RETURN
525 ONSGOTO530,530,530,530,530,550,555,565,570
530 IFB=38ORB=39ORB=42ANDP(7)=0THENGOSUB4000
531 IFB=38ORB=41ORB=42ANDP(10)=0THENGOSUB4000
532 IFB=40ANDP(10)=1THENGOSUB4000:P(7)=0:GOTO535
533 IFB=40ANDP(7)=1THENGOSUB4000:P(10)=0:GOTO535
534 IFB=40THENGOSUB4000:P(7)=0:P(10)=0
535 IFB<39ANDB>34THENGOSUB4000:P(1)=0:P(2)=0
536 IFB>42ANDB<46THENGOSUB4000:P(1)=0:P(3)=0
537 IFB>43ANDB<49THENP(4)=1:P(8)=1:P(7)=1
540 GOTO570
550 IFB=32ORB=48THENGOSUB4000:RETURN
551 IFB<37ANDB>31ANDB<>34THENP(6)=1:P(7)=1
552 IFB=34THENP(6)=1
553 IFB>43ANDB<49ANDB<>46THENP(4)=1:P(7)=1
554 GOTO570
555 IFB=38ANDP(10)=1THENGOSUB4000
556 IFB=42ANDP(7)=1THENGOSUB4000
557 IFB=39THENP(1)=1:P(2)=1:P(9)=1:P(3)=1
558 IFB=40THENP(1)=1:P(2)=1:P(3)=1
559 IFB=41THENP(1)=1:P(2)=1:P(3)=1:P(8)=1
560 IFB>40ANDB<46THENP(2)=1:P(8)=1:P(4)=1
561 IFB<38ANDB>34THENP(3)=1:P(9)=1:P(6)=1
562 IFB=38ANDP(7)=0THENGOSUB4000
563 IFB=42ANDP(10)=0THENGOSUB4000
564 GOTO570
565 IFB=39ORB=35THENP(3)=1:P(7)=1
570 IFB>46THENP(7)=1
571 IFB<34THENP(10)=1
572 IFB<40ANDB>34THENP(3)=1
573 IFB<37ANDB>31THENP(6)=1
574 IFB>43ANDB<49THENP(4)=1
575 IFB=46THENP(4)=1
576 IFB=38THENP(1)=1:P(2)=1:P(3)=1
600 RETURN
1000 M=31:O=49:C=3:I=10
1005 FORX=102TO84STEP-6:FORY=MTOOSTEP6
1012 IFP(I)=1THENOUTPUT"-",X,Y,0:GOTO1020
1015 OUTPUT"-",X,Y,C
1020 IFY=OTHENM=M+3:O=O-3
1022 I=I-1:NEXTY:NEXTX:RETURN
2000 IFB<35ANDB>31THENGOSUB206:P(10)=1:P(9)=1:P(6)=1
2001 IFB<32THENGOSUB206:P(10)=1
2005 IFB=35THENP(1)=0:P(2)=0:P(8)=0
2010 IFB=36THENP(1)=0:P(2)=0:P(4)=0
2015 IFB=37THENP(1)=0:P(2)=0:P(10)=0
2020 IFB=38THENP(10)=0
2025 IFB=40THENP(4)=0:P(6)=0:P(7)=0:P(10)=0
2030 IFB=41THENP(7)=0
2035 IFB=42THENP(3)=0:P(10)=0
2040 IFB=43THENP(1)=0:P(3)=0:P(7)=0
2045 IFB=44THENP(1)=0:P(3)=0:P(6)=0
2050 IFB=45THENP(1)=0:P(3)=0:P(9)=0
2055 IFB>45ANDB<49THENGOSUB206:P(7)=1:P(8)=1:P(4)=1
2056 IFB>48THENGOSUB206:P(7)=1
```

```
2065 GOSUB1000
2075 GOTO5000
4000 FORI=1TO10:P(I)=1:NEXTI:RETURN
5000 S=0:BAL=BAL+1:FORI=1TO10:IFP(I)=0THEN5030
5025 S=S+1
5030 NEXTI:SOUND7,4096
5040 IFBAL=1ANDS=10THENTONE10,1000:FORI=1TO10:TONE5,1000:NEXTI
5045 IFBAL=2ANDS=10THENTONE50,1000
5050 IFBAL=2ANDS<10THENTONE10,1000
5055 IFN=0THEN6995
5060 IFN=1THEN8995
6995 IFBAL=2THEN7048
6996 IFF=10RF=2THEN7020
6997 IFF=11ANDL$(F-2)="X"ANDL$(F-1)="X"THENL(F-2)=L(F-3)+20+S:GOSUB8020
7001 IFF=11ANDL$(F-1)="/"THENL(F-1)=L(F-2)+10+S:GOSUB8030:F=13:RETURN
7002 IFF=11ANDS=10THENL$(F)="X":GOSUB7090:F=12:RETURN
7003 IFF=11ANDS<10THENF=12:GOTO210
7004 IFF=12ANDL$(F-1)="X"THENL(F-2)=L(F-3)+20+S:GOSUB8020:F=13:RETURN
7010 IFL$(F-2)="X"ANDL$(F-1)="X"THENL(F-2)=L(F-3)+20+S:GOSUB8020
7020 IFL$(F-1)="/"THENL(F-1)=L(F-2)+10+S:GOSUB8030
7030 IFS=10THENL$(F)="X":GOTO7090
7045 GOTO210
7048 IFF=12THENL(F-2)=L(F-3)+10+S:GOSUB8020:F=13:RETURN
7050 IFL$(F-1)="X"THENL(F-1)=L(F-2)+10+S:GOSUB8030
7060 IFS=10THENL$(F)="/":GOTO7090
7070 IFS<10THENL$(F)="-":L(F)=L(F-1)+S:GOSUB7090
7080 GOSUB8000:OUTPUTL(F),70,64,2:RETURN
7090 OUTPUTL$(F),L,64,2:L=L+6:RETURN
8000 OUTPUT"***",76,64,0:OUTPUT"000",76,64,0:RETURN
8020 GOSUB8000:OUTPUTL(F-2),70,64,2:RETURN
8030 GOSUB8000:OUTPUTL(F-1),70,64,2:RETURN
8995 IFBAL=2THEN9048
8996 IFF=10RF=2THEN9020
8997 IFF=11ANDR$(F-2)="X"ANDR$(F-1)="X"THENR(F-2)=R(F-3)+20+S:GOSUB10020
9000 IFBAL=2THEN9048
9001 IFF=11ANDR$(F-1)="/"THENR(F-1)=R(F-2)+10+S:GOSUB10030:F=13:RETURN
9002 IFF=11ANDS=10THENR$(F)="X":GOSUB9090:F=12:RETURN
9003 IFF=11ANDS<10THEN210
9004 IFF=12ANDR$(F-1)="X"THENR(F-2)=R(F-3)+20+S:GOSUB10020:F=13:RETURN
9010 IFR$(F-2)="X"ANDR$(F-1)="X"THENR(F-2)=R(F-3)+20+S:GOSUB10020
9020 IFR$(F-1)="/"THENR(F-1)=R(F-2)+10+S:GOSUB10030
9030 IFS=10THENR$(F)="X":GOTO9090
9045 GOTO210
9048 IFF=11THENR(F-1)=R(F-2)+10+S:GOSUB10030:F=13:RETURN
9050 IFR$(F-1)="X"THENR(F-1)=R(F-2)+10+S:GOSUB10030
9060 IFS=10THENR$(F)="/":GOTO9090
9070 IFS<10THENR$(F)="-":R(F)=R(F-1)+S:GOSUB9090
9080 GOSUB10000:OUTPUTR(F),70,12,2:RETURN
9090 OUTPUTR$(F),R,10,2:R=R+6:RETURN
10000 OUTPUT"***",75,12,0:OUTPUT"000",76,12,0:RETURN
10020 GOSUB10000:OUTPUTR(F-2),70,12,2:RETURN
10030 GOSUB10000:OUTPUTR(F-1),70,12,2:RETURN
```

CHECKERS

A BASIC Game Program for 1 or 2 Players

With this Microsoft 8K BASIC program, the traditional game of checkers comes alive on your TV screen in brilliant color. No more hunting for the missing pieces--they're stored in your computer's memory, so they can't get lost! Play the computer or a friend using standard Checkers rules to jump, double-jump, and "king" your pieces.

8K Fast ......

LOADING THE PROGRAM: Load the Microsoft 8K BASIC tape by depressing the RESET button, pressing the "L" key, rewinding the tape if necessary, then depressing the READ cassette button to load the BASIC interpreter into your computer. When the BASIC "OK" prompt appears, remove the BASIC tape and insert the CHECKERS program cassette into the tape drive. Depress the RESET button, press the "L" key, rewind the tape if necessary, then depress the READ cassette button to load the CHECKERS program into your computer. When the BASIC "OK" prompt reappears, type RUN to execute the program.

GAME OPTIONS: After you type RUN, the computer asks you to define the game ("GAME? 2=2 PEOPLE, C=AGAINST COMPUTER"). Type "2" to play with two people, and plug in both entertainment controllers. If you type "C" to play the computer, the computer asks which color it should play ("COMPUTER PLAYS L=LEFT(BLACK), R=RIGHT(WHITE)"). If you want to play with the black pieces, type "R" and the computer will play white. You should then plug your entertainment controller into the left joystick port. If you want to play with white pieces, type "L" and the computer will play black. In this case, plug your controller into the right joystick port.

GAME PLAY: Black always moves first. Black player moves pieces toward the bottom of the screen; white moves toward the top. When it is your turn, use the joystick lever to move the flashing cursor over the piece you want to move. Depress the hit button to pick up the piece. Then move the cursor to the square where you want to place the piece and depress the hit button again to put the piece down. If you attempt to make an illegal move, the computer will not allow you to move and will make a deep (negative) tone. If you want to cancel your move and start over, move the cursor back to the square in which you picked up the piece and depress the hit button to put the piece back down. You will hear a medium tone cancelling the move. A high-pitched tone acknowledges that you have picked up a piece.

To execute a multiple jumping move, position the cursor and pick up the piece. Move it to the appropriate square for the first jump and depress the hit button to execute the jump. Because there are still other jumps

you can make, your piece is still considered to be picked up. Do not press
the hit button to pick the piece up again. Move the cursor to the square
appropriate for the next jump, then depress the hit button to execute the
jump. Repeat this process until all jumps for that turn have been made.

In a game against the computer, the blinking cursor appears in the upper
or lower left corner of the playing board to signal that it is your turn.
In the two player game, the flashing cursor appears in the upper left corner
if it is black's turn and the lower left corner if it is white's turn.

RULES: The CHECKERS program plays according to the following rules:

  1. If you have a jump opportunity, you must take it.

  2. Non-king pieces can move only toward the opponent's side of the board.
      Pieces can move on the diagonal only, one square at a time.

  3. When you move a piece into your opponent's backmost row, the piece
      is transformed into a king (identified by its larger size). Kings
      can move in an either upward or downward direction. Like the smaller
      pieces, however, kings can move only on the diagonal, one square
      at a time.

PROGRAM LISTING: The attached program listing has been provided so that
you can examine the code for this BASIC game program, and use it to learn
more about writing your own BASIC game programs. This listing was produced
using an Interact equipped with the Micro Video RS232 peripheral interface,
RS232 BASIC, and a COMPRINT 912-S lineprinter.

```
30 CLEAR:Q=1:G=7:L=2:DJ=3
40 A=RND(-PEEK(24559)):COLOR4,1,7,0
100 CLS:OUTPUT"GAME?",35,57,2:OUTPUT"2 = 2 PEOPLE",20,37,2
120 OUTPUT"C = COMPUTER",20,27,2
130 A$=INSTR$(1):IFA$="2"THENCO=7:GOTO240
150 IFA$<>"C"THENTONE500,93:GOTO130
200 CLS:OUTPUT"COMPUTER PLAYS",20,52,2:OUTPUT"L = LEFT(BLACK)",10,42,2
201 OUTPUT"R = RIGHT(WHITE)",10,32,2
210 CO=-1:A$=INSTR$(1):IFA$="L"THEN240
220 CO=1:IFA$<>"R"THEN210
240 GOTO7000
700 CLS:W$="LEFT":IFPL<0THENW$="RIGHT"
705 IFCO<7ANDPL<>COTHENW$="COMPUTER"
```

```
706 OUTPUTW1+" WINS",20,50,2
707 IFFIRE(0)ANDFIRE(1)THEN707
710 CLS:OUTPUT"SAME GAME ?",30,42,2
712 IFFIRE(0)=0ORFIRE(1)=0THEN712
713 POKE24529,0
715 A=PEEK(24529):IFATHEN720
716 IFFIRE(0)ANDFIRE(1)THEN715
717 DN=1:RETURN
720 IFQ=89ORQ=121THEN717
730 IFA=78ORA=110THENDN=0:RETURN
740 GOTO713
2000 V=L
2001 IFV>ZTHEN2010
2004 CX=X:CY=Y:GOSUB5300
2010 IFABS(A(X,Y))=2THEN2030
2020 PLOT22+X*9,6+Y*9,V,5,3:PLOT23+X*9,5+Y*9,V,3,5:RETURN
2030 PLOT21+X*9,5+Y*9,V,7,5:PLOT22+X*9,4+Y*9,V,5,7:RETURN
5000 CX=FX:CY=FY:IFFIRE(PJ)=0THEN5000
5010 X=23+9*CX:Y=6+9*CY:V=0:VJ=DJ:GOSUB5200:IFV=2THENVV=3:GOTO5020
5015 VV=2
5020 PLOTX,Y,VV,3,3:IFFIRE(PJ)=0THENPLOTX,Y,V,3,3:RETURN
5030 CJ=JOY(PJ):IFCJTHENPLOTX,Y,V,3,3:GOTO5100
5035 VJ=VJ-1:IFVJTHEN5020
5037 VJ=DJ
5040 PLOTX,Y,V,3,3:IFFIRE(PJ)=0THENRETURN
5050 CJ=JOY(PJ):IFCJTHEN5100
5055 VJ=VJ-1:IFVJTHEN5040
5057 :VJ=DJ:GOTO5020
5100 CJ=JOY(PJ):CX=(CX-(CJAND1)+(CJAND2)/2)AND7
5110 CY=(CY+(CJAND4)/4-(CJAND8)/8)AND7:GOTO5010
5200 V=VAL(MID$("33022",A(CX,CY)+3,Q))
5201 IFV>0THENRETURN
5300 V=1-((CYANDQ)+(CXANDQ))ANDQ:RETURN
6000 PLOT19,2,1,74,74:PLOT20,3,0,72,72
6060 FORP=0TO7:R=(PAND1)*9:FORQQ=0TO3
6100 PLOT20+9*P,R+3+18*QQ,1,9,9:NEXT:NEXT:RETURN
6200 RU=Z
6201 M=-Q
6210 FORI=ZTOG:FORJ=ZTOG
6230 M=-M:A(I,J)=0
6240 IFJ>LANDJ<5THEN6280
6250 IFM<ZTHEN6280
6260 A(I,J)=Q:V=L:IFJ>LTHENA(I,J)=-Q:V=3
6270 X=I:Y=J:GOSUB2001
6280 NEXT:M=-M:NEXT:RETURN
7000 CLS:DN=1:GOSUB6000:GOSUB6200:PL=1
7040 PL=-PL:GOSUB7200:IFDN=1THEN7040
7090 GOSUB700:IFDN=1THENGOTO7000
7110 GOTO100
7200 IFPL=COTHENGOSUB8000:M=0:IFDN=1THEN7320
7210 FX=Z:FY=0:PJ=1:M=0:IFPL<0THENFY=7:PJ=0
7220 RU=1:GOSUB8000:RU=0:IFDN=0THENRETURN
7230 GOSUB5000
7240 IFA(CX,CY)*PL>ZTHEN7270
7250 TONE500,99:GOTO7230
7270 FX=CX:FY=CY:TONE99,99
7280 GOSUB5000
7285 IFFX<>CXORFY<>CYTHEN7290
```

```
7287 IFM=0THENTONE200,99:GOTO7250
7288 RETURN
7290 IFA(CX,CY)=ZTHEN7320
7300 TONE500,99:GOTO7280
7320 IFABS(FX-CX)=1AND(CY-FY=PLORABS(A(FX,FY))=2)THEN7550
7325 IFABS((FX-CX)*(FY-CY))=4THEN7340
7330 GOTO7300
7340 TX=(FX+CX)/L:TY=(FY+CY)/L
7350 IFA(TX,TY)*PL>=0THEN7300
7355 XX=CX:YY=CY
7360 X=FX:Y=FY:V=Z:GOSUB2001
7460 X=XX:Y=YY:V=L
7465 A(X,Y)=A(FX,FY):IFY=7ORY=ZTHENA(X,Y)=PL*L
7470 IFPL<ZTHENV=3
7480 GOSUB2001
7490 V=Z:X=TX:Y=TY:GOSUB2001
7500 A(TX,TY)=Z:A(FX,FY)=Z
7505 IFYY=7ORYY=ZTHENA(XX,YY)=PL*L
7510 M=M+Q
7515 FX=XX:FY=YY
7525 CX=-1:C=PL:D=PL:IFABS(A(FX,FY))=2THENC=-1:D=1
7527 FORA=-QTOQSTEPL:FORB=CTODSTEPL
7529 I=FX+A:J=FY+B
7530 IFI<GORI>60RJ<QORJ>6THEN7535
7532 IFA(I,J)*PL<ZANDA(I+A,J+B)=ZTHENCX=I+A:CY=J+B
7535 NEXT:NEXT:IFCX<0THENRETURN
7540 IFPL<>COTHEN7280
7545 GOTO7285
7550 XX=CX:YY=CY
7555 IFM<>0ORM2=2THEN7300
7570 X=FX:Y=FY:V=Z
7580 GOSUB2001
7590 X=XX:Y=YY:V=L
7600 IFPL<ZTHENV=3
7605 A(X,Y)=A(FX,FY):IFY=GORY=ZTHENA(X,Y)=PL*L
7610 GOSUB2001
7630 A(FX,FY)=Z:RETURN
8000 MO=-20:M2=0
8030 FORX=ZTOG:FORY=ZTOG
8040 IFA(X,Y)*PL<=0THEN8130
8090 ONABS(A(X,Y))GOTO8400,8500
8130 NEXT:NEXT:IFMO=-20THENDN=0
8180 RETURN
8400 C=PL:D=PL
8405 FORA=-QTOQSTEPL:FORB=CTODSTEPL:I=X+A:J=Y+B
8410 IFI<>(IANDG)ORJ<>(JANDG)THEN8460
8420 IFA(I,J)=0ANDM2<2THENOM=2-2*ABS(A(X,Y)):M2=1:GOTO8600
8430 IFA(I,J)*PL>=0THEN8460
8440 I=I+A:J=J+B:IFI<>(IANDG)ORJ<>(JANDG)THEN8460
8450 IFA(I,J)=ZTHENOM=3:M2=2:MO=-20:GOTO8600
8460 NEXT:NEXT:GOTO8130
8500 C=-Q:D=Q:GOTO8405
8600 IFJ=3.5+3.5*PLANDABS(A(X,Y))<2THENOM=OM+3
8605 IFRUANDM2=2THENRETURN
8606 IFRUTHENMO=-19:GOTO8450
8610 IFY=3.5-3.5*PLTHENOM=OM-2
8620 IFABS(A(X,Y))AND(J-Y)*PL>0THENOM=OM-1
8660 P=(I-X)/M2:QQ=(J-Y)/M2
```

```
8665  IFI=0ORI=7THENOM=OM+1:GOTO8680
8666  IFJ+QQ<0ORJ+QQ>7THEN8680
8670  IFA(I+P,J+QQ)*PL<0THENOM=OM-1
8675  IFA(I-P,J+QQ)*PL<0ANDA(I+P,J-QQ)=0THENOM=OM-1:GOTO8680
8680  FORP=-1TO1STEP2:FORQQ=-1TO1STEP2
8685  IFX=0ORX=7ORY=0ORY=7THEN8695
8690  IFA(X+P,Y+QQ)*PL=-2THEN8692
8691  IFQQ<>PLTHEN8695
8692  IFA(X+P,Y+QQ)*PL<0ANDA(X-P,Y-QQ)=0THENOM=OM+1
8695  NEXT:NEXT
8700  IFOM<MOTHEN8460
8710  IFMO=OMAND(1AND(16*RND(1)))THEN8460
8720  FX=X:FY=Y:CX=I:CY=J:MO=OM:GOTO8460
```

COMPUTER QUIZ

Program Instructions

Computer Quiz is a program that lets up to five players or teams compete
in their knowledge of facts on a variety of topics. These topics,
including books, movies, sports, current events, history, and general
trivia, are provided in sets of 80 questions on cassette tape for use
with the program. You'll find some questions easy, others challenging.
You can use this program to expand your awareness of useful (and some-
times useless!) facts. Or, used in a group of people, the program promises
to stimulate conversations on a wide variety of subjects.

LOADING THE PROGRAM: Insert the program cassette into the tape drive,
depress the RESET button, press the "L" key, rewind the tape if necessary,
then depress the READ cassette button to load the program into the computer.

LOADING DATA SETS: When the Computer Quiz master program has loaded,
the computer instructs you to enter a data set. Follow the instructions
as presented on the sequence of screens to load in one of the available
data sets.

THE QUIZ: When the data set has been loaded off the cassette, the computer
asks you to enter the first player's name. Enter the player or team
name (up to 9 characters), and press the "CR" key. Repeat for other
players or teams. Up to 5 players or teams may compete in Computer Quiz.
If you want to play with fewer than 5 players, simply press the "CR"
key when asked for player name.

After all players' names are entered, the computer identifies the player
whose turn it is and presents, one at a time, a series of clues. Type
in the answer to each clue and press the "CR" key. The computer will
tell you whether the answer is wrong or right and calculate your score
accordingly. If you enter an incorrect answer, the computer presents
the correct response. The number of questions asked of each player will
vary from data set to data set; this number has been predetermined.
Follow the guidelines presented on the following page to ensure that
you are not "cheated" out of points for your response.

At the end of each player's turn, the computer displays that player's
score. After all players' turns are over, the computer presents the
normalized scores, with winner(s) highlighted.

NEW GAME: When all players have had a turn and the normalized scores
have been presented, the computer asks "SAME PLAYERS? Y=YES, N=NO".
If you respond "Y" to this question, the computer begins the game again
with the same players, drawing from questions in the data set that have
not been used. No questions will be repeated until all have been asked
once. If you respond "N" to this query, the computer asks "SAME SUBJECT?
Y=YES, N=NO". To play with the same data set, but change players, respond
"Y" to this question. The computer will then ask you to enter new player
names. If you respond "N", the computer asks you to enter a new data
set.

GENERAL GUIDELINES FOR ANSWER ENTRY

Due to the Interact's graphic resolution, we're limited in string space
for both our clues (64 characters) and the answers (34 characters).
In the space available, we've tried to be as flexible as possible in
answers allowed. However, on long answers in particular, you'll be limited
to one correct answer. To avoid frustrations of entering an answer that
is "almost correct", we suggest you follow these guidelines:

1) In most cases, "a", "an", and "the" are extraneous and can be left
   out. Use these words only when part of a title (e.g., movie or book).
   Follow the general rule "when in doubt, leave it out".

2) If use of "and" is required, substitute a plus sign (+).

3) Generally, you can enter a person's last name only for a correct
   answer. Exceptions are when a person is more commonly referred to
   by his or her first or nickname or when more than one famous person
   in the same time period share the same last name.

   This is particularly important in the "Famous People" data set.
   In that set, remember to use "+" for "and" and to use last names
   only, with exceptions as above. Always put the two "halves" of the
   pair in the order in which they are commonly used or in the order
   in which they are presented in the clue.

4) Use periods (.) following abbreviations.

5) People's titles are usually omitted. For example, enter "Nixon"
   as opposed to "President Nixon".

EARTH OUTPOST I

Program Instructions

(for 1 player)

The mission of your space station is to defend the world against attacking alien forces in this electrifying action game.

LOADING THE PROGRAM: Insert the program cassette·into the tape deck, depress the RESET button, depress the "L" key, rewind the tape if necessary, then depress the READ button to load the program into the computer.

PLAYING THE GAME: The object of this game is to shoot down invading space craft and prevent them from destroying the Earth's cities. You score points for every alien missile or rocket you successfully defend against. The game ends when one of the cities is hit and demolished.

To play, plug a controller into the left controller port. You control the movement of your space station with the joystick lever and can move the outpost up, down, diagonally, or across the screen to position it for defense against the invading forces. You shoot at the rockets or missiles by pressing the joystick hit button.

Your space station can be hit by the incoming missiles or rockets. With each such collision, the outpost sustains damage that reduces its speed by approximately 25%, making the Earth's defense more difficult.

SCORING: The alien forces have two types of space craft: MV315 Destructor Rockets (white) and MV502 Anti-Matter Missiles (red). You score between 10 and 20 points for annihilating a Destructor Rocket, depending on the altitude above Earth from which you shoot. Highest points are scored when the outpost is near the top of the screen. Destruction of an Anti-Matter Missile scores 20 points in addition to the altitude-based score.

For each alien missile that hits the Earth, 7 points are deducted from your score. Note that an Anti-Matter Missile has twice the destructive range of a Destructor Rocket.

When a city is destroyed, ending the game, the computer displays your score and the best previous score. A new game begins automatically.

A NEW RECORD: If your score tops the highest previous score, the computer prints "A NEW RECORD!" and asks for your name. Type in your name and press the "CR" key to begin a new game.

MICRO ▨ VIDEO™

FOOTBALL

Program Guide   (for 2 players)

FOOTBALL gives you the excitement of a football stadium on your own television screen. Run, pass, and punt the ball to lead your team down the field to victory in this action-packed game.

LOADING THE PROGRAM: Insert the program cassette into the tape deck, depress the RESET button, depress the "L" key, rewind the tape, then depress the READ button to load the program into the computer. Plug in joysticks.

STARTING THE GAME: The game begins as the two teams run onto the field and "huddle" at the 20-yard line. The starting offensive team is selected at random by the computer. On the screen, the offensive team is indicated by the color of the yard markers and the text at the bottom of the screen.

PLAY SELECTION: When the players are in the "huddle," each team chooses a formation. Eight different formations are available. To choose a formation for your team, move the joystick lever in one of the eight possible directions. The offensive team chooses a formation first. After the defensive formation is selected, the offensive team has 10 seconds to hike the ball and begin a play before a 5-yard penalty is charged.

HIKING THE BALL: Hike the ball to begin a play by pressing the hit button on your joystick.

PLAYER CONTROL AND MOVEMENT: You control only one player on the team, the player whose marker flashes. On offensive plays, the player you control can change. The player with the ball is the player whose movements you control with your joystick lever. Movements of other players on the field are computer-controlled. Note that movement corresponds to the joystick positions. A left or right position moves your player up or down the field (across the TV screen). An up or down position moves your player across the field (up or down your screen). (The joystick should be held so that the cord extends away from you.)

PASSING THE BALL: To pass the ball, move the joystick lever in the direction you want to throw the ball while pressing the joystick hit button. Note that the joystick lever position controls the direction of the pass in the same manner as described for player movement. If the ball is intercepted, the defensive player who intercepts it becomes the ball carrier and can run with the ball.

PUNTING THE BALL: In a fourth down situation, you may elect to punt the ball. To punt, depress the joystick hit button while you are selecting your offensive play formation. Then, after you hike the ball, move the joystick lever in

the direction you want to kick the ball while pressing the hit button again. The ball will be "kicked" a random distance (determined by the computer) and it can be run back by the receiving team. The joystick lever position controls the kicking direction in the same manner as described·for·player movement. You can also use this feature to "quick kick" on other downs.

SCORING: You score a touchdown -- 7 points -- for carrying the ball into your opponent's end zone. You score a safety -- 2 points -- for trapping your opponent in his own end zone.

GAME TIME: The regulation game time is 5 minutes. The game clock runs only during plays. If the score is tied at the end of regulation play, a 1 minute sudden death overtime period begins. Initial possession of the ball in overtime is determined randomly by the computer. Sudden death overtime periods are repeated until one team scores, breaking the tie.

MICRO *M* VIDEO

HOT ROCKS

An Action/Arcade Game for 1 Player

The year is 2058. As long predicted by doom-sayers, the proliferation of nuclear arms and continuing territorial confrontations have led to World War III. The fiery holocaust fragments the earth and the force of the explosions sends its pieces hurling into space.

The sole survivors are two groups of people—the Good Guys and the Bad Guys—who had earlier been sent to inhabit space stations on the moon. Among other technological advances, We (the Good Guys, of course) have developed a laser that effectively reduces the half-life of radioactive material. This weapon is all we have to protect our new world against contamination by the "hot rocks," and the Bad Guys want it! On each mission into space, your job is to aim and fire your laser to destroy all the hot rocks you encounter. The Bad Guys compound your task—not only must you demolish the hot rocks, but you must also fend off their attack craft to survive. Then, when you've accomplished your mission, you must successfully land your ship on the lunar surface to recharge your laser. Then, you're off again on another mission in this high-tension adventure in space.

P.S. Notice anything unusual about the front screen and opening graphics sequence?

Loading the Program: Insert the program cassette into the tape drive. Depress the RESET button, then press the "L" key. Rewind the tape if necessary using the REWIND cassette button, then depress the READ cassette button to load the program into the computer's memory. Plug in the left entertainment controller.

Game Options: After the program has loaded, the computer asks you to select a skill level for play ("1-5, HARD-EASY"). The skill level setting controls the speed at which the hot rocks move on the screen. Press 5 for the easiest game or one of the lower numbers for more difficult game play.

Game Play: The game begins automatically after you select a skill level. The game is a series of missions into space, each of which consists of two parts. You must first destroy all hot rocks, then reland on the lunar surface to refuel. You begin the game, regardless of skill level, with four space-ships. For every 10,000 points you score, you are awarded an extra ship. You can lose ships in three ways:

- Collision with a hot rock
- Collision with or direct hit from the enemy saucer
- Unsuccessful lunar landing

After the opening graphics sequence, the lunar landscape appears on the screen, and your ship automatically takes off. When the lunar surface drops

from view, the hot rocks begin drifting across the screen around your ship.
Use the joystick to control the direction and motion of your ship, and the
fire button to shoot at the hot rocks and the Bad Guys' saucer when it
appears. Press the joystick lever to the <u>right</u> to rotate your ship in a <u>clock-wise</u> direction. Press the lever to the <u>left</u> to rotate <u>counter-clockwise</u>.
This allows you to aim the nose of your craft in any direction for firing.
Depress the fire button to shoot at the rocks after aiming. Note that you
can hold the fire button down to shoot your laser repeatedly and quickly.
However, only four blasts from your gun can be active on the screen at any
one time. To move your craft around on the screen, press the joystick up
(forward) for thrust. Your ship will move in the direction in which the nose
of your craft is pointed when you fire your thrust jets. And, careful--a
little thrust goes a long way in a gravity-free environment!

If you lose your spaceship during this phase of the game through a hot rock
collision or a bullet from the enemy saucer, your ship explodes and disappears
from the screen. To resume play with a new ship, pull the joystick lever
down (toward you).

When you have reduced all hot rocks to an inactive state, that phase of your
mission is complete. The lunar surface reappears on the screen, and your
ship flies in from the left side of the screen. You must land your craft,
upright, in the flat area between the two craters. Use the joystick lever
to control the direction of your ship as described previously (left=counter-clockwise, right=clockwise). Use thrust (joy up) as necessary to position
your ship and to slow your descent for a safe touchdown. You will crash and
lose a ship if you descend too fast, if you land outside the target area,
or if the nose of your craft is not pointed upward. You score bonus points
for a successful landing as described below.

Scoring: You score points for reducing the radioactive half-life of the
hot rocks according to their size. The less radioactive the material, the
smaller it appears and the harder to hit. The largest hot rocks are worth
50 points, the medium-sized are worth 90 points, and the smallest are worth
120 points. You score 500 points each time you destroy the enemy saucer.
Note that, as your score increases, the saucer appears more frequently and
fires more often.

As you begin the landing process after each mission, a fuel counter appears
in the upper right corner of the screen. This counter reflects how much fuel
you consume in firing your thrust jets during landing. You are awarded bonus
points for a successful landing at a rate of ten times the amount of fuel
remaining. The longer you play, the normal wear-and-tear of interplanetary
travel takes its toll on your craft, making it less fuel efficient, and you
burn more fuel during the landing process.

The maximum score you can achieve is 655,350. If you reach this score, the
game counter resets to zero and the skill level is reset to the easiest level
of play.

Game End: The game ends when the last of your available ships is destroyed.
The computer displays the current high score, the scoreholder's name, and
your score. If your score beats the previous high score, the computer asks
you to enter your name ("A NEW RECORD! TYPE YOUR NAME"). Type in your name
and press the "CR" key. The computer then asks you if you want to play again
at the same skill level ("SAME GAME? Y/N"). Press the "Y" key or the fire
button to play another game at the same skill level. Press "N" to select
a new skill level for play.

MICRO 🄼 VIDEO

305 North First St.
P.O. Box 7357
Ann Arbor, MI 48107
(313) 996-0626

Jumping Jack

An Action/Arcade Game for 1 Player

Who's the top stunt person in Hollywood? The challenge is to climb a
series of buildings in a desperate race against the clock. You must scramble
through ladder and scaffolding labyrinths, jump over rolling barrels and
holes in the platforms, and dodge falling objects to reach the top. It's
the ultimate test of your athletic prowess and stamina.

LOADING THE RPOGRAM: Insert the program cassette into the tape drive.
Depress the RESET button, press the "L" key, rewind the tape if necessary,
then depress the READ cassette button to load the program into the
computer. Plug in the left entertainment controller.

GAME PLAY: When the program has loaded, the first scaffolding maze appears
on the screen and game play begins. The object is to reach the top of
the screen in an alloted amount of time. To do this, you'll have to scurry
up ladders, leap over holes in the platforms, and hurdle over or climb
out of the way of barrels rolling across the platform. Score bonus points
by picking up the silver dollars scattered along the way. Use the joystick
lever to move Jumping Jack across the boards and up and down the ladders.
Press the firebutton to make Jack jump in the direction you're holding
the joystick lever. The clock at the top center of the screen continuously
counts the seconds down.

If you reach the top of the first building, you find yourself at the bottom
of another, more complex structure. Objects fall from the rooftop, and
there are more ladders and shorter platforms. The objective is, however,
the same — reach for the top!

At the top of the second scaffolding maze, you enter a new building. The
scaffolding mazes are similar to those in the first building, but the
challenge is greater because there are more barrels and falling objects.
You are given a longer time to get to the top, however. When you complete
the second building, you enter a third. The only limit to the number of
buildings in a single challenge is that of your endurance.

You start with three chances in the competition. You lose one each time
a rolling barrel or falling object knocks you off or you jump too soon
or far and plummet to the pavement below. You also lose a chance if you
don't make it to the top in time. But, you gain one extra chance when
you reach the top of the second scaffolding, and for each 5,000 points
you score thereafter. You can accumulate a maximum of five chances at
any one time.

GAME SCORING: Your running score appears at the top right of the screen. The number of points you earn for jumping over a barrel or picking up a silver dollar is equal to the total time allotted to complete that screen. As you move through the building levels, the time and bonus points increase from 40 to a maximum of 90. Also, the number of barrels on the screen at any given time increases from three to a maximum of six. Every time you successfully reach the top, you earn a bonus equal to the seconds remaining on the clock multiplied by 10.

GAME END: The game ends when you use up all your attempts to climb the buildings, and is signalled by the "GAME OVER" message. After your first play, or each time you reach a new high score, you will see the message "A NEW RECORD! TYPE YOUR NAME." Type any name up to 14 characters long, then press the "CR" key. The computer then displays your high score and the question, "PLAY AGAIN? (Y/N)". To play again, either hit "Y" or the fire button. If you do not reach a new high score, you will immediately see the "PLAY AGAIN" screen.

MICRO *MV* VIDEO™

305 North First St.
P.O. Box 7357
Ann Arbor, MI 48107
(313) 996-0626

MAZES AND MONSTERS

A Three-Dimensional D&D-Style Strategy Game

The Sacred Orb, an object of worship by all, has vanished! It is believed
to be hidden in the deepest ¯ecesses of a dungeon in a 7th century castle.
Your quest is to find it and return it to the castle. To accomplish your divine
mission, you must brave the p∍rils of the dungeon, using your fighting skills
and magical powers to overcome the many and varied evil forces that lurk in
it's passages.

Mazes and Monsters is a D&D-style adventure game that was patterned using
many of the rules and attributes common to such games. You can choose to play
as one of three races — elf, human, or dwarf, with a set of personal character-
istics that influence how well you fight and use magic.

Every step of the way, your location in the dungeon is depicted with graphics
that give you a three-dimensional perspective on play. Monsters and evil
spirits that you encounter are also displayed as colorful images. Better make
a map, or you'll never find your way back out. The dungeon has 8 levels, each
of which is a 16 x 16 grid. (That's 2,048 rooms to keep track of!)

Mazes and Monsters is a game you might well play forever. An entirely new
maze is generated each time you load or restart the program, unless, of course,
you've saved a previous maze on tape for later replay — that facility is also
built into the program.

LOADING THE PROGRAM: Insert the program cassette into the tape drive. Depress
the RESET button, then press the "L" key. Rewind the tape if necessary using
the REWIND cassette button, then depress the READ cassette button to load
the program into the computer's memory.

BUILDING YOUR CHARACTER: When the program first loads, the computer asks
you if you have a previous game stored on tape that you wish to reenter "SAVED
GAME?" If you are playing for the first time or if you did not save a previous
game on tape, respond by pressing the "N" key. (If you have a previous game
saved, see the instructions for "Loading A Saved Game". Your previous character
& associated traits will be reestablished as well.)

The computer then asks you to select your character, that is, the race you
wish to belong to and your character traits, which determine how well you
perform in combat and other situations.

Race

You will belong to one of three races: DWARF, HUMAN, or ELF. The different races have different general strengths and weaknesses, as follows:

Dwarves: can see in the dark without using magic and therefore never have to use a "LIGHT" spell (see section on non-combat spells). They are the strongest fighters, but poorest magic users.

Humans: have balance in all abilities.

Elves: are light weight so they will never spring trap doors. They also are the best magic users. Their fighting skills are below average, however.

Character Traits

When you've selected your race, the computer asks you if you want to play the game with a given set of character traits. These traits are rated numerically and generally determine your success in the dungeon. The five traits over which you have some initial control are listed below.

Strength: determines how much damage you can dish out in combat with monster. This trait is generally high in Dwarves and low in Elves.

Dexterity: determines how well you can avoid a monster's attempt to strike you.

IQ: , or your intelligence, determines how well you magic spells work. IQ is usually high in Elves and low in Dwarves.

Vigor: determines how much damage you can sustain as you attain greater skill in the game. It is higher in both Dwarves and Elves.

Hits: determine the total amount of injury you are able to withstand. As long as your Hits remain at zero or above, then you are alive, but should an encounter with a monster or pit reduce your Hits below zero, then it's curtains!

The computer displays a number associated with each trait and asks if you want to play with that set of skills. The numbers generated for Strength, Dexterity, IQ, and Vigor will range between 3 and 20. The number of Hits will vary between 2 and 12. The higher the number, the better your abilities in that area. If you're satisfied with that set of skills, press the "Y" key. If not, press "N", to generate a different, random set of values.

BUILDING THE DUNGEON: After you've selected your character & traits, you'll see the message "BUILDING THE DUNGEON" on the screen. It is at this time that the computer randomly generates the 16 x 16 x 8 dungeon layout and scatters the monsters and other items throughout the levels.

Mazes and Monsters

The dungeon is rebuilt and its layout changes under two conditions:

1) Whenever the program is loaded from tape and a "NO" response given to the "SAVED GAME?" query.

2) Whenever you use the RESET-R sequence to restart the game and choose a new character.

The dungeon is not rebuilt if your character is killed and the game restarts, even if you choose a character of a different race.

When construction of the dungeon is completed, you have two options: You can either 1) enter the maze or 2) see your game statistics. (Although the screen display indicates that you also can save the maze on tape, that option is not available to you at this point in the game. See Saving Maze on Tape for further information.)

GAME STATISTICS: If you press S to review your current status, you will see two screens of information that give you a more detailed breakdown of your standings. The first screen looks like this:

```
RACE ELF
STR 8      IQ 13
DEX 8     VIG 11

RANK 0
EXP  0
MAG  4
HITS 7/7
```

In addition to the traits you select, your character has certain other associated statistics:

Rank: this is a numerical representation of your rank, or level of expertise, in playing the game. Your rank increases as you gain experience. All characters begin with a rank of zero. As you gain in rank, your other powers, such as magic and hits, also rise.

Experience: you get experience points by playing monsters or finding certain treasures. Gold, for example, may be converted to experience points. Accumulating experience is the only way to increase your rank. Experience points and rank are related as shown below:

Experience Points	Rank
0 – 999	0
1,000 – 1,999	1
2,000 – 3,999	2
4,000 – 7,999	3
8,000 – 15,999	4
16,000 – 31,999	5
32,000 – 63,999	6
64,000 – 127,999	7
128,000 – 249,999	8
250,000 – 374,999	9
375,000 – 499,999	10

etc.

At a rank of 8 and beyond, you will attain another rank
promotion after every additional 125,000 Experience points.

Magic: you begin the game with a certain number of magic units,
determined by your race. Dwarves start with 2 units, human
3, and elves 4. Your magic powers increase by 2 units each
time your rank increases. How well you use magic, that is,
the probability of your spells succeeding is related to
your IQ.

Other information is presented in the two statistics screens, such as the
amount of gold you've found and which magic items you're carrying. These will
be discussed in more detail later.

You can examine the game statistics at almost any point in play by typing
SHIFT-S. The exception is that you cannot call your statistics to the screen
when you encounter a monster and are faced with a combat situation.

Press the 'CR' to advance to the second statistics screen. Press 'CR' again
to return to game play.

THE CASTLE: You start out from the castle. The entrance/exit is at the north-
west corner of the first dungeon level. The castle is a friendly place where
you are revitalized from your travels. When you find your way back there,
the computer asks you if you want to go in. Press "Y" to enter the castle;
press "N" to remain in the dungeon maze. If you choose to enter, your magic
and any hits you've lost are replenished. Also, any gold you're carrying is
automatically converted to experience points when you arrive there.

THE DUNGEON: Press "E" to leave the castle and enter the dungeon. As stated
earlier, the dungeon consists of a series of eight levels, each of which is
a 16 x 16 grid.

When you are in the dungeon, your surroundings appear three-dimensionally.
You have full view of what is directly ahead of you and partial view of what
is immediately left or right. If there is no wall or door ahead, you can also
see part way into the next square of the grid. The direction you are facing
appears at the top of the picture as an N, E, W, or S. Each level of the maze
is composed of the following:

Walls: which appear as plain rectangles.

Doors: which appear as a wall rectangle but with a door at its
center.

Secret Doors: which cannot generally be seen, except by dwarves, who
can see in the dark. Other races must cast a "LIGHT" spell
(see Non-Combat Spells) to see them. Secret doors appear
to be walls, but function like ordinary doors.

Openings: which are simply the absence of any of the previous three.
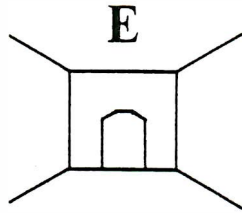
Example:

This view shows walls to your right and left and a door directly in front of you; you are facing east.

Example:
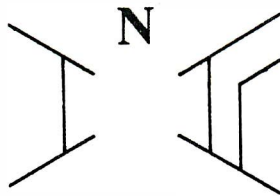
This view shows a wall to your immediate left and a door to your immediate right. Since there is an opening directly ahead, you can see the walls of the next square; you are facing north.

The entire dungeon is accessible, so if you appear to be in a passage without exit, start trying the walls for secret doors. Remember, you cannot walk through a wall!

MOVING AROUND IN THE DUNGEON: Use the following keys to move about in the maze:

    R = turn right
    L = turn left
    T = turnaround
    F = go forward into the next square via an opening only
    SHIFT-F = go forward into the next square through a door

You can also move in the dungeon by using magic spells (see Non-Combat Spells).

As you move through the mazes, you may come upon aspects of the maze other than doors and walls:

Stairs: These connect levels of the dungeon. When you encounter a flight of stairs, the computer will tell you what direction they go, up or down, and ask whether you want to follow them. If yes, you will arrive at a predetermined spot on a new level. Note that stairs may take you one or two levels up or down.

Trap Doors: If you step on one of these, you will fall down to the next level, unless you are an elf, as they do not weigh enough to spring trap doors. If you have cast a "LEVITATE" spell and step on a trap door, you are given the option of falling through it or not. Elves always have this option.

Pits: These are a nuisance factor only. Should you fall in one, your "HITS" statistic will drop by 1-6 points, possibly killing you.

The best advice for the brave adventurer is to make a map, using a sheet of graph paper. Being lost in the maze is no picnic!

MONSTERS AND COMBAT: No adventure is complete without a nemesis or two – this game has <u>fifteen</u> different kinds, all with individual identities, attributes and fighting skills. Different monsters have different strengths and advantages. A Dragon is a more formidable opponent than a Spider. Whenever you encounter a monster, you are forced to take action. You options are to fight it out, cast a magic spell, run away, or use a magic item. If you choose the fight option, you attack with your sword until either you or your adversary is slain. The spell option allows you to cast a combat spell. (see Combat Spells). Magic items you can use in combat are an amulet, a cube, a spear, and a wand. You can only use a magic item if you have one however, the effectiveness of each is explained in the section titled Treasures.

You may try to run away if any of the other possibilities seem doomed to failure. However, you cannot run out of a square that is completely closed in by walls or doors – it must have at least one open side. Your ability to escape is modified by your dexterity and some randomness and if you fail, be prepared to fight! Remember, too, you can't run from Death!

The deeper you descend in the dungeon, the fiercer and more difficult to destroy the monsters becomes. Your rank relative to the dungeon level you are on is the main factor in determining your success in combat. In general, your rank should be at least four times the number of the dungeon level you're on before you can feel secure. For example, do not descend to level 5 until your rank is at least 20. You can give it a try, but you'll likely pay for it with your life.

The table on the following page identifies the various monster types, their fighting strengths, and special strengths. Column one shows how many Hits you can expect to lose if struck in combat by the various monsters. You also gain Experience every time you kill a monster, shown in column two.

To interpret the data on the table, multiply the dungeon level you are currently on by numbers shown. If you are on level 3 and encounter a Troll, you can expect to lose 9-30 Hits in a fight, should the monster strike you. Then multiply the level by the monster's point value to see how much Experience you gain by slaying it. A close encounter with a Troll on level 3 would net you 120 points (3 x 40).

The third column in the table, Special Features, identifies any characteristics that are unique to a particular monster. The fourth column tells you about the nature of the beast. This information may assist you in deciding what kind of spell or object will be most effective in combat with a monster.

(6)

MONSTERS IN THE MAZE

Monster	Hits Lost By You (X Level You Are On)	Experience Point Value (X Level You Are On)	Special Features	Nature
DEATH	3-12	100	Can attack you even if you're invisible/can't run from	U
DEMON	4-16	150	Can attack you even if you're invisible	U
DRAGON	6-20	200	None	M
GARGOYLE	3-12	50	None	M
GHOST	2-5	75	Can attack you even if you're invisible	U
MIND FLAYER	1-4	35	Can lower your IQ	*
SERPENT	2-8	25	None	F
SLIME	2-9	60	Can eat your gold	*
SPHYNX	3-24	250	Is unaffected by type 1 or 2 magic	M
SPIDER	2-16	20	None	F
THIEF	1-4	10	Can steal your magic items	F
TROLL	3-10	40	None	F
VAMPIRE	4-18	160	Can attack you even if you're invisible/can lower your Strength	U
WARRIOR	1-8	15	None	F
WIZARD	2-9	20	None	F

U = Undead type monsters
M = Mythical type monsters
F = Flesh & Blood type monsters
* = These two monsters are unique in nature and do not fall into any specific category.

In combat, Slime may eat any gold you are carrying instead of attacking you physically; you will be notified by the message "MUNCH" should this happen. Likewise, Thieves might steal one of your magic items (if you have any); you will see the word "GOTCHA!" if this occurs. Mind Flayers and Vampires may opt to lower your IQ and Strength, respectively, by one point. This is signalled by a "ZAP!" message. Note that if either your IQ or Strength fall to zero, you are dead.

MAGIC: You are able to cast a total of twenty different magic spells. Ten of them are intended for combat use only; the other ten can assist you in moving about the mazes. There are three types of magic spells. The type of the spell determines how hard it is to cast and how many of your magic units are used up when you cast one. For example, a type 2 spell, whether in combat or non-combat, uses up 2 magic units. You can only cast a spell if you have enough magic units. You cannot cast a type 3 spell if you have only 1 magic unit available.

Non-Combat Spells

Ten non-combat spells are available to you. Although some of these spells, such as Vanish, may help you avoid combat with a monster, you can only cast them when you are not in a conflict situation. Some spells, such as Depth and Ceiling or Floor Pass, are instantaneous and last a single move only. Others, like Light and Levitate, will remain in effect over a number of moves.

If you are not in a combat situation and want to find out whether one of your five spells that have duration (Light, Shield, Levitate, Speed, or Vanish) is still working, type "M", and asterisks will appear by the spell names that are currently active. This feature erases the screen; to return to the game play, hit any key.

    Type 1 - You expend one magic unit to cast these spells.

        Depth: tells you what level of the dungeon you are on. This spell is useful if you have just gone down a flight of stairs and want to know whether you went down 1 or 2 levels.

        Heal: will replenish 1-8 Hits if you have lost any.

        Light: Dwarves don't need this one, but it lets Humans and Elves see Secret Doors and spot monsters ahead of you in the dark. The monsters will appear as shadows. The spell lasts 16-30 moves.

        Shield: will make you less vulnerable to monster attacks. It lasts 8-16 moves.

    Type 2 - You expend two magic units to cast these spells.

        Levitate: keeps you from falling into pits and through trap doors. If you do step on a trap door, you are given the option of whether you want to fall thru it or not. This spell lasts 16-30 moves.

        Speed: gives you greater dexterity in combat and helps avoid a monster's attempt to hit you. A speed spell lasts 8-16 moves.

Vanish: makes you invisible, and therefore not attackable, except by the monsters that can see you whether you are invisible or not. This spell lasts 16-30 moves. It can be very useful if your Hits are low and you want to avoid a fight.

Wall Pass: lets you go through a wall <u>directly in front of you</u>. This is an easy way out if you find yourself surrounded by monsters.

Type 3 - You expend three magic units to cast one of these spells.

Ceiling Pass: raises you up one level in the dungeon. If you are on level 3 and use this spell, you move up to level 2. If you use it on level 1 (the uppermost floor), you will be taken into the castle. The advantage of this spell is that you can move between levels without having to search for stairs.

Floor Pass: is the opposite of Ceiling Pass in that it takes you down one level. This spell fails if you attempt to use it on level 8.

Combat Spells

You can cast these spells only when you encounter a monster. Some of them are all-or-nothing spells, that is, either they kill the monster or don't. Others cause damage to the monster, possibly killing it.

If you cast a spell and it had no effect on a monster, you will get the message "SPELL FAILED". You must then fight the monster with your sword. If the spell damages the monster, but does not slay it, you will be told "MONSTER HURT". You then must fight the monster, although it attacks at a disadvantage because it is injured.

Type 1 - You expend one magic unit by casting these spells.

Hold: is an all-or-nothing spell. If it works, it will kill <u>most</u> flesh and blood type monsters, including those with mythical properties (see Monster Traits).

Missile: shoots a magic particle squarely at the attacking creature. It damages most monsters.

Sleep: is a very special spell. This one will automatically destroy weaker flesh and blood creatures on dungeon levels 1-3. The spell <u>never</u> works below level 3.

Pray: is a prayer for divine intervention and most effective on undead types.

Type 2 - You expend two magic units when you cast these spells.

Dispel: is another all-or-nothing spell. You will find it most effective on monsters that are not bound to physical bodies.

Fireball: is exactly what it sounds like. It is a powerful spell, but some monsters that are accustomed to high temperatures are unaffected by it.

Ice: is the opposite of the Fireball.

Wrack: causes terrible internal destruction to a monster. It is a very potent spell.

Type 3 - You expend three magic units when you cast either of these spells.

Phaseout: will work on any monster about 50% of the time, slaying it by banishing the creature into another dimension.

Teleport: sends your attacker off into oblivion, if it works. Should the spell backfire, however, you will be carried to a random square in the maze on the level you are currently exploring.

The effectiveness of any spell is enhanced or harmed by a combination of your Rank, IQ, race, the dungeon level you are on, and a bit of luck!

If you accidentally start the spell routine, or decide that you don't really wish to throw a spell, you can get out of it by pressing the 'CR' key.

TREASURES: Mazes and Monsters offers great rewards for the battle-weary dungeon traveller. Many times, when you kill a monster, you will be remunerated beyond just Experience points. In this section we will cover other things you might find.

Gold
By far the most common treasure is gold; it is plentiful and found in greater and greater amounts as you go deeper in the dungeon. Finding gold is a real bonus because, when you return to the castle, all gold is converted into Experience points. Gold therefore helps you advance in rank!

Magic Items
Sometimes, you may come upon a magic item. These enchanted articles will assist you during your travels. Each has its own unique characteristics.

Amulet: in combat, destroys "undead" type monsters.

Cube: destroys all monsters when used.

Spear: slays monsters of a mythical nature (dragons, gargoyles, sphynxes) when used.

Wand: kills weaker flesh and blood monsters when used.

Lamp: lets you see in the dark. It's like a light spell that never runs out.

Three other items you may find are Shields, Swords, and **Armor**. Each time you find one of these, the value of that piece of equipment increases by one. If you look at your statistics, usually your sword, shield, and armor are listed as "NORMAL". If you find a sword, your sword statistic changes to "+1". Any one of the equipment statistics can be increased up to "+3". These pluses help protect you from a monster's attacks. The greater the statistical amount, the more protection the item gives you.

Additionally, every time you find a magic item, you get an Experience bonus of 500 points.

Books: In your journey through the maze, you may also find a magic book. When you do, you are given the option of whether or not to read it. If you do not read it, the book vanishes, unread forever.

Warning! Not all books are beneficial!

Reading the book might raise one of your four primary traits (Strength, Dexterity, IQ, Vigor) by one point. But, a Book of <u>Death</u> will take away all your Hits. A <u>Cursed</u> book will teleport you to a random square in the maze on the level where you are. A book may even cause polymorphism, that is, a character mutation that changes your race!

Some books may turn out to be unreadable.

In order to experience the effect of a book, you must read it.

The Sacred Orb
Your quest is to find The Sacred Orb, and it is guarded by one of the monsters down on level 8. If you find it and return it to the castle, you win and may retire to a royal life of leisure.

SAVING THE GAME ON TAPE: You can do this after you return to the castle by pressing "W" when asked what you want to do. You can also save the game on tape while in the maze by typing a SHIFT-W. Insert a blank cassette tape in the tape drive, and follow the instructions given on the screen to store your game statistics and current maze structure onto it.

RELOADING A SAVED GAME: Every time you load the program or restart it via RESET-R, the first question you'll see is "SAVED GAME?". If you previously saved a character and maze on tape and wish to continue the game with that character, insert the tape containing the data into the tape drive and press the "Y" key. Follow the instructions for loading the data from tape.

Mazes and Monsters

When the tape has loaded, the dungeon as it previously existed is restored.
Your character has the same rank experience and other statistics as when you
saved it. Your character is placed in the exact location in the maze from
which the save game function was involved.

Because Mazes and Monsters is extremely complex and detailed, you'll find
the save game feature useful for extending game play over multiple sessions.

KEYS ACTIVE WHILE EXPLORING THE MAZE:

           R - turn right
           L - turn left
           T - turn around
           F - go forward
  SHIFT-F - go through a door
  SHIFT-S - see current game statistics
           M - see spells that are in effect
           S - cast a spell
  SHIFT-W - write your character to tape

MINDBENDER

Program Instructions   (for 1 player)

MINDBENDER is a challenging test of your deductive reasoning powers.
The object of the game is to break the computer's secret code--four
numbers chosen by the computer--in as few moves as possible. You
not only have to guess the correct numbers, but put them in the correct
sequence as well, given clues about your previous guesses. WARNING:
MINDBENDER may be habit-forming!

LOADING THE PROGRAM: Insert the program cassette into the cassette
deck, depress the RESET button, depress the "L" key, rewind the tape
if necessary, the depress the READ cassette button to load the program
into the computer.

GAME PLAY: After the program loads, the computer selects four numbers
and asks you to guess them and put them in the correct sequence.
It then presents the game board: a row of nine numbers (1-9) on the
bottom of the screen, a group of four red squares in the center of
the screen that mask the secret code, and a "scoreboard" at the top
of the screen that gives clues about your guesses. Press the joystick
lever to the left or right to move the selection arrow under the number
you want to guess. Then depress the hit button to place that number
in the first of the four red squares. Repeat this sequence to select
the second, third, and fourth numbers in the sequence. After you
have selected the fourth number, the computer "scores" your guess
on the "scoreboard" with two numbers separated by a colon. The number
to the left of the colon indicates how many of the numbers in your
guess are both correct and in the right place in the sequence. The
number to the right of the colon indicates how many numbers in your
guess are correct, but in the wrong place in the sequence.

At this time you may elect to see your previous guesses and their
"scores" for analysis, or to move to your next guess without seeing
them. To see your previous guesses, move the joystick lever to the
"up" position. (Note: tap the lever upwards only, do not hold the
lever in the up position.) To proceed with your next move after reviewing
your previous guesses, press the joystick lever up again. If you
do not want to see your former tries, simply depress the joystick
hit button to move on to your next guess.

When you have guessed all four numbers in the correct order, 4:0 shows
on the "scoreboard". The computer then prints the number of tries
it took you to break the secret code.

NEW GAME: When the secret code has been broken, a new game starts
automatically.

MICRO *M* VIDEO

TATOOINE TRAVELS

A Microsoft 8K BASIC
Adventure Game

It's adventure on the desert planet of Tatooine! Your destination is the space-port at Mos Eisly. To get there, you must travel across 200 miles of hot desert sands on your giant lizard, overcoming hazards you encounter along the way. The sand people are continually in pursuit of you, and you may have to do battle with Jawas or Imperial troops. Use your lizard's travelling speed and water rations wisely, or you may end up as vulture food. No two games are ever the same. Rated absolutely addicting by our reviewers.

LOADING THE PROGRAM: Insert the Microsoft 8K BASIC cassette into the tape deck. Depress the RESET button, press the 'L' key, rewind the tape if necessary, then depress the READ cassette button to load the BASIC interpreter into the computer. When the BASIC 'OK' prompt appears, remove the BASIC cassette and insert the Tatooine Travels cassette in its place. Repeat the RESET-L sequence described above to load the program. The program will execute automatically when it has loaded.

GAME OPTIONS: When the program starts, you are asked to select a skill level for play. Your choices are "A=TENDERFOOT, B=CONFIDENT, C=DESERT FOX". This selection controls several variables: your lizard's maximum energy level, the number of drinks you can carry in your canteen, and the speed at which the Sand People pursue you. Press key A, B, or C to select an easy, average, or hard game, respectively.

GAME PLAY: Tatooine Travels is a command-driven game. You tell the program what you want to do by selecting from six available commands. The command list is:

        D = DRINK
        F = FULL SPEED
        H = ½ SPEED
        R = REST FOR THE NIGHT
        S = STATUS CHECK
        C = COMMAND LIST

The effects of these commands and their uses are as follows:

D = DRINK -- Press the 'D' key to take a drink when it is necessary to do so. You must take a drink from your canteen at least every fourth turn. The computer will warn you when you must drink.

F = FULL SPEED -- Press the 'F' key to travel at full speed. Your·lizard burns
    3 energy units in each turn at full speed and will carry you between 5
    and 20 kilometers.

H = ½ SPEED -- Press the 'H' key to travel at half speed. Your lizard burns
    only 1 energy unit in each turn at half speed, and will carry you between
    1 and 10 kilometers.

R = REST FOR NIGHT -- Press the 'R' key to stop and rest for the night. After
    a good night's sleep, your lizard's energy level is restored to full capacity.
    Stopping to rest counts as a turn (you may be forced to take a drink after-
    wards).

S = STATUS CHECK -- Press the 'S' key to check your lizard's energy level,
    how many drinks left in your canteen, and how many turns you can take without
    drinking. A status check does not count as a turn.

C = COMMAND LIST -- At any time, you can display the command list by pressing
    the 'C' key. This does not count as a turn.

HAZARDS: During your travels on Tatooine, you may encounter hazards,. such as
sand storms, Sand People, Jawas, and Imperial troops, that threaten to impede
your progress. In some cases, you will have to make a decision as to what
to do; follow the computer's instructions if this occurs.

MOISTURE FARM:. If you reach a moisture farm, your water supply is automatically
replenished to maximum level. You also take advantage of the abundance of
water and take a drink, which resets your turns without drinking to the maximum
of 4.

Good luck, you'll need it. Have a safe trip!

PROGRAM LISTING: A complete program listing for Tatooine Travels is given
on the following pages. You can use this listing to make modifications in
the program or to learn more about this type of game programming in BASIC.

```
5 REM TATOOINE TRAVELS
8 REM COPYRIGHT 1981, MICRO VIDEO CORP.
9 CLS:PRINTCHR$(8)
11 CLEAR(2):COLOR0,3,4,7:H=1:R=1:J=RND(-PEEK(24559))
14 CLS:PRINT"WELCOME TO THE    DESERT WORLD OF  TATOOINE--":PRINT:PRINT"GOOD LUCK
  ":PRINT
17 GOSUB227:CLS
20 PRINT"SKILL LEVEL":PRINT:PRINT"A= TENDERFOOT"
23 PRINT:PRINT"B= CONFIDENT":PRINT:PRINT"C= DESERT FOX":TONE25,200
26 C$=INSTR$(1)
29 IFC$="A"THENL=12:DR=9:SS=10:GOTO41
32 IFC$="B"THENL=9:DR=7:SS=12:GOTO41
35 IFC$="C"THENL=7:DR=5:SS=15:GOTO41
38 PRINT"HUH?!":PRINT:TONE25,200:C$=INSTR$(1):GOTO29
41 CLS:PRINT"TYPE 'C' TO SEE  COMMAND LIST, ANYOTHER KEY TO     START"
44 TONE25,200:C$=INSTR$(1):IFC$="C"THENGOSUB212
47 WINDOW60:LX=L:DX=DR:SD=0
50 PLOT0,0,0,112,77:PLOT0,61,2,112,17
53 OUTPUT"YOUVE GONE",5,67,3
56 PLOT70,62,0,25,6:OUTPUTD,70,67,3:OUTPUT"K",98,67,3:IFD>=200GOTO446
59 IFD<>0THENR=0
62 IFD>=20THENGOSUB158:IFLS=1GOTO404
65 IFH=0THENH=1:GOTO71
68 LS=0:GOTO236
71 IFLS=1GOTO404
74 IFT=3THENGOSUB338:PRINT"GET A DRINK!":PRINT

77 IFLS<>1THENPRINT:PRINT"COMMAND?":PRINT:C$=INSTR$(1)
78 OUTPUTC$,55,23,1
80 IFC$="D"GOTO230
83 IFC$="C"THEN GOSUB212:GOTO77
86 T=T+1
89 IFT>4THENPRINT"YOU DRIED AND     SHRIVELLED UP!":PRINT:GOTO404
92 IFC$="F"GOTO107
95 IFC$="H"GOTO137
98 IFC$="R"GOTO179
101 IFC$="S"GOTO185
104 T=T-1:PRINT"HUH?!":PRINT:GOTO77
107 L=L-3
110 IFL>=0GOTO122
113 PRINT"YOU RAN YOUR LIZ-ARD TO DEATH...":PRINT:PRINT"HIS AND YOURS!"
116 GOSUB227
119 LS=1:GOTO71
122 PRINT:PRINT"YOUR LIZARD BURNSACROSS THE":PRINT"TATOOINE SANDS..."
125 PRINT:GOSUB227
128 D1=INT(RND(1)*21)
131 IFD1<5GOTO128
134 D=D+D1:GOTO53
137 L=L-1:IFL>=0GOTO146
140 PRINT"YOUR LIZARD JUST DROPPED DEAD!":PRINT
143 GOSUB227:GOTO404
146 PRINT:PRINT"YOUR LIZARD IS    MERRILY TROTTING ALONG...":GOSUB227
149 PRINT
152 D1=INT(RND(1)*11):IFD1<1GOTO152
155 D=D+D1:GOTO53
158 SP=INT(SS*RND(1)+2.5)
161 SD=SD+SP
164 IFSD>=DGOTO173
167 PRINT"THE SAND PEOPLE  ARE";D-SD;"KILOS"
170 PRINT"BEHIND YOU":PRINT:RETURN
173 LS=1:PLOT0,0,0,112,60:PRINT"THE SAND PEOPLE  CAUGHT YOU!":PRINT
176 PRINT"THEY LOVE LIZARD AND HUMAN STEW!!":PRINT:GOSUB227:RETURN
179 PRINT"YOUR LIZARD":PRINT"THANKS YOU!":PRINT
182 R=1:GOSUB227:L=LX:GOTO62
185 PLOT0,0,0,112,60
188 PRINT"YOUR LIZARD HAS"
191 PRINTL;"ENERGY UNITS":PRINT
194 PRINT"YOU HAVE";DR;
197 PRINT"DRINKS LEFT":PRINT
200 T=T-1
203 PRINT"YOU HAVE";4-T;"TURNS"
206 PRINT"WITHOUT DRINKING"
209 FORI=1TO2000:NEXT:PLOT0,0,0,112,60:GOTO71
212 PLOT0,0,0,112,60:PRINT"COMMANDS:":PRINT
215 PRINT"D=DRINK"
218 PRINT"F=FULL SPEED":PRINT"H=1/2 SPEED"
221 PRINT"R=REST FOR NIGHT":PRINT"S=STATUS CHECK"
224 PRINT"C=COMMAND LIST":FORI=1TO3000:NEXT:PLOT0,0,0,112,60:RETURN
227 FORI=1TO1000:NEXT:RETURN
230 DR=DR-1:IFDR<0THENPRINT"!!OUT OF WATER!!":GOTO404
231 T=0:PRINT"GULP!":PRINT
233 IFDR<=DX/2THENPRINT"BETTER WATCH FOR A MOISTURE FARM!":T=0:LS=0:GOTO71
236 H=RND(1)*100
239 IFH>95GOTO350
242 IFH>18GOTO254
```

```
245 IFR=1THENR=0:GOTO254
248 PRINT"YOU HAVE REACHED A MOISTURE FARM!":PRINT:DR=DX:T=0
251 GOSUB227
254 IFH>30ANDH<35THENGOSUB263:IFLS=1GOTO404
257 IFH>20ANDH<27THENGOSUB338:GOSUB311:H=0:R=0:GOTO53
260 LS=0:GOTO71
263 PLOT0,0,0,112,60:PRINT"THE JAWAS HAVE    CAUGHT YOU!"
266 PRINT:PRINT"YOU MANAGED TO    RADIO FOR HELP":PRINT
269 GOSUB227
272 PRINT"YOU HAVE A CHOICEOF COMMANDS:":PRINT
275 PRINT"A: WAIT FOR HELP":PRINT:PRINT"B: TRY ESCAPE"
278 PRINT:C$=INSTR$(1)
281 IF C$<>"A"ANDC$<>"B"THENPRINT"HUH?!":PRINT:GOTO278
284 IFC$="B"GOTO293
287 IFRND(1)<.3THENPRINT:PRINT"SO SORRY,NO HELP!":LS=1:RETURN
290 PRINT:PRINT"HELP ARRIVES, YOUARE FREED!":PRINT:RETURN
293 IFRND(1)>.3GOTO305
296 PRINT:PRINT"YOU WERE CRUSHED BY A RENEGADE    DROID--"
299 LS=1
302 PRINT:PRINT"YOU DIDN'T STAND A CHANCE, SUCKER!":GOSUB227:RETURN
305 PRINT:PRINT"MAZELTOV!":GOSUB227
308 PRINT:PRINT"A SUCCESSFUL":PRINT"ESCAPE!!":PRINT:RETURN
311 PRINT"!!SANDSTORM!!":PRINT:PRINT"LOTSA LUCK!"
314 GOSUB227:D1=INT(RND(1)*11)
317 IFD1=0THENGOSUB227:PRINT:PRINT"NO CHANGE":PRINT:RETURN
320 IFH<23GOTO332
323 PRINT
326 D=D-D1:PRINT"YOU HAVE BEEN     DRIVEN BACK BY"
329 PRINTD1:PRINT"KILOMETERS!":GOSUB227:PRINT:RETURN
332 D=D+D1:PRINT:PRINT"YOU HAVE BEEN":PRINT"DRIVEN";D1
335 PRINT"KILOMETERS AHEAD!":GOSUB227:PRINT:RETURN
338 SOUND3,30:SC=24888:PLOT0,0,0,112,60
341 OUTPUT"...WARNING...",12,12,1:POKESC,1
344 FORI=1TO5
347 PRINT:NEXT:POKESC,32:PRINT:PRINT:SOUND7,4096:RETURN
350 PRINT:PRINT"YOU HAVE SPOTTED IMPERIAL TROOPS  HEADED YOUR WAY"
353 PRINT:PRINT"YOU MUST CHOOSE:":PRINT
356 PRINT"A: WAIT IN AMBUSH":PRINT:PRINT"B: TRY TO HIDE"
359 TONE25,200:C$=INSTR$(1)
362 IFC$="B"GOTO389
365 PLOT0,0,0,112,60:OUTPUT"AMBUSH!",32,27,1
368 SOUND3,264
371 FORI=1TO10
374 COLOR3,7,2,4:COLOR4,6,8,2:COLOR6,0,2,3:COLOR5,3,2,6
377 SOUND5,392:COLOR5,6,0,3:COLOR3,4,5,1:COLOR0,3,4,7:NEXT
380 SOUND7,4096:PLOT0,0,0,112,60
383 IFRND(1)>.75THENPRINT"YOU WERE TOTALLY VAPORIZED":GOSUB227:GOTO404
386 PRINT"YOU INCINERATED  THEM!!":GOTO71
389 IFRND(1)<.5GOTO398
392 PRINT:PRINT"YOU ESCAPED! THEYPASSED YOU BY!"
395 GOSUB227:PRINT:PRINT"SOMETIMES COWARD-ICE PAYS OFF!!":PRINT:GOSUB227:GOTO71
398 PRINT:PRINT"THE STORMTROOPERSGOT YOU!":GOSUB227
401 PRINT:PRINT"YOU'VE HAD IT,    CHUMP!!":GOSUB227
404 WINDOW77:PLOT0,0,0,112,77:R=RND(1)*10
407 PRINT"YOU DIED IN THE  DESERT LIKE THE  WITLESS FOOL YOU ARE!":PRINT
410 GOSUB227
413 IFR<6GOTO419
416 PRINT"STUPID PEOPLE":PRINT"SHOULD STAY OUT  OF THE DESERT!!":GOTO431

419 IFR<3GOTO425
422 PRINT"JAWAS USE YOUR    SKULL FOR A BEER STEIN!":GOTO431
425 PRINT"YOUR WORTHLESS    REMAINS ARE EATENBY RAVENOUS SAND PEOPLE!"
428 GOSUB227
431 PRINT:PRINT"PLAY AGAIN?":C$=INSTR$(1)
434 IFC$="N"THENCLS:OUTPUT"COWARD!",31,38,1:END
437 IFC$="Y"THENCLS:H=1:R=1:T=0:D=0:LS=0:GOTO20
443 GOTO431
446 WINDOW77:CLS:PRINT"CONGRATULATIONS! YOU HAVE ARRIVED AT MOS EISLY"
449 PRINT"SPACEPORT..."
452 GOSUB227:PRINT"YOUR TROUBLES":PRINT"HAVE JUST BEGUN!!":PRINT:GOSUB227
455 GOTO431
```

MICRO *MV* VIDEO™

THINKCARDS

(for 1 player)

Like traditional "flashcards," THINKCARDS allows you to test
your memory by providing a "clue" item from one category and
asking you to supply the correct item from a corresponding
category. Unlike "flashcards," your subject matter is not
restricted to what is printed on cards. You program THINKCARDS
to include only the information you need to study--on subjects
such as math, vocabulary, geography, medicine, chemistry.

LOADING THE PROGRAM: Insert the program cassette into the tape
deck, depress the RESET button, press the "L" key, depress the
REWIND cassette button to rewind the tape, and then depress
the READ button to load the program into the computer.

PROGRAM OPTIONS: When the program is ready, the computer asks
"WHAT NEXT?" and presents a program "menu." Type "I" to define
a new set of pairs or to change old pairs; type "L" to list
the pairs you've entered; type "R" to read pairs from tape;
type "W" to save pairs on tape; type "D" to drill yourself on
the pairs you've entered. Type a CONTROL/S (depress the CONTROL
and "S" keys simultaneously) to stop execution of any program
option and return to the program menu.

DEFINING NEW PAIRS: When you type "I", the computer asks "ERASE
OLD PAIRS? Y=YES, N=NO." Type "Y" to erase pairs currently in
memory. Type "N" to correct or add to existing pairs (see below).
When you type "Y", the computer asks you to "ENTER CATEGORY 1
TITLE." Type in a descriptive title and finish by pressing the
"CR" key. The computer then asks "ENTER CATEGORY 2 TITLE". Type
the title for the second, corresponding category and press "CR"
when done. The computer then asks for entry of the pairs, using
the category titles you've defined. Enter the category 1 "clue"
item for the first pair and finish by pressing the "CR" key.
Then enter the category 2 "clue/answer" item that corresponds
to the clue item in category 1 to complete the first pair, and
press "CR". In DRILL mode, you select one of the categories
for "clue" items. The computer selects randomly from the items
in that category and asks you for the item from the corresponding
category that completes the pair. You can supply more than one
item as a correct response for use in DRILL mode by separating
additional items with commas. The computer continues to prompt
you for up to 100 pairs until you type a CONTROL/S to leave
INPUT PAIRS mode.

For example, if you want to set up pairs to drill yourself on states and their capitals, you might enter "STATE" as the category 1 title and "CAPITAL CITY" as the category 2 title. Then, when the computer prompts "ENTER STATE," you might respond with "MICHIGAN". The computer then prompts "ENTER CAPITAL CITY," and you would type "LANSING".

LISTING PAIRS: To list pairs for review, type "L". The computer prints the two category titles you defined and then the clue items for each category in the first pair. Type any key to display the next pair in the list. After the last pair has been displayed, the computer automatically returns to the program menu. Type CONTROL/S to stop LIST PAIRS mode before all pairs have been displayed.

CORRECTING OR ADDING TO OLD PAIRS: Type "I" for INPUT PAIRS mode, then type "N" when the computer asks "ERASE OLD PAIRS?" The computer prints the category 1 clue item of the first pair and waits for you to enter changes. Type in the corrected clue item (note that you must type the entire line again) or simply press the "CR" key to leave that item unchanged. The computer then prints the category 2 clue item for the first pair, and so on. After the last category 2 clue item has been presented for correction, the computer will ask for additional pairs by prompting "ENTER category 1" as described in "DEFINING NEW PAIRS". Type a CONTROL/S to stop INPUT PAIRS mode when all desired corrections or additions have been made.

DRILLING: To test yourself on the pairs you've entered from the keyboard or read from tape, type "D" for DRILL mode. The computer asks "USE WHICH CATEGORY FOR PROMPT?" and prints the titles of the two categories. Type "1" to use items in category 1 as the clues, or "2" to use items from category 2. The computer then asks "MAXIMUM POINTS PER QUESTION? (1-9)". Enter a number between 1 and 9, inclusive. This is the maximum number of points you will be awarded for a correct answer. The computer deducts one point for each incorrect answer, but you will always score at least one point for a correct response, no matter how many times you've answered incorrectly. The computer then selects clues randomly from the category designated for prompts. It prints the clue category title, the category clue item, and the other category title, and waits for you to enter the item that matches the given clue and completes the pair. (Note that the computer will accept more than one response as correct if you entered more than one clue item for the category and pair in INPUT PAIRS mode.) If your answer is correct, the computer prints "CORRECT", the number of points awarded, and a running total of your score. If your answer is wrong, the computer prints "INCORRECT" and shows the correct answer. (Note that spelling is important, but spacing between words and letters is not.) The computer randomly repeats any pairs you missed until you have completed all pairs correctly. It then prints "FINAL", the number of points you scored out of the total points possible, and your score as a percentage.

SAVING PAIRS ON TAPE: To save your categories and pairs on tape,
type "W". The computer prints "INSERT TAPE, PUSH REWIND, PRESS
ANY KEY WHEN READY." Remove the THINKCARDS program cassette,
insert a data tape, depress the REWIND cassette button, and press
any key to rewind the tape. (If you do not wish to rewind the
tape, follow the steps above, but do not depress the REWIND
button.) The computer prints "REWINDING TAPE, PRESS ANY KEY
WHEN READY." When the tape is rewound, type any key. The
computer than asks for a name to assign the set of pairs ("PAIRS
NAME?"). To save the pairs with a pairs name, type in a name
(up to 16 characters). This feature allows you to save more than
one set of pairs on a single data tape and to recall them by set
name. If you do not want to save the pairs with a name, simply
press the "CR" key. The computer then instructs you to "PUSH
READ AND WRITE, PRESS ANY KEY WHEN READY." Depress the READ and
WRITE cassette buttons simultaneously, then press any key. When
you do, the computer prints "WRITING TAPE." The tape has been
written when the program menu reappears on the screen. Note that
if you rewind the tape, the new set of apirs overwrite anything
currently saved on the tape.

READING PAIRS FROM TAPE: To read in categories and pairs from
tape, type "R". Remove the program cassette, insert the data
tape containing the pairs you want to enter, and rewind as
described for "SAVING PAIRS ON TAPE." The computer asks you for
the name you assigned to the pairs when you saved them on tape
("PAIRS NAME?"). Enter the name of the set of pairs you want to
read into the program. If you do not enter a name, the computer
reads in the first set of pairs it finds on the tape. The
computer then instructs you to "PUSH READ, PRESS ANY KEY WHEN
READY" to load the pairs set from the tape. The message "READING
TAPE" means the contents of the tape are being read into memory,
and the loading process is complete when the program menu
reappears on the screen. You can then add to, change, list, or
drill with the newly input pairs.

THE TROLL HOLE ADVENTURE

A Puzzling Game for 1 Player

Journey through the Troll's cave and find all the stolen booty he has hidden
there. You'll face many dangers as you wander through the underground
maze, trying to discover its secrets. Can you ge the priceless ruby from
the deadly King Cobra? Can you determine what the strange inscription
means, or why evil eyes watch your every move? Your computer is your eyes
and ears in this challenging puzzle--you command it to move through the
passages and manipulate objects it finds in this strange world. You may
come out rich...if you make it out alive!

LOADING THE PROGRAM: Insert the program cassette into the tape drive.
Depress the RESET button, press the "L" key, rewind the tape if necessary,
then depress the READ cassette button to load the program into your computer.

PLAYING THE GAME: When the program first loads, the computer asks you
if you have a partially completed game on tape that you want to enter
("SAVED GAME?"). If you have saved a game on tape and want to reenter
it, see "RE-ENTERING SAVED GAME" in these instructions. If you have not
saved a previous game, type "N" to proceed.

On the top of your screen, the computer tells you where you are and in
what directions you can obviously move. For example, you might see

        BEDROOM
        N,E,U,D

There are six directions in which you can move. They are north (N), south
(S), east (E), west (W), up (U), and down (D).

On the other part of your screen, the computer tells you what it sees and
asks you for instructions ("WHAT NOW?"). For example, you might see

        BENCH
        CUP
        ARCHWAY

        WHAT NOW?

At the WHAT NOW? prompt, you enter instructions you want your computer
to perform. In addition to the direction codes, you can instruct your
computer with simple one or two word commands. For example, you might
say

        WHAT NOW? TAKE CUP

Conclude any instruction to your computer by pressing the "CR" key. If
you type in something the computer does not understand, it will respond

> WHAT?
> WHAT NOW?

Sometimes you can move to a specific place by typing GO and the place.
For example,

> WHAT NOW? GO ARCHWAY

Some room descriptions may scroll off the screen. If this happens and
you want to review the room and its contents again, type

> WHAT NOW? LOOK

and the computer will repeat the information for you. If you want to review
what you are carrying, type "I" for inventory. To look more closely at
an item you find, use the EXAMINE command.

> WHAT NOW? EXAMINE BENCH
>
> ITS VERY HEAVY

Treasures are identified by a plus sign (+) preceding them. For example,

> WHAT NOW? LOOK
>
> +DIAMONDS
> TABLE

At any time, you can find out how many moves you have taken and what your
current score is by typing "SCORE".

The computer reads only the first three characters of each command word,
so you can abbreviate your commands if you wish. For example, it understands

> WHAT NOW? EXA BEN

as an instruction to examine the bench. Your computer knows over 100 words.
If it does not understand something you say, try another word. Part of
the fun of ADVENTURE is discovering how to communicate with your computer!
Some of the words it knows are: LOOK, EXAMINE, DROP, GET, GO, and READ.

SAVING PARTIALLY COMPLETED GAMES: Because ADVENTURE is not a game you
complete quickly, it contains an option to save your partially completed
game on tape. To use this option, type the "SAVE" command. You will then
see a read-write screen displayed. Remove the Adventure cassette and insert
a blank data tape. Then depress the READ and WRITE cassette buttons and
press the "W" key. Note that the Interact does not have an erase head,
so you must not save a game over a previous game or you will lose both.
Before hitting the "W" key to save the game on tape, you can press the
"S" key to turn on the tape motor. Then depress the READ cassette button
to position your tape past any material existing on the tape before you
continue the writing sequence. When you have saved your game on tape,

can press the "CR" key to continue the existing game. If you accidentally
hit reset, you can resume the game by answering "Y" to the CONTINUE? prompt.

RE-ENTERING A PARTIALLY COMPLETED GAME: When you load the ADVENTURE program,
the first question you are asked is "SAVED GAME?" To reload a partially
completed game, answer yes ("Y") to this query. The computer will display
the read-write screen. Remove the ADVENTURE cassette from the tape drive
and insert the cassette containing your game. Use the "S" key to turn
on the tape motor and position the tape appropriately. Then depress the
READ cassette button and press the "R" key to read in your saved game.
The computer will then ask you "CONTINUE?" Type "Y" to begin the game
where you left off.

MORE ABOUT ADVENTURE: If you want to learn more about ADVENTURE in general,
we suggest you consult the following references:

    SOFTSIDE MAGAZINE. Special ADVENTURE issue, July 1980

    CREATIVE COMPUTING. Special ADVENTURE issue, July 1980

    RECREATIONAL COMPUTING. Fantasy simulation games issue, July-August 1980

    BYTE. Special ADVENTURE issue, December 1980

HELP!!!: At any time during your journey, you can type HELP to get a clue
from your computer. Sometimes these clues are helpful, sometimes they
are not. In general, try looking around, examining things, and trying
different courses of action. If you can't get past something, go back
and see what objects you find. Try the logical first, and if that doesn't
work, try the odd or unusual. The puzzles are difficult, but a 13-year-
old has successfully solved them.

And, if you ever get absolutely stuck and can't go any further, you have
one last resort. Send us a note describing how far you have gotten and
what your problem is (if you know it). Include a self-addressed, stamped
envelope, and we will send you back a clue to help you along.

GOOD LUCK, AND HAPPY ADVENTURING!!!

MICRO ⚡ VIDEO

305 North First St.
P.O. Box 7357
Ann Arbor, MI 48107
(313) 996-0626

PACK WARS

An Action/Arcade Game for 1 or 2 Players

A new member of the family Eateus Dadotteus has been discovered, and it's a lot more aggressive than its relatives! If you like to eat and run, Pack Wars is for you. Fight a friend or the computer in a consuming battle for your next square meal. Hours of ravenous run are in store for gluttons of all ages!

LOADING THE PROGRAM: Insert the Pack Wars cassette into the tape drive. Depress the RESET button, and press the "L" key. Rewind the tape if necessary, then depress the READ cassette button to load the program into the computer.

GAME OPTIONS: Once Pack Wars has loaded into the computer, the screen displays "GAME TYPE? 2-PLAYER, S=SLOW ROBOT, F=FAST ROBOT". If you wish to play against another player, type "2" and plug in both controllers. If you want to play against the computer, you have the option of facing the Slow Robot or the Fast Robot. Type "S" for slow or "F" for fast and plug in the left controller. When you've made your selection, the maze appears on the screen, and the game begins.

GAME PLAY: If you choose the two-player option, the object is to devour all your color squares in the maze before your opponent eats his (or hers!). Each maze constitutes one battle. The Pack War is over after a total of five battles. The left player is red; the right green. It's impossible to eat the wrong color. You can blast your opponent and send him or her hurtling through hyperspace into another part of the maze by pressing the fire button, your "photon gun". This feature adds an interesting element of strategy to the game, as it's possible to win the Pack War while losing all the individual battles if you are particularly proficient at ambushing your opponent with your photon gun.

If you choose to play against the computer, you have two options -- Slow Robot or Fast Robot. The "Slow Robot" moves the same speed as you do, the "Fast Robot" is just that--FAST--and he moves at twice your speed. He's not a good opponent for the easily flustered. Your objective in these games is slightly different than in the two-player option. The idea is to gobble down all 160 red squares in each of five mazes before the Robot tracks you down. As you race around the maze you leave a green "trail". Like a good hunting dog, the Robot sniffs out and follows your scent whenever he finds it. You can befuddle your pursuer by leaving crossing trails. If he chooses the wrong trail he'll dart off in the wrong direction--but not for long. If he chooses the right trail and finds you, he'll consume you quickly and you die. Immediately the next maze appears on the screen. As in the two-player mode, your only defense is your photon gun.

GAME SCORING: For each square you consume, you score 10 points. In the
two-player version there are 160 sqaures of each color on the screen.
Therefore, you can score a total of 1600 points per screen on square meals
alone! In addition, you'll earn 100 points each time your opponent takes
a photon in the gut. Wolf down all your squares first in each battle round,
and you'll score an additional 500 points. The Robot version is similar
to the two-player version, except that only you have squares to eat and
there is no 500 point bonus for eating all of them. When playing against
the computer, your score is displayed in the bottom center of the screen. In
the two-player version, your running score is displayed at the bottom
of the screen, next to the color of the squares you are eating. In the
two-player mode, the battle number also appears at the bottom of the screen
between the two scores. If you are playing against the computer, your
battle victories are shown on the left and the Robot's are on the right.
The person with the most total points wins the "Pack War".

GAME END: The game ends after five battles in the two-player version.
When you play against the computer, the game ends after you or the computer
win five battles, so it is possible to play up to nine battles in one
Pack War. When the game ends, the prompt "SAME GAME?" will appear. Press
"Y" or the fire button to play again, or "N" to play again with different
game options.

MICRO *MV* VIDEO

SILVER DRAGON

An Action Fantasy Game for 1 Player

For hundreds of years, everything was peaceful in the Scarlet Mountains.
Until recently, the 10 beautiful Sylphs have lived at the base of the
mountains protected by you, the gleaming, firebreathing, Silver Dragon.
Now a band of evil genies has gathered belligerent creatures from all
over the continent to invade your mountains. The whirlwind genies and
deadly manticores want to destroy you, while the giant eagles swwop down
and carry your Sylphs away. If you can rescue her in time, your Sylph
won't mutate with the giant eagle to form a hideous harpy, a third creature
that uses its power to destroy you.

The challenge is clear--use your skill to destroy the intruders, protect
the Sylphs, and gain the experience necessary to reach the exulted position
of "High Level" Silver Dragon.

LOADING THE PROGRAM: Insert the program cassette into the tape drive.
Depress the RESET button, then press the "L" key. Rewind the tape if neces-
sary, then depress the READ cassette button to load the program into the
computer's memory. Plug in the left entertainment controller.

GAME OPTIONS: When the program has loaded the computer asks you to select
an experience level with the prompt:

> CHOOSE LEVEL
>
> NOVICE
> INTERMEDIATE
> EXPERT

Press "N" for novice, "I" for intermediate, or "E" for expert. Next you'll
see the prompt "HOW MANY GENIES (1-5)?". Press a numerical key between
one and five indicating the number of whirlwind genies that you wish to
be in the first attack wave. In novice play, one giant eagle and one manti-
core accompany each genie at the beginning of an attack wave. Two eagles
and one manticore join each genie in intermediate play. In expert play,
two eagles and two manticores are allied with each genie.

GAME PLAY: After you have selected the game options, the game screen
will appear, with mountain ridges in the background. The Silver Dragon
begins at the bottom center of the screen. Sylphs also move along the
base of the mountains. The first monster attack begins immediately. The
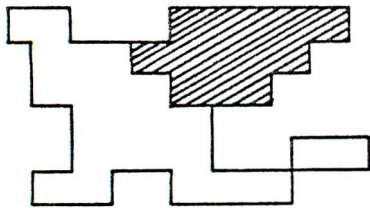various game creatures and their characteristics are as follows:

Color Key

white	red	yellow

FRIENDLY CREATURES

THE SILVER DRAGON: The Dragon's movements correspond to the direction you move the joystick. You move it up and down in the center of the screen when you push the joystick lever forward or pull it backward. You can change flight direction, shown by mountains scrolling across the screen, by moving the joystick lever left or right. When you press the fire button, the dragon shoots a fireball. It will shoot them continuously if you hold down the button. The fireball follows a direct horizontal path in the direction of the dragon's flight. If your fireball hits a monster, the creature is destroyed, you earn bonus points, and the game pace increases slightly. The dragon can also destroy all monsters, except the harpy, by colliding with them. However, this strategy is highly inadvisable, since the dragon is also destroyed. You begin the game with four regenerating potions--each time you are destroyed, you use one of them. The bottles of potion remaining are displayed in the lower right corner of the screen. For each experience level you reach, you are awarded one more bottle of potion. (You may actually possess more potions than are displayed on the screen.) Each time the dragon is regenerated, it reappears at its original starting point--the bottom center of the screen.

The Silver Dragon
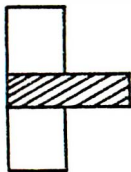
SYLPHS: These are beautiful creatures, similar to nymphs. They move along the base of the mountains at regular intervals. As you reach each new experience level, all missing Sylphs are regenerated. If all 10 are carried away or destroyed, the mountains stop scrolling and gradually disappear, and the game pace increases substantially as punishment for your negligence. The Giant Eagles are the Sylphs' only direct enemies. They can mutate with the Sylphs into hideous harpies, who then become enemies of the Silver Dragon.
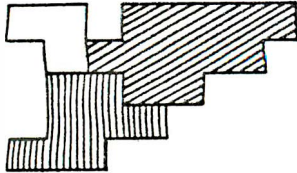
Sylph

BELLIGERENT CREATURES
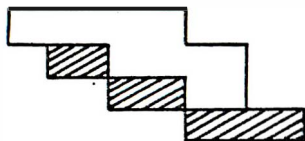
GENIES: Each genie appears as a whirlwind. Genies are the smallest and fastest targets on the screen. Genies also have the power to become invisible for short periods of time; they use this power to flash on and off, making them more elusive targets.

Genie
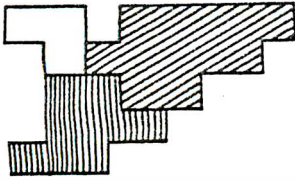
Giant Eagle

Giant Eagle carrying Sylph

Harpy

Manticore

GIANT EAGLE: The eagles have the power to abduct the Sylphs and mutate with them to form hideous harpies. Each eagle flies horizontally across the screen, then dives when it spots a Sylph. When it reaches the Sylph, it picks her up and flies slowly upward. They will mutate into a harpy only if you allow them to reach the top of the screen.

If you mistakenly hit a Sylph with a fireball while it is being carried away, it will be destroyed and the eagle will remain intact. On the other hand, if you hit the eagle while it is carrying a sylph, it will be destroyed and the unconscious Sylph will fall to the earth. If the Sylph hits the ground she will die on impact. However, you can earn hundreds of bonus points by catching her in mid-air with the Dragon's claws and returning her gently to the ground. The dragon can carry more than one Sylph, but he has to set each down separately. Death of the dragon results in the destruction of all sylphs in its claws.
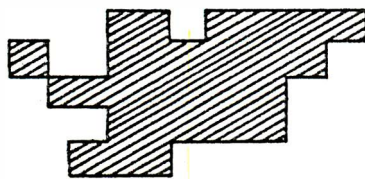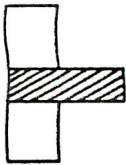
HARPY: The harpy is a despicable creature, with the lower body and wings of an Eagle, and the upper torso and head of a woman. Formed by the mutation of an eagle and a Sylph, a harpy appears as a solid red object on the screen. It has a powerful "magnetic" personality--if the dragon comes within half the body length of the harpy, it will fall prey to the magnetic field and be destroyed.
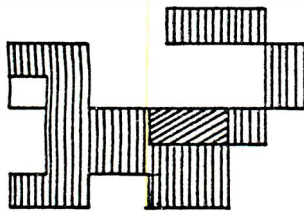
MANTICORE: The manticore has the body of a lion, the wings of a bat, and the face of a man. He has deadly spikes at the end of his tail and can throw them in any direction. Short beeps signal the release of a spike, which appears as a solid white object. It is wise to avoid being caught directly in front of this enemy--always attack from below the level of its raised tail or from the rear. Most of the time the manticore hurls its tail spikes in the direction of its flight.

SCORING: Your current score and experience level are displayed in the lower left corner of the screen. You earn 300 points for each creature that is destroyed, whether by your fireball, a manticore's tail spike, or a collision. You score 700 points for catching a falling Sylph, and an additional 700 points for returning her safely to the ground.

You move up in levels by reaching certain scores. In novice play you increase a level every 10,000 points. In intermediate play you reach the second level at 10,000 points and increase another level with every subsequent 20,000 points. In expert play you also reach the second level at 10,000 points, but have to score 30,000 points to move up levels from that point.

Increase in levels, added potions, and missing Sylphs are awarded only during regeneration of the dragon or at the beginning of a new monster wave, not during actual game play.

GAME END: The game ends when the Dragon is destroyed and has no regeneration potions left. You'll see the prompt "ENTER INITIALS, HIGH LEVEL: XXX". Type in three initials. You'll then see your current level with the score of your most recent game, and your highest level reached with the score. At the bottom of the screen you'll see the question "PLAY AGAIN (Y/N)?". Press the fire button or "Y" for yes, or "N" for no. If you indicate "yes" your current level of expertise and the number of genies you previously chose are displayed with the prompt "KEEP (Y/N)?". Press "Y" or the fire button to begin another game. Press "N" to respecify the game options, beginning with your experience level.