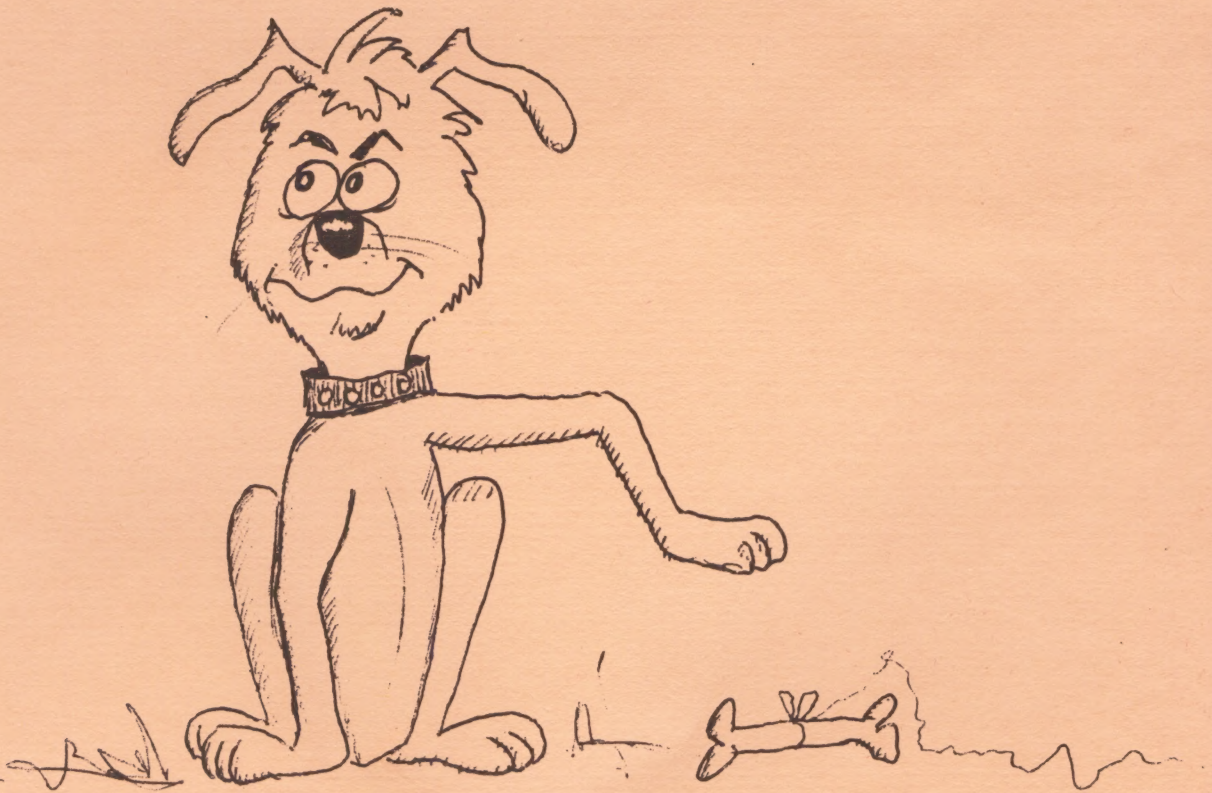


DALLAS ATARI COMPUTER ENTHUSIASTS

VOLUME 4 ISSUE 8

AUG 1983



PRESIDENT'S PERSPECTIVE

First, I would like to thank Tom for writing the Perspective for me last month. When the boss says that you will spend three weeks in Oregon you go.

NEWS FLASH!! The new 800's that were shipped in the last few months have a new CPU board in them. NO RUMOR, this is fact because I have one in my computer. The new board has a new 65XX microprocessor which we think is the 65C102. A phone call to the Arlington service center uncovered that the new chip is the same one that is used in the 5200 game machine. The most noticeable change to the computer is the lack of it going to sleep during disk reads and writes.

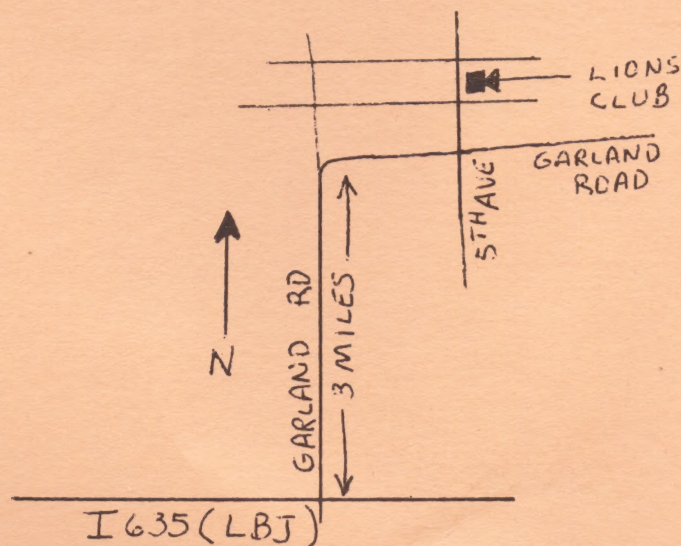
By now most of you know that we have moved to a new meeting place. I know that it is not the easiest place to get to, but the move was long needed. Our membership is now over 400 and still growing. Just because we have moved, please keep looking for bigger and better meeting places.

With our move to the new building, a few changes are in order. First, when you come into the building, you will show your club card at the door. For guests, they will be asked to sign a list which we will check each month. After two months they will be asked to please join the club. Second, newsletters will be picked up at the door. Third, the dealer area will be available only to paid club members.

Below, for those of you that have not made it to the meeting, is a map showing how to get there.

Happy Computing
Gary

Gary



ANATOMY OF DOGGIES

(by Garry Francis, reprinted from ACE N.S.W., Australia)

Who the Hell is Stan Ockers Anyway?

Way back in June 1981, the Eugene ACE Newsletter published a lunar lander program written by a fellow named Stan Ockers. This in itself was not unusual, but Stan returned with another program in the next issue. And the next. And the next. In fact, this prolific programmer from Lockport, IL has now written a program (or sometimes two) for every single issue of the ACE Newsletter since June 1981.

In December 1981, editor Mike Dunn awarded Stan with the First Editor's Award. His work is often reprinted in ANTIC and the Michigan ACE Newsletter and some programs (such as Chicken) are already considered classics.

Stan's programs may be divided into 2 very distinct categories — games and utilities. They are like a developing series of tutorials in advanced graphics techniques from BASIC. His philosophy has been "...perhaps the best way to explain what is going on is to go through an actual program..." But even if you don't want to learn about fancy graphics, the programs are still fun to use.

I have always admired Stan Ockers' work, but I was astounded to find out nearly all his programs were developed on a cassette based Atari 800 with 16k RAM and a black and white TV! He didn't even have an assembler! The moral is, of course, you don't need 256k and a quadruple density disc drive to write good programs, as many people seem to think. You merely need a good imagination and a lot of hard work.

Enter Doggies

Doggies has always been one of my favourite of Stan Ockers' games. I originally looked through the code to see how on earth it worked. Just recently, I decided it is an excellent program for the magazine. Stan has often pointed out minor deficiencies (such as lack of color), which he leaves for readers to fix themselves. So I set about to do a bit of polishing up to Doggies. Unfortunately, my polishing up got out of hand and I have now completely restructured the program, removed a couple of minor bugs, improved the response time, screen display and colors and renumbered it. But despite the weeks of work I put into it, it is still Stan Ockers' program. He should be congratulated on doing such a fine job and making my task such a pleasure.

Program Flow

Doggies begins by jumping to a massive block of initialization code which is placed at the end of the program so as not to slow things down when the main program is executing later on.

I believe the first thing you should do in any program is to have something happen immediately after typing RUN, even if you only clear the screen and print the title. A pause of 2 or 3 or even 20 seconds before anything happens is totally unacceptable, as the user may think the program has crashed. You should also make no assumptions as to screen conditions before the program was run. Therefore, as a rule of thumb, always start the initialization with a GRAPHICS statement, followed by the necessary POKES and SETCOLOR statements to set margins, colors and so on, even if you are using default values. Doggies also clears out and reserves an area of memory at the top of RAM (see Memory Allocation), then prints the instructions and the word "INITIALIZING" so you know what's going on. You can now read the instructions oblivious of the initialization which continues as you read. This takes about 11 seconds. When finished, "INITIALIZING" is overprinted by another message to "Press START to begin". When you've done so, the screen will be cleared, player missile graphics are enabled and the vertical blank interrupt routine is set up. The BREAK key is also disabled to force you to use SYSTEM RESET to abort the program. This is the only way to ensure all system parameters are back to normal when you return to BASIC. Whew!

We now set up the screen for a new game and cycle through the main program loop from lines 50 to 80. Yes, that's right! The main program loop is a mere 4 lines long! The remainder of the program is subroutines for specific actions. Each subroutine commences on a line number of a multiple of 100 and is preceded by a REM to indicate its function. I will delve into some of these later. The remainder are fairly straightforward.

Doggies uses a number of the ATARI's special features such as interrupt processing, player-missile graphics and a redefined character set. We'll now take a look at some of these aspects, but be warned: A good working understanding of BASIC is assumed.

Memory Allocation

As a lot of our members are beginners with minimum systems, I intended right from the start for Doggies to run on a cassette based system with only 16k RAM. It does this quite admirably, but if you've got a disc based system, you'll need to delete all the REMarks if you expect to run it in 16k. (You shouldn't really be using a disc drive with only 16k anyhow, as there is just too little room left for programs or high resolution graphics modes.)

Figure 1 shows the maximum memory requirements for Doggies when typed in exactly as shown in the listing. RAMTOP is an Operating System pointer at memory location 106 (\$6A) which tells us the page number of the first non-RAM byte in memory. (A page is a block of 256 bytes). By POKing a lower value into RAMTOP, we can fool the system into thinking it has less RAM than it actually has. When we carry out a GRAPHICS command, the display memory and display list will be written below the new value of RAMTOP, thereby reserving an area for our own use. We use this technique in Line 2000 to set aside an area for player-missile graphics and the redefined character set.

The player-missile graphics require 4 pages (or 1024 bytes) of memory for double line resolution and must start on a 1k boundary. We define the starting location with the variable START. The first 384 bytes of this area (i.e. START to START + 383) are not used. The area from START + 384 to START + 511 is reserved for missiles, but we will not be using them. It seems a shame to waste 512 bytes, so seeing the redefined character set requires 512 bytes, we can store it in this area. We also need a 1 page buffer between the display memory and START due to the RAMTOP dragon who gobbles up the first 64 bytes above RAMTOP whenever a GRAPHICS command or clear screen command are executed.

The GRAPHICS 0 display used for the instructions is shown in figure 1 as it is more memory intensive than the GRAPHICS 18 display used for the game itself. Just out of interest, GRAPHICS 18 only requires 20 bytes for its display list and 240 bytes for its display memory.

You can see from Figure 1 the BASIC program itself requires 7245 bytes just to reside in memory. (The structure of the tokenized BASIC program is not important at this point.) When run, it will then take up an extra 61 bytes for string storage plus a variable amount for the run-time stack (this keeps track of return addresses for GOSUBs and FOR...NEXT loops.) The area labelled "DOS" is only applicable if you've got a disc drive booted. The figures are for DOS 2. The area labelled "Operating System RAM" is reserved by the Operating System for a variety of functions. (Maybe we could delve into all this stuff in future issue.) By adding up the number of bytes used by your system, you can see a cassette based system requires well under 16k (note 16k is 16384 bytes), whereas a disc based system requires just over 16k, hence the need to delete the REMarks for a disc based system with only 16k.

Player-Missile Graphics

Whenever using player-missile graphics, you must first reserve an area (as discussed above) to store the images and then clear this area of any extraneous garbage. I used a little trick to accomplish the latter by issuing a GRAPHICS 21 command prior to reserving the P-M graphics area (see Line 2000). This has the effect of clearing out 960 bytes at the top of memory for the GRAPHICS 21 display. When RAMTOP is moved down, the 640 bytes used by P-M graphics remains undisturbed and set to all zeros. GRAPHICS 0 or 5 could also be used to clear out the same amount of memory, but these may have caused a disturbing flash of blue from the background color of the text portion of their displays. Care to minor details like this makes a really professional program.

Only Player 0 is used (for the bone) and this is double width. Its shape is read directly into the Player 0 area (START + 512 to START + 639) at Line 2140.

Vertical Blank Interrupt Processing

The vector to the Vertical Blank Interrupt (VBI) service routine is set by the call X=USR(1536) as the very last function of the initialization code. See the assembled source code for more details. For further information on how a VBI works, see "Flashing Cursor" in the December 1982 issue of INSIDE INFO. x Once set up, the VBI reads the joystick to determine whether it has been pushed left or right and moves the bone accordingly. As this is done 50 times per second regardless of what else is happening in the program, it results in a beautiful smooth motion. Again refer to the assembled source code for more details.

Redefined Character Set

Two pages of the character set in ROM (i.e. 64 characters or 512 bytes) are copied to the new locations in our reserved area. This process is speeded up by utilizing a machine language routine stored in MLS. (Refer to the assembled source code to see how it works.) It was stored in a string so you can easily incorporate it into any of your own programs. The general form of the call is X=USR(ADR(MLS),57344,CH). 57344 is the address of the start of the character set in ROM. Change this to 57856 if you wish to copy the second half of the character set (i.e. lower case letters and control graphics characters). CH is the address of the start of your new character set. You may also use this routine to copy the whole of the character set (i.e. 128 characters or 1020 bytes) by changing the last number of the DATA statement in Line 2170 from 2 to 4. This indicates how many pages are to be copied.

The Operating System pointer CHBAS at location 756 (\$214) tells ANTIC where the character set starts. Note we do not change this until after the instructions are cleared from the screen, otherwise some of the letters change into parts of little doggies while we try to read them. Very annoying!

When the character set has been copied, we redefine 34 of the 64 characters. This is the most time consuming part of the initialization. Allocating which characters are to be changed is quite a challenge, as the doggies require 34 characters, the score requires 10, and the titles and various messages I want to print require about 24 characters — a total of 68 characters, but only 64 characters are in a GRAPHICS 2 character set. I couldn't delete any parts of the doggies or the digits for the score, but by carefully rewording the messages and re-allocating some of the characters for the doggies I was able to reach an acceptable compromise. Hence messages like "GREAT STUFF" instead of "EXCELLENT", as the "X" became part of a doggie. Every character except the comma is used at some time somewhere in the screen display.

When the new characters are put together in the correct pattern, they will form the shapes for the various doggies. The shape of each doggie is made up of 6 characters in a 2 by 3 grid. As an example, the stationary doggie is shown in Figure 2. The characters for each shape are stored in DATA statements. Lines 1000-1013 are for the white doggies and Lines 1100-1113 are for the brown doggies. There must be at least 6 characters in each DATA statement, so the trailing blank spaces in Lines 1010-1013 are replaced by inverse blanks. This is important. The program will crash and you'll be given an error message otherwise. The DATA statements in Lines 1100-1113 are exactly the same as Lines 1000-1013, but in inverse video. This is how we get the 2 different colored doggies. A major subroutine of the program occurs at Lines 200-220. It prints the string DOGS at POSITION X,Y. DOGS is determined by the variable LINE. By changing LINE, we can change the shape of the printed doggie.

The current position of the doggies is stored in P5 (1 for a white doggie, 2 for a brown doggie, and 0 for the blank space). By comparing P5 with the final position represented in FS (Line 440), we can determine if he end has been reached.

Attract Mode

I could say Doggies is constantly in attract mode, as something is always happening irrespective of what the user is doing. Every time through the main loop, the program checks to see if the fire button has been pressed. If it hasn't, then it randomly selects a doggie and moves him in accordance with one of 3 randomly selected subroutines. These have the effect of making him bark, wag his tail and stomp his feet. Even when the game is over, the attract mode continues while waiting for you to press START for a new game.

The ability to move a doggie when the game is over might cause havoc. This is averted by setting a flag called ATTRACT. When ATTRACT is 0, the game is in progress and the GOTO in Line 80 makes sure the fire button is checked. When the game is over, ATTRACT is set to 10 so Line 80 will jump past the fire button routine.

Colors

The display uses a total of 6 colors in a very pleasing combination. The black background is colored by the background color register and the scungy yellow bone is colored by the Player 0 color register. Playfields 0 and 2 are used for the white doggies and brown doggies respectively. Playfields 1 and 3 are used for the green writing and blue writing respectively (you won't see the blue until the game is over). Green writing can therefore be achieved by printing a message in lower case and blue writing can be achieved by printing in inverse lower case. A problem arises here when we wish to print the score, as the digits 0 to 9 are in the range of characters colored by Playfield 0, but we want them to be colored by Playfield 3. We therefore need to convert the score to a string and manipulate the individual digits to change their color from white to green. This is done in Line 430.

The colors show up fairly well on a black and white TV and the only evidence of color "bleeding" is from the brown doggies. This is an unavoidable problem common with dark luminances. It can only be overcome by careful selection of color combinations.

Sounds

BARKS contains the range of tones for the doggies' cute barking sound used in Line 500. This was one point with which Stan Ockers was not pleased, but I could not come up with anything better. Apart from which, I quite like it.

The subroutine for the footsteps sound in Lines 900-910 is unusual. The odd numbered distortion value causes the speaker to click, then silence. Turning the voice off then causes another click. When executed together, the 2 clicks are indistinguishable and sound as though they are combined to form one single louder click. Prior to entering this subroutine, the variables V and INC are specified. V is the volume of the footsteps and INC is the increment by which the volume is increased or decreased. In this way the footsteps may stay at a constant volume while the doggie is stationary (INC=0), decrease as the doggie walks away (INC=-1) or increase as the doggie approaches (INC=1). Using a common subroutine ensures the timing of the footsteps is always the same. This subroutine is also the reason for the apparent duplication of DATA statements at Lines 1010-1013 and 1110-1113.

Scoring

Your score is incremented every time a doggie is moved. The object is to move all the doggies in as few moves as possible. 15 moves is the best you can do, but scores of around 21 are more common.

There is a sort of bug in the scoring routine. Even though it will never be encountered under normal use, I mention it here as a good example of deciding where to "draw the line" for certain error conditions. The score's color conversion mentioned above will crash if there are more than 3 digits in the score. I decided on a maximum of 3 digits for 2 reasons. Firstly, more than 3 digits will wraparound onto the next line and mess up the display. Secondly, you will have to be a complete moron to need 1000 moves. Even the smart alec who intentionally tries to crash the program will have to move one doggie every 5 seconds for an hour and a half before the program will complain. If the idea of this bug still worries you, just increase the size of MOVES in Line 2070 to 4. Our hypothetical smart alec will then be moving doggies for 14 hours!

Human Engineering

One of the most important aspects of any program is its human engineering. Here is where Doggies excels. It couldn't possibly be any simpler to use! After initialization, you need only press START for the game to begin. In fact, you can then press START again at any time and the game will restart. This comes in handy if you can see you've made a mistake and don't want to carry the game through to completion knowing you'll get a bad score.

The bone is controlled by pushing left or right on a joystick plugged into Port 1. If it goes off the screen, it will wrap around to the other side. A doggie is selected by placing the bone under the doggie you wish to move and pressing the fire button. The bone is not very fussy when it comes to selecting a doggie. It doesn't have to be directly under a doggie, just make sure at least half of it is under the doggie you wish to move. If it is exactly in the middle of 2 doggies, a choice still has to be made, so it opts for the doggie on the right.

The program won't let you make an illegal move. I won't spoil the surprise by telling you what happens, so try it and see for yourself.

As there is no keyboard input, the well known random color switching normally comes into effect after just under 11 minutes. As Doggies is such a compulsive game, you will quite likely be playing for well over 11 minutes, so the random color switching must be catered for. Any legal input (i.e. pressing START, pushing the joystick left or right or pressing the fire button) will reset the ATTRACT flag at location 77 (\$4D), thereby avoiding the random color switching. If however you have to answer the phone or you go on holidays and forget to turn the computer off, the random color switching will still be enabled as usual. This will protect the phosphors in the TV set. When you come back, just move the joystick, press the fire button or press START and everything will return to normal.

All in all, the human engineering is so well done even the youngest child (or the oldest computer critic) can also enjoy playing Doggies.

Conclusion Well, that about wraps it up for Doggies. If you've read this far and haven't keyed it in yet, then you're missing out on quite a treat. Doggies is undoubtedly the most professional game we've printed so far, so give it a go. We'll publish the solution next issue.



DOGGIES

```

1 REM #####
2 REM # DOGGIES #
3 REM # by Stan Ockers #
4 REM # Eugene A.C.E. Newsletter #
5 REM # January 1982 #
6 REM # Modified by Garry Francis #
7 REM # Reprinted by A.C.E.(N.S.W.) #
8 REM # February 1983 #
9 REM #####
10 GOSUB 2000
19 REM *** Main loop ***
20 POKE 77,0: ? #6;CHR$(125);" dog
gies"? #6;? #6;" number of moves ";CH
R$(16)
30 P#="1110ZZZ":FOR C=1 TO 7:GOSUB 100
:IF A THEN GOSUB 210
40 NEXT C:POKE 209,120:POKE 53248,120:
MOVE=#:ATTRACT=#
50 IF STRIG(0)=0 THEN GOSUB 300:GOTO 5
0
60 IF PEEK(53279)=6 THEN POKE 209,0:PO
KE 53248,0:GOTO 20
70 C=INT(7*RAND(0))+1:GOSUB 100:IF A TH
EN GOSUB 500+100*INT(3*RAND(0))
80 GOTO 50+ATTRACT
99 REM *** Which doggie? ***
100 A=VAL(P$(C,C)):IF A=1 THEN LINE=10
00
110 IF A=2 THEN LINE=1100
120 X=3*C-3:RETURN
199 REM *** Draw doggie ***
200 FOR J=1 TO 14:NEXT J
210 RESTORE LINE:READ DOG$:POSITION X,
6: ? #6;DOG$(1,2)
220 POSITION X,7: ? #6;DOG$(3,4):POSITI
ON X,8: ? #6;DOG$(5,6):RETURN
299 REM *** Process player's move ***
300 POKE 77,0:B=0:C=INT(PEEK(209)/24-0
.5):IF C<1 THEN C=1
310 IF C>7 THEN C=7
320 B=B+1:IF VAL(P$(B,B)) THEN 320
330 IF B=C THEN RETURN
340 IF C<B-2 OR C>B+2 THEN GOSUB 100:G
OSUB 000:RETURN
350 TEMP=C:FOR C=1 TO 7:IF C=TEMP THEN
400
360 GOSUB 100:IF A=0 THEN 400
370 IF C=TEMP THEN LINE=LINE+6
380 IF C>TEMP THEN LINE=LINE+7
390 GOSUB 210
400 NEXT C:C=TEMP:GOSUB 100:GOSUB 500:
GOSUB 600:LINE=LINE+8:V=8:INC=0:GOSUB
900:LINE=LINE+2:INC=-1:GOSUB 900
410 LINE=LINE+2:GOSUB 900:POSITION X,6
: ? #6;" " :P$(B,B)=P$(C,C):P$(C,C)="0"
: C=B
420 GOSUB 100:LINE=LINE+12:V=1:INC=1:G
OSUB 900:LINE=LINE-2:GOSUB 900:LINE=L
INE-10:GOSUB 700:GOSUB 600
430 MOVE=MOVE+1:MOVE#STR$(MOVE):POSIT
ION 17,2:FOR I=1 TO LEN(MOVE$): ? #6;CH
R$(ASC(MOVE$(I,I))-32):NEXT I
440 IF P#OFF THEN RETURN
450 POSITION 2,4:IF MOVE=15 THEN ? #6;
" great stuff"!GOTO 400
460 IF MOVE<20 THEN ? #6;" good goin
g"!GOTO 400
470 ? #6;"could be better"
480 POSITION 0,11: ? #6;"press start to
begin":ATTRACT=10:RETURN
499 REM *** Bark ***
500 LINE=LINE+5:GOSUB 210:FOR I=1 TO 6
:SOUND 0,ASC(BARK$(I)),12,14-I*2:SOUND
1,ASC(BARK$(I)),14,I*2:NEXT I
510 LINE=LINE-5:GOSUB 210:SOUND 0,0,0,
0:SOUND 1,0,0,0:RETURN

```

```

599 REM *** Wag tail ***
600 LINE=LINE+1:FOR I=1 TO 3:LINE=LINE
+1:GOSUB 200:LINE=LINE-1:GOSUB 200:NEX
T I:LINE=LINE-1:GOSUB 200:RETURN
699 REM *** Hop ***
700 LINE=LINE+3:V=8:INC=0:GOSUB 900:LI
NE=LINE-3:GOSUB 200:RETURN
799 REM *** Shake head ***
800 FOR I=1 TO 3:LINE=LINE+6:GOSUB 200
:LINE=LINE-6:GOSUB 200
810 LINE=LINE+7:GOSUB 200:LINE=LINE-7:
GOSUB 200:NEXT I:RETURN
899 REM *** Footsteps sound ***
900 FOR I=1 TO 3:V=4:INC:LINE=LINE+1:G
OSUB 200:SOUND 0,6,13,V:SOUND 0,0,0,0
910 LINE=LINE-1:GOSUB 200:SOUND 0,11,1
3,V:SOUND 0,0,0,0:NEXT I:RETURN
999 REM *** Shapes of doggies ***
1000 DATA '( )XZL
1001 DATA '( )XZL
1002 DATA '( )XZL
1003 DATA '( )XZL
1004 DATA '( )XZL
1005 DATA '( )XZL
1006 DATA '( )XZL
1007 DATA '( )XZL
1008 DATA >?Z+8
1009 DATA >?J X-
1010 DATA @HLK_
1011 DATA @HLK_
1012 DATA XQ_
1013 DATA XQ_
1100 DATA '( )XZL
1101 DATA '( )XZL
1102 DATA '( )XZL
1103 DATA '( )XZL
1104 DATA '( )XZL
1105 DATA '( )XZL
1106 DATA '( )XZL
1107 DATA MY<ZL
1108 DATA >?Z+8
1109 DATA >?J X-
1110 DATA @HLK_
1111 DATA @HLK_
1112 DATA XQ_
1113 DATA XQ_
1999 REM *** Initialisation ***
2000 GRAPHICS 21:START=PEEK(106)-4:POK
E 106,START-1:GRAPHICS 0:POKE 710,0:PO
KE 709,12:POKE 752,1:POKE 82,1
2009 REM *** Instructions ***
2010 POSITION 16,1: ? "DOGGIES"? : ? "T
here are 3 white doggies on the left"
2020 ? "side of the screen and 3 brown
doggies"? : ? "on the right side. You n
ust reverse"
2030 ? "their position by moving one d
oggie at"? : ? "a time"? : ? "Use the jo
ystick in Port 1 to put the"
2040 ? "bone under the doggie you wish
to"? : ? "move, then press the fire butt
on. The"
2050 ? "doggie will move into the empt
y space"? : ? "but only if he is next to
it or no"
2060 ? "more than one doggie away"? :
? "It can be done in 15 moves! Can y
ou"? : ? "do it?"
2070 POSITION 13,22: ? "INITIALISING"
:IDDH BARK$(6),HL$(32),DOG$(6),MOVE$(3)
,F8(7),P#(7):F#="2220111"
2079 REM *** Data for barking ***
2080 RESTORE 2090:FOR I=1 TO 6:READ A:
BARK$(I)=CHR$(A):NEXT I
2090 DATA 97,109,97,9,5,5

```

```

2099 REM *** VBI service routine ***
2100 FOR I=1536 TO 1578:READ A:POKE I,
A:NEXT I
2110 DATA 104,160,10,162,6,169,7,76,92
,228,173,0,211,72,41,8,208,6,230,209,1
04,24,144,7,104,41,4,208,11,198,209
2120 DATA 169,0,133,77,165,209,141,0,2
08,76,98,228
2129 REM *** P-M Graphics ***
2130 POKE 54279,START:POKE 53256,1:P#
256:START:POKE 53248,0:POKE 209,0:POKE
704,30
2140 FOR I=P#*602 TO P#*604:READ A:POK
E I,A:NEXT I
2150 DATA 195,255,195
2159 REM *** Move character set ***
2160 FOR I=1 TO 32:READ A:HL$(I)=CHR$(
A):NEXT I:CH=256:START:X=USR(ADR(HL$),
57344,CH)
2170 DATA 104,104,133,204,104,133,203,
104,133,206,104,133,205,162,2
2180 DATA 166,0,177,203,145,205,136,20
0,249,230,204,230,206,202,206,240,96
2189 REM *** Redefine characters ***
2190 READ X:IF X=-1 THEN 2560
2200 FOR I=CHX TO CHX+7:READ A:POKE
I,A:NEXT I:GOTO 2190
2210 DATA 8,7,15,31,61,109,111,110,111
2220 DATA 16,224,240,248,186,182,246,1
18,246
2230 DATA 24,12,12,7,3,3,7,15,31
2240 DATA 32,48,48,224,192,192,224,240
,248
2250 DATA 40,31,31,31,31,24,240,248
2260 DATA 48,248,248,248,248,24,31
,31
2270 DATA 56,0,7,31,63,125,237,239,206
2280 DATA 64,0,224,248,252,198,183,247
,115
2290 DATA 72,15,12,15,7,3,7,15,31
2300 DATA 80,240,48,240,224,192,224,24
0,248
2310 DATA 88,31,31,31,27,248,248,0
2320 DATA 104,248,248,248,248,216,31,3
1,0
2330 DATA 112,0,7,15,31,63,63,55,55
2340 DATA 120,0,128,192,192,64,127,255
,255
2350 DATA 208,55,7,7,3,7,15,31
2360 DATA 216,254,224,254,252,192,224,
240,248
2370 DATA 224,127,7,127,63,3,7,15,31
2380 DATA 232,236,224,224,192,192,224,
240,248
2390 DATA 240,0,7,31,63,127,239,239,20
7
2400 DATA 248,0,224,248,252,254,247,24
7,243
2410 DATA 256,1,7,15,11,3,1,3,7
2420 DATA 320,128,224,248,208,192,128,
192,224
2430 DATA 336,7,7,7,7,7,4,28,0
2440 DATA 344,224,224,224,224,224,32,5
6,0
2450 DATA 392,192,192,128,192,192,192,
192,96
2460 DATA 440,0,1,3,3,2,254,255,255
2470 DATA 446,3,3,1,3,3,3,6
2480 DATA 456,0,224,248,248,252,252,23
6,236
2490 DATA 464,15,15,15,7,3,7,15,31
2500 DATA 472,15,12,15,71,67,71,47,31
2510 DATA 480,240,48,240,226,194,226,2
44,248
2520 DATA 488,15,15,15,71,67,71,47,31

```

```

2538 DATA 496,240,240,240,226,194,226,
244,248
2540 DATA 504,240,240,240,224,192,224,
240,248
2550 DATA -1
2559 REM *** Return to user ***
2560 POSITION 9,22:?"Press START to
begin":POKE 53279,8
2570 FOR I=1 TO 20:IF PEEK(53279) < 6 T
HEN NEXT I:POKE 755,2-PEEK(755):GOTO 2
570
2580 POP :GRAPHICS 18:POKE 16,64:POKE
53774,119:POKE 756,START:POKE 559,46:P
OKE 53277,3
2590 POKE 708,14:POKE 709,104:POKE 710
,20:POKE 711,136:A=USR(1536):RETURN
    
```

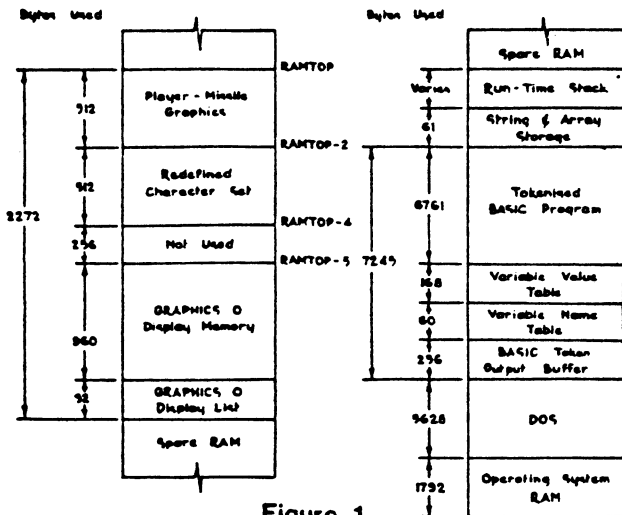


Figure 1

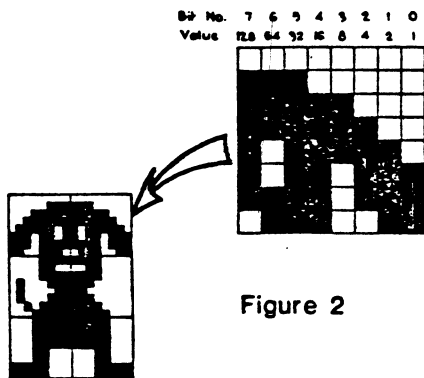


Figure 2

```

00010 .LI OFF
00020 OR #SEDD
00030 *****
00040 * UN-COMPACTOR *
00050 * NEWS LETTER ARTICLE (PART3) *
00060 * BY TOM NEWMAN *
00070 * DECEMBER 1982 *
00080 *****
00090 *
00100 *****
00110 * EQUATE TABLE *
00120 *****
00130 BUFFER .EQ #F0
00140 BUFFLO .EQ #F0
00150 BUFFHI .EQ #F1
00160 TABLE .EQ #F2
00170 TBLLO .EQ #F2
00180 TBLHI .EQ #F3
00190 SAVEY .EQ #F4
00200 REPEATS .EQ #F5
00210 HOMELO .EQ #F6
00220 HOMEHI .EQ #F7
00230 DATA .EQ #F8
00240 *****
00250 * INITIALIZATION *
00260 *****
00270 START LDA #150
00280 STA BUFFLO
00290 STA HOMELO
00300 LDA #161
00310 STA BUFFHI
00320 STA HOMEHI
00330 LDA #100
00340 STA TBLLO
00350 STA SAVEY
00360 LDA #140
00370 STA TBLHI
00380 LDY #1C0
00390 *****
00400 * PROGRAM BODY *
00410 *****
00420 CHANGE LDY #100
00430 LDA (TABLE),Y
00440 STA DATA
00450 JNY
00460 LDA (TABLE),Y
00470 STA REPEATS
00480 CLC
00490 LDA TBLLO
00500 ADC #02
00510 STA TBLLO
00520 LDA TBLHI
00530 ADC #100
00540 STA TBLHI
00550 BLOCK LDY SAVEY
00560 LDA DATA
00570 STA (BUFFER),Y
00580 CLC
00590 LDA BUFFLO
00600 ADC #120
00610 STA BUFFLO
00620 LDA BUFFHI
00630 ADC #100
00640 STA BUFFHI
00650 BEZ
00660 BEZ NEXTCOLM
00670 REENTER INC REPEATS
00680 BNE BLOCK
00690 JMP CHANGE
    
```

EDUCATIONAL SOFTWARE REVIEW - WORDMAKER (APX)

REVIEWED BY: Nat Lewis - DALACE Education Special Interest Group

SOFTWARE FURNISHED BY: Software, Etc. - Dallas

PRICE: \$24.95 APX Retail, \$21.80 at Software, Etc. (DALACE discount)

REQUIRES: 40K(disk) 32K(tape); BASIC, Joysticks, optional printer

OVERVIEW: This object of this game is to make as many 3 or 4 letter words as possible within a time limit. You can play against another person or the computer. The game is recommended for ages 5 and older dependent on word skills.

GAME PLAY: The game begins with a short tune and title screen. After the tune has played, a menu is used to allow joystick selected options of 1 or 2 players, time limit, and 3 or 4 letter words. When the game starts, one or two groups of the letters "AAA" or "AAAA" appear depending on whether 3 or 4 letter words were chosen. Moving the joystick sideways places an arrow under a position of the group of letters. Moving the joystick forwards and backwards changes the letter at the position of the arrow. When the player thinks they have made a valid word, pressing the joystick button checks the word against a built in dictionary and adds 5 points to their score if the word is good or subtracts 5 points for bad words. The timer starts at the number of minutes selected and counts down to end the round (a 30 second warning is given to allow players to plot their ending strategy). The round can be ended early by pressing START. As good words are made, they are saved in an on-screen word list. Once a word is used, it may not be reused, regardless of whether one or two players are playing. At the end of the round, the word list on the screen can be optionally printed before the scoreboard is displayed.

GOOD POINTS: The game is user friendly with a protected BREAK KEY, colorful screens, fast joystick response, and an accurate, complete word list including such unusual ones as rad or rue.

NEGATIVE POINTS: The print capability is limited in that it only prints words on screen at the end of a round. It would be more useful if all words could be printed with a single request at the end of a session. The full dictionary cannot be listed, the program is copy protected and the word list cannot be modified.

SUMMARY: The program had no detectable bugs, was fast and easy to use and is reasonably priced. If you need to learn all the valid 3 and 4 letter words, this is an enjoyable way to do it.

Are You A Computer Nerd?

64K internal RAM memory CPU with 16-bit 8088 microprocessor. Extensive software available soon.

High resolution Coke bottle monitor

Input/Output ports

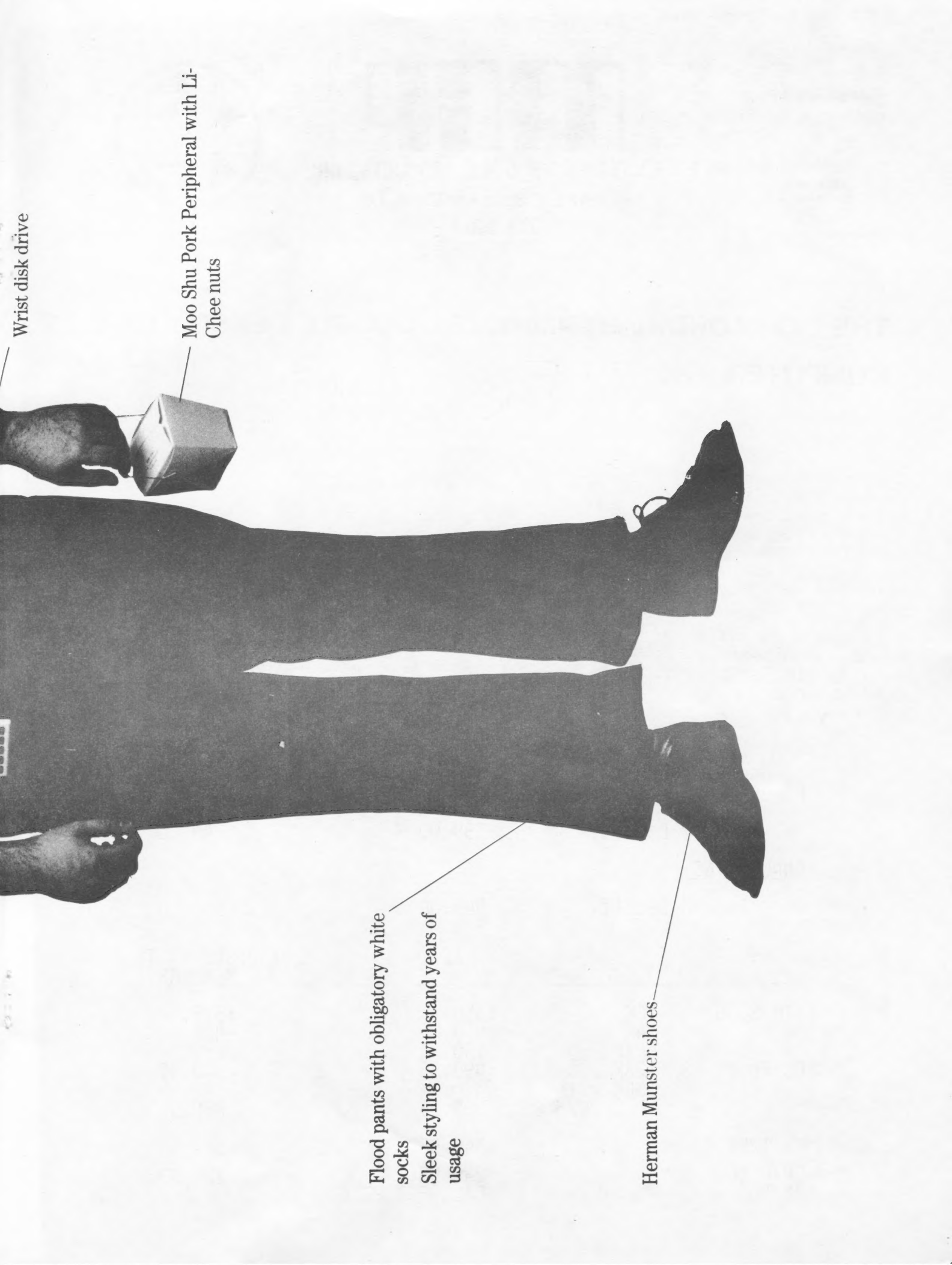
Ring around the collar

Pressure sensitive keyboard with full alphabetic, numeric, graphic and control function keys

Hi! Gee, I'd love to accept those free tickets to the Rolling Stones concert tonight, but I have to download several Comuserve files. Are you sure that's an RS-232-C interface? What's your operating system? CPM? I think that CPM is too kludgy, so I'm writing my own. Problems? Last night I was up until six A.M. debugging. When my roommate plugged in Mr. Coffee, it crashed the system. Now I'm looking for my own place. Well, it's been nice chatting, but I've got to go renumber my GOSUBS.

Nerd-Pak™ Printer; underlining, 12 colors, quiet and efficient operation for word processing. Print quality: illegible





Wrist disk drive

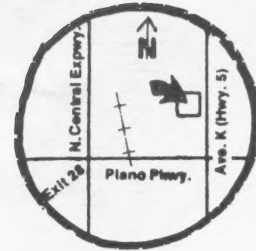
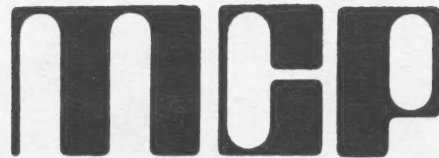
Moo Shu Pork Peripheral with Li-Chee nuts

Flood pants with obligatory white socks

Sleek styling to withstand years of usage

Herman Munster shoes

Open M-T-W-F 9-6 p.m.
 Thurs. 9-7 p.m. Sat. 10-3 p.m.



METROPOLITAN COMPUTER PRODUCTS, INC.
 801 Ave. K, Suite 4 in Plano, Tx.
 578-8504



THE EXTRAORDINARY 4MHz, Z80, DOUBLE DENSITY CP/M 2.2 COMPUTER: THE ATR8000

Standard features include the ability to run up to four 5¼" or 8" drives. Densities, sizes and types can be inter-mixed! The ATR8000 has a standard CENTRONICS parallel port and a RS-232 port.

It interfaces to the ATARI 800/400 or to a RS-232 terminal. COMING SOON: COMMODORE VIC-20 and TI 99/4 interfacing!



List

MCP

Tandon Drives with
 Power Supply for ATR 8000

Single Density	399.00	249.00
----------------	--------	--------

ANNOUNCING:

Double Density	549.00	349.00
----------------	--------	--------

LIST

10% DISCOUNT TO ATARI
 MEMBERS

ATR 8000	16K	\$350.	\$315.
	64K	450.	405.
	CP/M	150.	135.
Co-Power 88	128K	499.95	449.95
	256K with CP/M 86	999.95	899.95
128K Add-on		349.95	314.95
CP/M 86		249.95	224.95
MS/DOS		CALL FOR PRICE	

EDUCATIONAL SOFTWARE REVIEW

WHIZ KID
by ROMOX

REQUIRES 8K, CARTRIDGE, JOYSTICK, KEYBOARD
PRICE \$41.50
REVIEWED BY Harvey Cobb

Whiz Kid is a one or two player graphic hockey game that also has some educational value. You use a joystick to control a character that is holding a hockey stick. Your character is on a skating rink filled with blocks of ice. Each block of ice has a letter on it. The objective is to use your character to rearrange the blocks until they match the word that appears on the bottom of the screen. Also on the screen are three computer controlled goalkeeper characters that chase your character. If your character is touched by one of the goalkeepers, your character loses one of three lives. You are awarded points for matching the word and for hitting the goalkeepers with a block of ice.

You have a number of options. You may choose the word option to use the preprogrammed words and equations or to type in a list of your own words or equations. You have a choice of five levels of difficulty. The higher the number, the faster the movement of the characters.

Whiz Kid can be played by anyone old enough to use a joystick. The educational value is geared for preschool aged children. The gaming strategy and design is geared for preteens.

FORTH SIG

The next FORTH SIG meeting will be held 08-11-83 at 7:30 the home of Don Care. 1405 Woodcreek Dr., Richardson, Tx. 75081 H-235-6126

DAL-ACE Business SIG

The next meeting of the DAL-ACE Business SIG will be held Aug 18 at 7:30 at the home of Rich Greenlee, 512 Copher Court, Euless, Tx. H-267-7428. 1/2 mile S. of 183 off Main St. exit.

DAL-ACE EDUCATION SIG

The next meeting of the DAL-ACE Education SIG will be held the Wed. prior to the regular Saturday meeting at 6:30 at Software etc., 14400 Dallas Parkway (across from Ewing Buick).

YOUNG PEOPLES SIG

This group meets 1 hour before the regular club meeting. Open to members 8 to 16. Contact Chrris Magid. 241-7320.

DISCLAIMER

The articles and ads contained in this newsletter do not necessarily reflect the views of this newsletter or of the club. (You the reader are the final judge on anything you read.)

```

8 REM *** SET MAX MSG LENGTH ***
9 REM
10 MSGLEN=120
20 DIM MSG$(MSGLEN),M$(MSGLEN),SAV$(1)
25 REM *** INITIALIZE MSG$ TO 0 (SCREEN SPACES) ***
26 REM
30 MSG$(1)=CHR$(0)
40 MSG$(MSGLEN,MSGLEN)=CHR$(0)
50 MSG$(2)=MSG$
54 REM
55 REM *** INPUT MESSAGE ***
60 GRAPHICS 0
70 POSITION 5,10
80 ? "ENTER MESSAGE ":? :INPUT M$
90 MSGLEN=LEN(M$):IF MSGLEN=0 THEN 60
94 REM
95 REM *** CALL SUBROUTINE TO CREATE 25 LINE DISPLAY LIST ***
96 REM
100 GOSUB 300
104 REM
105 REM *** CALL SUBROUTINE TO CONVERT MESSAGE TO ***
106 REM *** INTERNAL SCREEN CODE ***
110 GOSUB 600
191 REM
192 REM *** THIS LOOP SAVES THE FIRST CHARACTER OF MSG$, ***
193 REM *** MOVES THE REST OF THE STRING TO THE LEFT ONE ***
194 REM *** CHARACTER POSITION, PUTS THE SAVED CHARACTER ***
195 REM *** INTO THE LAST CHARACTER POSITION OF MSG$, ***
196 REM *** AND THEN DOES IT ALL OVER AGAIN. ***
197 REM
200 SAV$=MSG$(1,1)
210 MSG$=MSG$(2)
220 MSG$(MSGLEN,MSGLEN)=SAV$
230 FOR D=0 TO 15:NEXT D
240 GOTO 200
294 REM
295 REM *** SUBROUTINE TO CREATE 25 LINE DISPLAY LIST ***
296 REM
300 GRAPHICS 0:POKE 82,0:POKE 752,1
304 REM
305 REM *** TURN ANTIC OFF WHILE WE MESS ***
306 REM *** AROUND WITH THE DISPLAY LIST ***
307 REM
310 POKE 559,0
314 REM
315 REM *** INITIALIZE POINTERS TO MSG$ ***
316 REM
320 MSGHI=INT(ADR(MSG$)/256)
330 MSGLO=ADR(MSG$)-MSGHI*256
334 REM
335 REM *** INITIALIZE DISPLAY LIST POINTERS ***
336 REM *** TO OLD DISPLAY LIST ***
337 REM
340 DLLO=PEEK(560)
350 DLHI=PEEK(561)
355 REM *** ADD ONE TO THE POINTER OF THE DISPLAY LIST
356 REM *** SO THAT WE WILL BYPASS ONE BLANK LINE
357 REM *** INSTRUCTION. WHICH WILL GIVE US ROOM TO
358 REM *** INSERT THE 25TH LINE.
360 DLSTART=(DLLO+DLHI*256)+1
370 DLNEW=DLHI*256
380 POKE 560,0:POKE 561,DLNEW/256
384 REM
385 REM *** WE HAVE TO MOVE THE OLD DISPLAY LIST ***
386 REM *** TO MAKE ROOM FOR THE ADDED INSTRUCTIONS ***
387 REM
390 FOR I=0 TO 27
400 POKE DLNEW+I,PEEK(DLSTART+I)
410 NEXT I
414 REM
415 REM *** CREATE 25TH LINE AND LOAD DATA POINTER ***
416 REM *** WITH THE ADDRESS OF MSG$ ***
417 REM
420 POKE DLNEW+28,66
430 POKE DLNEW+29,MSGLO
440 POKE DLNEW+30,MSGHI
444 REM
445 REM *** POKE JUMP AND WAIT FOR VBLANK INSTRUCTION
446 REM *** AND ADDRESS OF START OF DISPLAY LIST
447 REM
450 POKE DLNEW+31,65
460 POKE DLNEW+32,0
470 POKE DLNEW+32,DLNEW/256
474 REM
475 REM *** LOOP TO WRITE LINE NUMBERS ON ***
476 REM *** LEFT HAND SIDE OF THE SCREEN ***
477 REM
480 FOR I=1 TO 24
490 POSITION 0,I-1
500 IF I<10 THEN ? " ";
510 ? I;
520 NEXT I
524 REM
525 REM *** TURN ANTIC BACK ON ***
526 REM
530 POKE 559,34
540 RETURN
594 REM
595 REM *** SUBROUTINE TO CONVERT FROM ***
596 REM *** ATASCII TO INTERNAL SCREEN CODE ***
597 REM
600 FOR I=1 TO LEN(M$)
610 T=ASC(M$(I,I))
620 IF T<32 THEN T=T+64:GOTO 640
630 IF T<96 THEN T=T-32
640 MSG$(I,I)=CHR$(T)
650 NEXT I
660 RETURN

```

TAPE USERS NEED THE *TIC-600* !!

Instead of using a program recorder, you can use the *TIC-600* which is a tape interface and controller that connects your Atari computer to your present tape deck.

1. The *TIC-600* is automatic! You can select between manual or automatic tape movement control. In auto mode the computer starts and stops tape movement which some programs require. In manual mode your reel-to-reel, 8-track, VCR, or cassette deck operates as it normally does when you manually press it's controls. This permits high speed tape movement and file searching.
2. The *TIC* talks ! The *TIC-600* talks to you with two lights that tell you when you have selected manual or auto mode during all times that the computer is accessing your tape drive. If you forget to press return as your computer requires before it will receive or store data on your tape, you will know it because a light has not lit. The light comes on only when the computer tries to get data from - or give data to your tape drive.
3. The *TIC-600* uses precision high quality components that provide maximum reliability for 600 baud rate data storage and retrieval. The quality of the standard program recorder is no longer a limiting factor in getting error-free loads.
4. The *TIC-600* allows usage of the better grades of tapes that the standard program recorder should not use, if your tape deck has the controls for high bias ferric, chromium dioxide or metal type tapes. These permit heavy use over long periods of time without data loss.
5. The *TIC-600* can be used reliably to handle special programs that contain the 900 baud rate software. It can be upgraded to reliably handle 1200 baud rate. This permits faster loads because the data density is increased 50% and 100% at the 900 and 1200 baud rates.
6. The *TIC-600* is available in kit form or as a fully completed and 100% tested accessory for your Atari computer.

Kits are available for \$35.00 Completed units are \$45.00

** If you purchase the *TIC-600* this month (August), the completed units will be **
 ** discounted \$7.50 !!! **

The *TIC-600* is manufactured by Crystal Cable Company which also makes home computer cables. The Dallas area representative is John Blaschka. You can contact him at (214) 620-8210 most days and evenings. Also, you can leave a message for #356 on the *DAL-ACE* BBS ... (214) 589-1254.

NEXT MEETING -

LIONS DEN - 600 N. FIFTH ST.
GARLAND, TX.
TAKE GARLAND RD. NORTH UNTIL
YOU REACH 5TH. TURN LEFT.
TIME: 2:00 PM DATE: SEPT. 3, 1983

MEETING AGENDA

1:00 TO 2:00 SALES
2:00 TO 2:30 BUSINESS MEETING
2:30 TO 2:45 GENERAL QUESTIONS
2:45 TO 3:15 TECHNICAL QUESTIONS
3:15 TO 3:45 BREAK, NEWSLETTERS AND SALES
3:45 TO 5:00 DEMOS ETC.

ADVERTISING RATES

Personal Classified Ads: FREE
Commercial Classified Ads: Half Page - \$25.00
(Must Be Camera Ready Copy!) Full Page - \$35.00
Deadline: 15 th of each Month Mail to Editor, c/o DAL-ACE

Purchasing : Jack Withrow 680-8947
 Ed Kobus 492-2922
Education : Jeff Rutherford 1-727-2945
BBS : 817-589-1254
Library : Ernie Runyon 817-595-2243

The clubs Technical Reference Manuals are available for \$5.00 + \$1.50 shipping or pick up at club meeting. A real good deal for a 104 pages that are nicely bound (spiral spine).

*** New prices on club disks !!! Club library disks are now \$5.00 each. Take advantage of a package deal and buy 3 - get one free.

FUTURE MEETING AGENDA

AUGUST - DATA BASES FEATURING FILEMANGER 800, DATA PERFECT, AND THE DATA BASE PROGRAM IN THE LIBRARY.

SEPT - GRAPHICS - FEATURING GRAPHIC PROGRAMS SUCH AS GRAPHIC MASTER, PAINT, MICRO-PAINTER, ETC.

DALLAS ATARI COMPUTER ENTHUSIAST
(DAL-ACE)

is an independent user education group not affiliated with ATARI, Inc.. This is the official newsletter of the DAL-ACE and is used for the education of its members as well as dissemination of information concerning ATARI computer products.

Subscriptions are \$16.00 per year, good for 12 months, with renewal shown on newsletter label. Other user groups may obtain free subscriptions with the exchange of newsletters.

All material may be reprinted in any form, provided that DAL-ACE and author, if applicable, are given credit.

DALLAS ATARI COMPUTER
ENTHUSIASTS

OFFICERS

President : Gary Sewell
1-727-6567
V-President: Tom Schaeper
242-2483
Secretary : Susan Henderson
817-497-3877
Treasurer : Ed Kobus
492-2922
Member : Jack Withrow
680-8947
Member : Jim Chaney
231-4402
Editor : Harry Hafele
348-7745

DAL-ACE
10451 CLARY DR.
DALLAS, TX. 75218

U.S. POSTAGE PAID
BULK RATE
PERMIT NO. 57
ALLEN, TEXAS 75002

