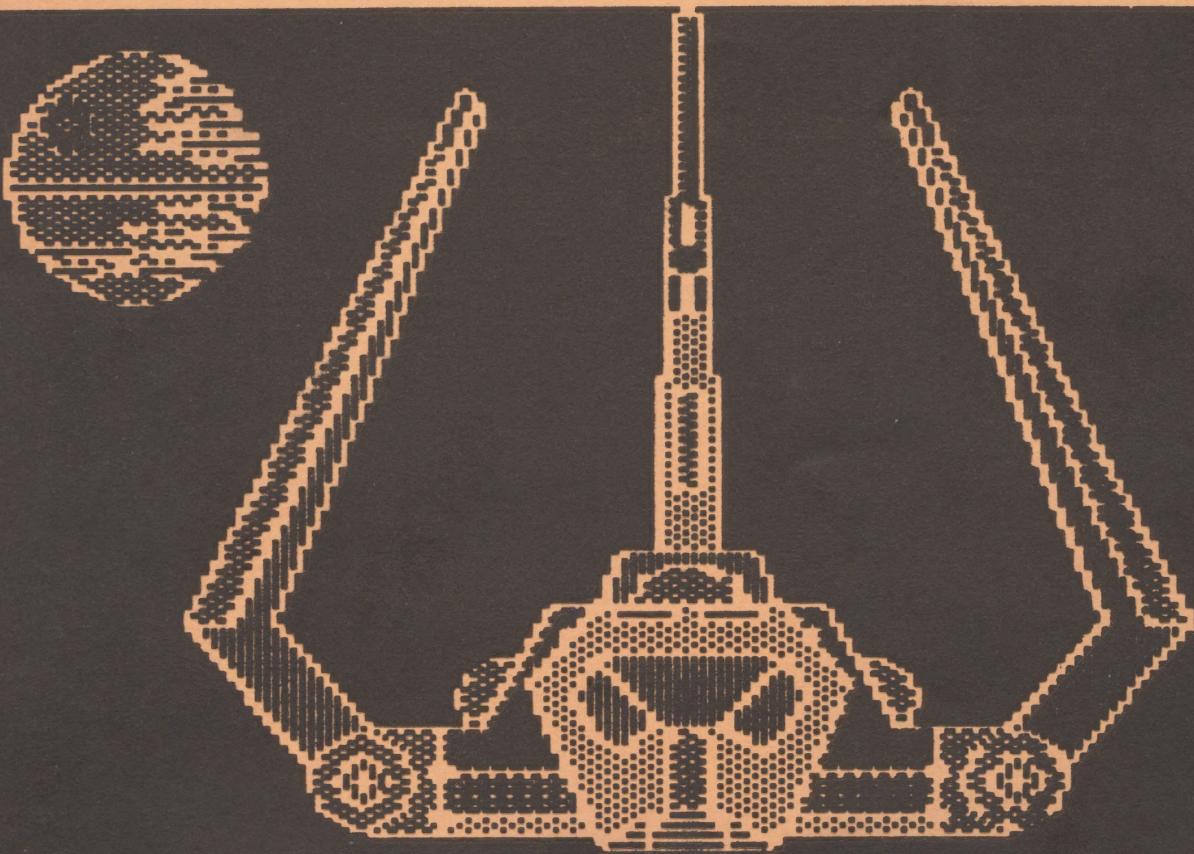


DALLAS ATARI COMPUTER ENTHUSIASTS

VOLUME 5 ISSUE 8 AUGUST 1984



PRESIDENTS PERSPECTIVE

Well, by now you have most surely heard the big news about ATARI. Only the name was not changed to unprotect the innocent. The most significant RUMOR (and most of this article will be just RUMORS) is that the Users Support Group has been eradicated! Well, we didn't get that much direct support anyway. Most all of the ACE groups are self supporting and we are no exception. BUT don't expect a re-issue of *DE RE ATARI* or *THE TECHNICAL USERS NOTES!* What we have now is what we will have tomorrow.

THE FACTS

Just in case you missed the announcements, here are the facts (as reported in the 30 JULY 1984 issue of *InfoWorld*). Jack Tramiel, founder of Commodore and its president until last February, purchased Atari for an estimated \$240 million in IOUs. Mr Tramiel has promised that the new Atari would work "hard and frugally" to provide economical computers for the public. Tramiel has dismissed most of Atari's former management and replaced them with his own people.

Market analysts expect Tramiel to discount the current Atari product line heavily to clear company inventories this Christmas. The current line of products will be continued until that time.

The new product line? Rumors are heavy that Tramiel will bring out a new computer in the \$1000 range that will compete with Apple's IIc and IBM's PCjr. Tramiel has predicted that his Atari company "will be able to be No. 1 within a year".

NOW THE RUMORS

The Atari underground has long been filled with hints at a 68000 based machine! It is said that Tramiel has kept the engineering staff for this project and re-activated work! This rumor would go along with the one that said Tramiel wanted to compete with the "big boys". If this new machine makes use of the new 7800 display chip (Maria), they would have a strong contender!

There is also a strong rumor that the new display chip for the 7800 Prosystem (256 colors per line and 100 players) will be retro-fitted (by kit) to the "800" machines. That sounds so good that it may be wishful thinking. Most experts do not expect any compatibility between the new machines and the current line (not even maintainance).

WHAT IT ALL MEANS?

First, DAL-ACE will stay in business for many moons to come! With the shut down of support for the current line, we become the only source for the Atari owners. Our numbers should grow rapidly for the next year or so. By then we will have become a source of support for the new product line (a natural transition).

Secondly, watch for real bargain prices on the XL computers (say in the \$150 to \$170 range). This will also bring new members to our doors! No, I don't think the "new Atari" will wipe us out. On the contrary, it all sounds pretty exciting!

FAILURE!

The DAL-ACE board meeting was scheduled for July 19, 1984. Only 5 of the 9 members attended. According to the current Bylaws, a 2/3 majority is required for all board meetings. Therefore, the board meeting was adjourned with no action taken. Fortunately, there were no emergencies requiring board action. Please ask your officers if they attended the last board meeting. To the best of my knowledge only one of the officers had a legitimate excuse for not attending (Ed Kobus was out of town on vacation)....jc

EDITOR'S STRING\$

First I want to thank Jeff Golden, John Pellet, Dave Gillen and Jim Chaney. Without these three members there would be no DAL-ACE newsletter. They each give an afternoon a month to the production of the DAL-ACE newsletter in addition to their significant contributions to the newsletter and club. THANK YOU!!

I will not be at the meeting this month. But that doesn't mean "yo doesn't haf ta write anythin fo da newsletter". While your reading this my family and I be lost somewhere in L.A. trying to find the USC campus to attend the Olympic Swimming Finals. The Olympics is the excuse I gave my wife for going to L. A. But, my real mission is to find out the real scoop on ATARI and find some great "deals" on software.

When I return I hope someone "suprises" me with reviews of SYNCALC, SYNFILE and SYNTREND and three never before published programs for the September newsletter. An additional surprise that would be NICE is someone that just can not go on living without being the DAL-ACE Advertising Manager.... EOL

**THE GREAT DRIVE DEAL!
PERCOM DOES IT AGAIN!**

Percom is again offering DAL-ACE a great deal on disk drives! How about a double density drive with printer port for only \$300 including cable and DOS? AND THAT INCLUDES SHIPPING! Contact Gary Sewell at (214)727-6567 for ordering information. NOTE: DRIVE UNITS ARE HALF-HEIGHT, THIN LINE DRIVES!

```
*****
*****
**
**
** V V I D D D E E E E O O O L A A N N D D D ! **
** V V I D D E E E O O L A A N N D D D ! **
** V V I D D E O O L A A A N N D D D ! **
** V I D D D E E E E O O O L L L L A A N N D D D ! **
**
**
** LOCATED ON PRESTON ROAD ACROSS FROM VALLEY VIEW **
**
*****
*****
```


PROGRAMMING

by Jeff Golden

Howdy. This is a continuation of the discussion about programming techniques and styles that was started in last month's newsletter.

The topic for today's discussion is the number one rule of programming, the KISS principle.

K.I.S.S. is the acronym for "Keep It Simple Stupid". Think about that for awhile. Read between the lines. Say it over again to yourself and remember it. It is the best advice that anyone can give to a new programmer.

This rule applies to just about every facet of programming. If you want your program to be user friendly, easy to write, bug-free, and easy to maintain, then K.I.S.S.

When programming, it is easy to get carried away by trying to do too much with one program. This situation is best exhibited when a programmer provides four different ways for doing essentially the same thing. This happens all too often in the real world.

For example:

```
RUN*D:PROGRAM
RUN "D:PROGRAM
RUN "D1:PROGRAM
RUN "D1:PROGRAM"
```

All of these formats do the very same thing and it can be argued that they are all nice to have. Why some of them even save two whole keystrokes.

But think of it this way. How much permanently allocated memory space is being wasted in order to provide the programming necessary to determine that all of these formats mean the same thing? How much documentation was necessary in order to explain to the newcomer that he can run his program with any of the above formats? How much confusion was generated by having to learn 4 different ways to run a program when one would have been sufficient?

Do I recommend that Atari rush out and change BASIC and DOS? No, after all they have to keep up with the Jones's. MS-DOS even provides another format, and I won't even get into what DOS-XL provides. I am just trying to prove a point. That multiple ways of doing the same thing are more common than we realize, and they add to the difficulty of programming,

documenting, learning, and using a program.

Programmers seem to come in three flavors; beginners, intermediates, and advanced programmers.

The beginners usually write nice straight-forward linear code. However because of their lack of experience, they tend to avoid the task of keying in sufficient error checking, and they also tend to spend very little time thinking about their design. Their lack of elegance sets their work apart from that of the more advanced programmers.

The intermediate programmer has learned how to use the fancy features of his machine, but he has not yet learned the first rule of programming. Even worse, he may know the rule, but doesn't use it. Perhaps he thinks the ability to write a complex program, that no one else can understand, is the mark of an expert programmer.

What he doesn't realize is that, any moron can write an incomprehensible program, and that an expert programmer is someone who has the ability to make a complex program look simple.

It is easy to recognize the code written by an intermediate. The code resembles a mess of spaghetti. It is full of meaningless names, tangled GOTOs, and multiple ways of doing things.

The sad part is, the intermediate programmer truly believes that he is programming at an advanced level, and may never advance any further. The intermediate works very hard at his job. Creating a mess is easy enough, but getting it to run is another story.

The advanced programmer is a former intermediate who has finally figured out why he has been working so hard. He has discovered the KISS principle, and also top-down design, and structured programming.

The KISS principle, top down design, and structured programming are interrelated. It is difficult to have one without the others.

Next month, we will discuss top down design and structured programming. What they are, and what they are not. To those of you who may have attended some formal programming classes, "what they are not", may come as a surprise.

Now, what was that rule # 1 again?

K.I.S.S. Where are you Suzy?

The Utility Room: FASTIO Revisited

By Dave Gillen

Welcome back to my Utility Room. In your utility room at home, you find tools and appliances to keep your home working. In mine, I will try to present you with programs and routines that you may use as tools for your Atari. Today I'm going to talk about a routine to do fast disk I/O.

Recently I've been puttering with some applications that do a LOT of disk I/O. Now, the Atari 810 is slow to begin with. Couple that with Atari BASIC and we're talking dog city (why I use an 810 and Atari BASIC is another story altogether). Anyway I remember reading an article in the "Atari Computer User Group Of Dallas Official Newsletter" (that's what this newsletter used to be called), back in the November 1982 issue on "A Quick and Easy I/O Routine for Atari BASIC." So I yanked out the routine and put it on my system.

It worked but not to my satisfaction. The routine had this bad habit of running FOREVER! It would never tell the calling program that there was no more data! Hmm, what's wrong with this picture? Well, being an adventurous sort, I went in and doctored it! While in the code, I also added a feature or two (a bad habit I picked up somewhere in my career). So for your enjoyment, I will present the "new and improved" version which I prefer to call, "FASTIO."

FASTIO is a machine language routine that you call as a USR function from BASIC. It allows you to read or write large chunks of a file much faster than if you use normal BASIC INPUT or PRINT statements (and, God forbid, GET or PUT statements). I don't want to mislead you that it is a cure-all or that it'll burn up your disk drive, but it does really help!

As I am wont to do, I'll explain the guts of the code to the curious (there's a little bit of voyeur in all of us) and then I'll describe how to use this routine. If you're not one of the curious, skip down a few paragraphs and join me down there.

First off, take a gander at listing #1.

Since I don't have line numbers, I'll explain the functional blocks (which I've separated by dashes). If (or when) you enter this program, you may omit those dashes.

In Block #1, we assign several values to labels. This is done with the EQU statements. The reason for this is to make the code more readable and easier to change down the road. It is a good programming habit to follow.

Block #2 is where we extract the parameters from the stack. These are placed on the stack in the reverse order that they were passed in the USR statement. The first value on the stack is the number of arguments sent by USR. For this routine, there must be three (otherwise, we'll loop forever, hanging up the machine).

The last argument removed from the stack (the ADR(bufferlen) value) is a bit unusual. We must take it off the stack and then load an address from it. This is because we are using this value as an address pointing us to a string containing the length of the buffer. The value itself is NOT the one that we are looking for.

Block #3 looks at the 1st auxiliary byte in the IOCB for the open file. This byte determines if the file is being appended to, read from, or written to. Based on this, the corresponding CIO command is placed in the IOCB. At this point CIO is called to perform the data transfer.

Block #4 deals with the return from CIO. If the end of the file has been read, then the actual number of bytes transferred are returned to the caller. Otherwise, we return to the caller with the original byte count left intact. In this instance, the status sent back by CIO is also returned.

Non-assembly folks, can pick-up here where they left off. Listing #2 is a routine that you may incorporate into your BASIC program. It is called to build the USR function. You should type it in and then LIST it to a file. I use FASTIO.BSC for the filename (the command would be: LIST"D:FASTIO.BSC").

To use this routine, you should write your own code and then ENTER FASTIO.BSC from disk (the command would be: ENTER"D:FASTIO.BSC"). Be sure not to use the line numbers that FASTIO.BSC uses, because they will be trampled by the ENTER command.

Your program should first do a GOSUB FASTIOSETUP to set up FASTIO%. Then use the USR function to invoke FASTIO%. There are four arguments for the USR function:

Argument 1, the address of the FASTIO% string

Argument 2, the channel number of the file. This value (or variable) was the same one used in the OPEN statement for this file.

Argument 3, the address of the string that will be filled (if reading) or emptied (if writing).

Argument 4, the address of a string that contains the length of the string from argument 3. This is a bit tricky. It is best to explain this one by example. Observe this little section of BASIC code.

```
DIM STRINGSIZE$(2), STRING$(300)
MULTIPLEOF256=INT(300/256)
EXCESS256=300-(MULTIPLEOF256*256)
STRINGSIZE$(1)=CHR$(EXCESS256):REM LOW-ORDER BYTE
STRINGSIZE$(2)=CHR$(MULTIPLEOF256):REM HIGH-ORDER BYTE
STAT=USR(ADR(FASTIO%),CHANNEL,ADR(STRING%),ADR(STRINGSIZE%))
```

Listing #3 is a sample program that you may use as a model. Happy computing and don't burn up your disk drive!

Listing #1:

```

TITLE 'FASTIO (13Nov83, 01:30)'
;
;       FASTIO
;
;... A USR function that allows for very rapid I/O
;... from a BASIC program.
;
; Calling Sequence:
; status =
;   USR(ADR(FASTIO*), iocb, ADR(buffer), ADR(bufferlen))
;
; Where:
; status, is the CIOV status returned from this function
; ADR(FASTIO*), is the address of the object code string
; iocb, is a variable containing the channel used for
;   I/O
; ADR(buffer), is the address of the buffer to be
;   used for I/O
; ADR(bufferlen), is the address of a 2-character string
;   containing the number of bytes to transfer. If,
;   during a read, end of file occurs then the number
;   of bytes read will be returned in this string.
;
;... This routine was originally written by Mark Harris.
;... It appeared on page 5 of the November 1982 issue of
;... the "Atari Computer User Group of Dallas Official
;... Newsletter." Acknowledgement is so made.
;
;   12Nov83, Dave Gillen
;
----- Block #1 -----
FR0      EQU $00D4;BASIC rtn value
ICCMD    EQU $342 ;COMMAND BYTE
ICBAL    EQU $344 ;BUFFER ADDRESS (LOW)
ICBAH    EQU $345 ;BUFFER ADDRESS (HIGH)
ICBLL    EQU $348 ;BUFFER LENGTH (LOW)
ICBLH    EQU $349 ;BUFFER LENGTH (HIGH)
ICAX1    EQU $34A ;ON OPEN - READ=4/WRITE=8 WRITE APPEND=9
GETCHR   EQU $07 ;GET CHARACTER COMMAND
PUTCHR   EQU $08 ;PUT CHARACTER COMMAND
CIOV     EQU $E456 ;CALL CIO FROM JUMP TABLE
CIOEOF   EQU $88 ;End of file
CIOINVCMD EQU $84 ;Invalid Command
          ORG $0600 ;ARBITRARY ORG (CODE IS RELOCATABLE)
;...
----- Block #2 -----
RETRY
  PLA ;PULL OFF ARGUMENT NUMBER
  CMP #$03 ; ARE THERE 3 ARGUMENTS
  BNE RETRY ;NO, THEN PULL AGAIN
  PLA ;PULL OFF #IOCB HIGH BYTE
  PLA ;PULL OFF #IOCB LOW BYTE
  ASL A ;FIGURE OFFSET VALUE FROM #IOCB
  ASL A
  ASL A
  ASL A
  TAX ;IOCB OFFSET MUST BE PASSED TO CIO FROM THE X-REGISTER
;USE ABSOLUTE INDEX INSTRUCTION FOR EASY CHANGE OF IOCB OFFSET
  PLA
  STA ICBAH,X
  PLA
  STA ICBAL,X
;
;... Remove the address of the string
;... containing the size. Store in
;... FR0 and then load the size
;... indirectly from it. BASIC
;... passes addresses in H/L order.
  PLA
  STA FR0+1 ; High address byte
  PLA
  STA FR0 ; Low address byte
  LDY #$00 ; Use as indirect index
  LDA (FR0),Y
  STA ICBLL,X ; Low byte of size
  INY
  LDA (FR0),Y
  STA ICBLH,X ; High byte of size
----- Block #3 -----
  LDA ICAX1,X ;ROUTINE TO DETERMINE COMMAND BYTE VALUE
  CMP #$04
  BNE WRITE
  LDA #GETCHR
  CLC
  BCC COMMAND
WRITE
  CMP #$08
  BNE APPEND
  LDA #PUTCHR
  CLC
  BCC COMMAND
APPEND
  LDY #CIOINVCMD ;Default err val rtned
  CMP #$09
  BNE EXIT
  LDA #PUTCHR
COMMAND
  STA ICCMD,X ;STORE GET OR PUT CHARACTER COMMAND HERE
  JSR CIOV ;READY FOR CIO CALL
----- Block #4 -----
;
;... Transfer the number of bytes
;... used in the I/O function to the
;... buffer length string in the
;... calling sequence.
  CPY #CIOEOF ; End of file?
  BNE EXIT ; ... No, don't alter
;
  LDY #$00 ; Indirect index
  LDA ICBLL,X
  STA (FR0),Y ; Low byte of size
  INY
  LDA ICBLH,X
  STA (FR0),Y ; High byte of size
  LDY #CIOEOF ; Restore status value
;
EXIT
  STY FR0 ;Return status from CIO
  LDY #$00
  STY FR0+1 ;High order is 0
  RTS ;RETURN TO BASIC
  END

```

Listing #2:

```

101 REM FASTIO (13Nov83)
121 REM
141 REM A BASIC USR function that supports quick disk I/O.
161 REM
181 REM Original routine written by Mark Harris
191 REM (of DALACE). It appeared on page 5 of the Nov.
201 REM 1982 issue of the "Atari Computer User Group
211 REM of Dallas Official Newsletter"
241 REM
261 REM Modified for local use by Dave Gillen
281 REM
301 REM Calling Sequence:
321 REM For part 1 (to setup USR function)
341 REM GOSUB FASTIOSETUP
361 REM
381 REM For part 2 (invocation of the USR function)
401 REM status = USR(ADR(FASTIO%), iocb, ADR(bufadr),
421 REM ADR(buflen))
431 REM
441 REM Where:
461 REM status: is the CIO status value returned for
471 REM the program's use.
481 REM iocb: the channel number on which the file is
491 REM opened.
501 REM ADR(bufadr): the address of the data buffer.
521 REM ADR(buflen): the address of a two-character
531 REM string that contains the length of the buffer.
541 REM In low/high byte order
551 REM
561 REM
571 REM
581 FASTIOSETUP=31200
31200 REM (FASTIOSETUP)Sets up FASTIO% for later calling.
31220 DIM FASTIO%(100)
31240 RESTORE 31320
31260 FOR I2=1 TO 100
31280 READ BYTE: FASTIO%(I2)=CHR% (BYTE)
31300 NEXT I2:RETURN
31320 DATA 104, 201, 3, 208, 251, 104, 104, 10, 10, 10,
10, 170, 104, 157, 69, 3, 104, 157, 68, 3
31340 DATA 104, 133, 213, 104, 133, 212, 160, 0, 177,
212, 157, 72, 3, 200, 177, 212, 157, 73, 3, 189
31360 DATA 74, 3, 201, 4, 208, 5, 169, 7, 24, 144, 17,
201, 8, 208, 5, 169, 11, 24, 144, 8
31380 DATA 160, 132, 201, 9, 208, 27, 169, 11, 157, 66,
3, 32, 86, 228, 192, 136, 208, 15, 160, 0
31400 DATA 189, 72, 3, 145, 212, 200, 189, 73, 3, 145,
212, 160, 136, 132, 212, 160, 0, 132, 213, 96

```

Listing #3:

```

100 REM FASTIOTEST (31Mar84)
200 REM
220 REM A sample program to show the use of FASTIO
230 REM a quick I/O routine.
240 REM
260 REM Type this in exactly as shown, then do an:
280 REM ENTER "D:FASTIO.BSC"
1000 REM
1100 DIM BUFFER$(1000),BUFSIZ$(2),FILENAME$(15)
1120 CIOOK=1:CIOEOF=136
1200 REM
1220 REM Initialize FASTIO%
1240 REM
1260 GOSUB FASTIOSETUP
1300 REM
1320 REM Get a filename from screen.
1340 REM
1360 ? "Enter file name to read";
1380 INPUT FILENAME$
1400 REM
1420 REM Open a channel to the file we want to read.
1440 REM
1460 OPEN #1,4,0,FILENAME$
1500 REM
1520 REM Build the buffer size string
1540 REM
1560 MULTIPLEOF256=INT(1000/256)
1580 EXCESS256=1000-(MULTIPLEOF256*256)
1600 BUFSIZ$(1)=CHR$(EXCESS256):REM LOW-ORDER BYTE
1620 BUFSIZ$(2)=CHR$(MULTIPLEOF256):REM HIGH-ORDER BYTE
1700 REM
1720 REM Loop until the end of the file
1740 REM
1760 BUFFER$(1)=" ":BUFFER$(1000)=" ":BUFFER$(2)=BUFFER$
1780 STAT=USR(ADR(FASTIO%),1,ADR(BUFFER%),ADR(BUFSIZ$))
1800 REM
1860 BUFSIZ=ASC(BUFSIZ$(2))*256+(ASC(BUFSIZ$(1)))
1900 REM
1920 ? BUFFER$(1,BUFSIZ)
1940 ? "*** (;BUFSIZ;) characters read ***"
2000 REM
2020 REM If not at end of file, go back for some more.
2040 IF STAT<>CIOEOF THEN GOTO 1700
2100 REM
2120 REM At this point, we're done.
2140 REM
2160 CLOSE #1
2180 END

```

THE FACTORY

by HesWare

review by Myron Walters

THE FACTORY is an educational game for children 6 years old and up. It will offer a challenge even to adults. THE FACTORY processes raw flat slabs of "steel" into products. It is made up of three levels. Each level may be entered from the initial Menu.

1. TEST A MACHINE - This level allows you to test the three different machines available individually: the PUNCH, the ROTATE and the STRIPE. Each of the three machines has several capabilities.

a. The PUNCH - You can punch either round or square holes. In addition you can punch from one to three holes.

b. The ROTATE - You use this machine to rotate your slab of "steel" either 45, 90, 135 or 180 degrees.

c. The STRIPE - This is a paint machine which paints either thin, medium or thick stripes on your slab of "steel".

2. BUILD A FACTORY - This level allows you to build a factory, selecting any of the previously described machines, with up to a total of eight machines in your factory. This selection introduces the interaction between each of the three machines and prepares you for the ultimate challenge of THE FACTORY, making a product upon request.

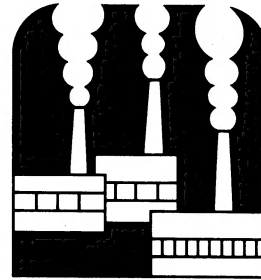
3. MAKE A PRODUCT - THIS IS THE CHALLENGE! The program presents a product and your job is to select the machines and put them in the correct order to manufacture the requested product, using any combination of the three machines. If you fail to make the requested product, you have as many opportunities as you want to make the requested product. Even a successful product may be repeated as many times as you wish for practice or in an attempt to make it with fewer steps in your factory.

THE FACTORY introduces to children concept of how products are made. It teaches and requires logical thought and offers a challenge.

THE FACTORY has color graphics in all three levels and sound in levels 2 and 3.

In my opinion this type of game offers more than the average educational software. THE FACTORY teaches how to think logically and then shows the results of your decisions in a non-threatening manner.

THE FACTORY is available for \$24 - \$29 from vendors at our meeting.



*** SOFTWARE REVIEW ***

by John Pellet

More Terminal Emulators

Two terminal programs have become available on CompuServe for Atari users - TSCOPE and DISKLINK. Both are fully compatible with the new Atari modems without the 850 interface.

TSCOPE supports the CompuServe binary file transfer protocol, which is necessary for downloading object or tokenized files from CompuServe. It also supports VIDTEX for color graphics portions of CompuServe (like stock charts and color radar). TSCOPE supports 40, 37, and 32 column text, 1200 or 300 baud, simultaneous printing of screen, and several other interesting features. It does not work with the ATR 8000. Basic is not required. It does not support autodialing or xmodem. In summary, TSCOPE is a limited terminal emulator that supports a file transfer protocol used only by CompuServe, some graphics, and the new Atari modems.

DISKLINK is a general use terminal emulator that supports the 850 interface and the new Atari modems. It supports autodialing on the new Atari's only, xmodem (but not, unfortunately, with some early versions of AMODEM or AMIS), file capture, 300 or 1200 baud, auto logon key sequences (in a limited fashion), and an online help file. It does not support the ATR 8000. Basic is not required. In summary, DISKLINK is a good general purpose terminal emulator that is likely to be most appreciated by owners of the new Atari modems.

Both of these programs are in the public domain and are available as downloads from CompuServe. They may also appear in the DALACE library at some point in the future. If anyone would like a copy I will try to have some copies available at the next few meetings. Neither of them can make a dent in my current preference for one of the later versions of AMODEM.

TEN COMMANDMENTS OF ELECTRONICS

1. Beware of lightning that lurketh in an undischarged capacitor, lest it cause thee to be bounced upon thy buttocks in a most ungentlemanly manner.
2. Cause thou the switch that supplies large quantities of juice to be opened and thusly tagged, so thou days may be long on this earthly vale of tears.
3. Prove to thyself that all circuits that radiateth and upon which thou worketh are grounded, lest they lift thee to a high frequency potential and cause thee to radiate also.
4. Take care thou usest the proper method when thou taketh the measure of high voltage circuits so that thou doth not incinerate both thee and the meter; for verily, though thou has no account number and so can be easily replaced, the meter doth have one, and, as a consequence, bringeth much woe unto the Supply Department.
5. Tarry thou not among those who engage in intentional shocks, for they are surely nonbelievers and are not long for this world.
6. Take care thou tampereth not with interlocks and safety devices, for this will incur the wrath of thy seniors and bringeth the fury of the safety officer down about thy head and shoulders.
7. Worketh thou not on energized equipment, for if thou doeth, thy buddies will surely be buying beers for thy widow and consoling her in other ways not generally acceptable to thee.
8. Verily, verily I say unto thee, never service high voltage equipment alone, for electric cooking is a slothful process and thou might sizzle in thine own fat for hours on end before thy Maker sees fit to end thy misery and drag thee into His fold.
9. Trifle thou not with radioactive tubes and substances, lest thou commence to glow in the dark like a lightning bug, and thy wife be frustated nightly and have no further use for thee except thy wage.
10. Commit thou to memory the works of the prophets, which are written in the instruction books, which giveth the straight dope and which consoleth thee, and thou cannot make mistakes, sometimes, maybe.

AUTHOR UNKNOWN

FREE SOFTWARE FOR YOUR ATARI

by Dorothy and David Heller

Review by Sandy Park

If you're the least bit intelligent as a consumer, you'll probably approach the title of this book with a great deal of suspicion. When I first looked at this book, I was convinced that the title was just sitting there on the cover to get my attention, and the book was probably worthless.

However, I'm delighted to report that I was totally wrong. The book really does tell about free software, and is a very handy reference tool as well.

Mainly, the book is about access to free software through bulletin boards. There are two excellent chapters on how to use modems and bulletin boards. The authors also suggest that readers contact Users Groups for free or almost free software.

There's a very good section on educational software, and several groups which develop and evaluate this type of product are listed.

The book has appeal to a very wide range of readers. Those who have limited experience with computers will really appreciate the style in which the book is written. It starts from scratch and explains things in a very direct -- and often entertaining -- manner.

At the same time, users who have been technobabblers for a long time will benefit from access to several of the resources in the book, including a listing of all of the Official ATARI Computer Users Groups, an Atari Modem Buyers Guide, a Directory of Atari Bulletin Boards, and a very extensive listing of Public Access Message Systems.

*** PERSONAL ADS ***

Personal Ads are printed free of charge for any paid up member of DAL-ACE. If you have something to sell (COMPUTER ORIENTED PRODUCT) or need something for your system, send your "copy" to the newsletter address by the 15th of the month for publication.

EDITORIAL STAFF

By Myron Walters

The Newsletter has several editors who will be sharing the responsibilities and who will be working as a team to assure that the DAL-ACE publication continues to be one of the best in the country. Below is the list of staff members and a description of their responsibilities. If anyone would like to help in any of these areas, please call one of the editors for information.

Art Editor: John Henson. John will be responsible for preparing the cover art each month. Also, he will help with the overall Newsletter design and layout.

Software Editors: Myron Walters and John Pellet. John and I will be responsible for writing or obtaining reviews of software. We will be encouraging DAL-ACE members to write software reviews, and we will also be examining other user-group newsletters for appropriate articles to reprint.

Hardware Editors: Mark Maxham and Jeff Rutherford. Mark and Jeff will be contributing articles and reviews about hardware. If you have any suggestions or comments about what they might want to include, please get in touch with them.

Programming Editors: Jeff Golden and Dave Gillen. Jeff and Dave are looking for good articles about programming. Also, they welcome submissions that contain even the briefest of programming tips. Remember, even though you may be very familiar with a particular programming technique, other members, especially the new members, may not know about it. Please give Jeff or Dave a call if there's anything you'd like to contribute.

Books and Magazines: Jeff Rush. Thousands of computer books and magazines are now available, and it's often difficult to know which ones are worth taking a look at. If you'd like to do a review of a particular publication, give Jeff a call.

Production Managers: John Pellet and Jeff Golden. John and Jeff will be working with the Senior Editors on production day to help prepare the copy for press. We'd love to have all of the copy "camera ready" before this session begins. If you'd like to submit something, take a look at a recent Newsletter to see how it is formatted. We can still take submissions on disk, but camera ready copy is terrific.

Distribution Managers: Roger and Steve Markeley, Cathy Barros, and David Miller. This committee will be helping with the distribution of the Newsletters at the meeting, and will also be responsible for the mailing. In addition, they will be updating the mailing list of Newsletters which are exchanged with other user groups. Roger has already begun organizing the distribution process, and will have it completed within a few weeks.

Senior Editor: Myron Walters Jack of all trades, master of none.

NEWSLETTER SCHEDULE

September Newsletter: copy should be turned in by August 18th. The production meeting will be August 18th.

October Newsletter: copy should be turned in by September 22nd. The production meeting will be September 22nd.

NEWSLETTER STAFF

If you would like to assist with the Newsletter, or if you have a submission or suggestion, please contact the appropriate staff member listed below.

ART: John Henson, 691-4154.

SOFTWARE: Myron Walters, 691-3401; John Pellet, 792-3175.

HARDWARE: Mark Maxham, 238-5949; Jeff Rutherford, 727-2945.

PROGRAMMING: Jeff Golden, 252-3268; Dave Gillen, 245-2732.

BOOKS AND MAGAZINES: Jeff Rush, 661-1289

PRODUCTION: John Pellet and Jeff Golden, numbers listed above.

DISTRIBUTION: Roger and Steve Markeley, 231-6918; Cathy Barros, 368-8499; and David Miller, 391-1926.

HELP WANTED!!!

Responses to previous calls for help have been tremendous, but there are still a few unfilled positions on the Newsletter staff. Please let us know if you'd like to help out. Here's what we need:

ADVERTISING MANAGERS. Recruit advertisements for the newsletter. Collect payment. Deliver camera ready copy to the editor. Deliver payment to the treasurer.

SIG EDITOR. Contact SIGS each month to find out about activities; prepare SIG NEWS article each month.

Call Myron Walters to volunteer for one or more of these exciting positions. The pay isn't too great, but there are a lot of intangible rewards that come from making contributions to the club.



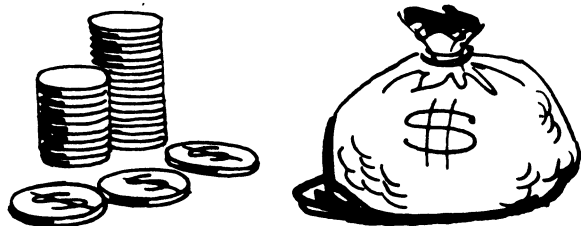
DAL-ACE REPORTS

By Bob Felice

SOFTWARE REVIEW: Murder on the Zinderneuf, Electronic Arts, \$40.00 **Category:** Game. **GRAPHICS:** Excellent **SDOUND:** Good **USER INTERFACE:** Good

Adventure games are limited: for instance, after you have solved the puzzle of how to get past the snake in Colossal Cave, you always get past it the same way. The situation never changes, and this limits the play value of the adventure. In a similar vein, once you know who murdered the victim in a mystery, you have lost a great deal of the value of the mystery. In Murder on the Zinderneuf, the authors have attempted to write a computer mystery with unlimited play value. Every game has a different victim, murdered for a different reason. There are sixteen characters in the game (one of whom has been murdered). In addition, you get to play the game as any one of eight different detectives, all of which are vaguely familiar. For instance, there is Miss Agatha Marbles, and Chief Inspector Emile Klutzeau. Each detective has a unique personality, and interrogates the witnesses in a different fashion. You also gather clues, by wandering around the Zinderneuf, a luxury Zeppelin. The setting is 1936, and the packaging, as always for Electronic Arts, is slick. This had all the potential of becoming a classic game, like M.U.L.E., another game from Electronic Arts.

Unfortunately, it didn't work out. Murder on the Zinderneuf is played in real time. You have 36 minutes to solve the case, and that isn't nearly enough time to search the Zeppelin for clues, interrogate all of the witnesses, and piece together all of the information you gather. **THE GAME DOESN'T HAVE A PAUSE BUTTON!** Unlike Witnesses, or Deadline, there is no way to think about the data you have collected. My favorite detective, Sherlock Holmes, described some of his favorite cases as "Two pipe problems". That is, he would smoke two bowls of pipe tobacco contemplating the solution to a particularly thorny case. That luxury is denied the player in Zinderneuf. If you need time to think about what's going on, my advice is to wait for Witness, or buy Deadline.



DAL-ACE RESOURCE EXPERTS

Bailey, Bill. 271-4784. (B)
 Chan, David. 495-8207. (B)
 Clarke, Tim. 960-7372. (B)
 Davis, Bo. 270-5544. (I)
 Dunayer, Adam. 680-9018. (I)
 Gillen, Dave. 245-2732. (I)
 Greenlee, Rich. 267-7428. (B)
 Hafele, Harry. 348-7745. (I)
 Maxham, Mark. 238-5949. (B)
 Mullens, Dow. 272-3004. (B).
 Newell, Wes. 423-1781. (I)
 Oradat, Cecil. 690-3155. (I).
 Parker, Travis. 840-9586. (B)
 Penn, Tandy. 276-8796. (I)
 Pennington, Jerry. 223-8132. (B)
 Rabinek, Tom. 681-2280. (I).
 Runyon, Ernie. (817) 485-0871. (I).
 Rush, Jeff. 661-1289. (I).
 Sadow, Phil. 644-3325. (I).
 Sagor, Fred. 424-0291. (I)
 Scott, Ron. 436-0297. (B)
 Sladeczek, Joe. 276-1443. (B)
 Taylor, Frank. 242-4598. (B)
 Wiant, Jim. 690-4188. (I)
 Williams, Edmund. 341-9297. (I)
 Zegub, Tom. 234-1958. (B)

*** SOFTWARE REVIEW ***

by John Pellet

Albert E. Spreadsheet

This spreadsheet is included with the Indus GT disk drive. It is a RAM-resident spreadsheet which uses Basic. It is very similar to a program called "Basi-Calc" which has been out on the Atari for some time. It is limited when compared to the more powerful spreadsheets out for CPM or MS-DOS machines but in the Atari world it has a chance. The major strengths of the program are: 1) price (free with the Indus - about \$50 retail), 2) simplicity (also a disadvantage but a good introduction to spreadsheet capabilities), and 3) RAM-resident so no drive access time. The major disadvantages of the program are: 1) SLOW (it runs in Basic), 2) SMALL (even in a 48K machine maximum spreadsheet size is about 20x25 - 500 cells), 3) limited formula capability, 4) copy-protected so no backup, and 5) very poor documentation - believe it or not, worse than early versions of Data Perfect.

In summary, Albert E. Spreadsheet is worth what it costs (the Indus is not available unbundled) but if I were investing in a spreadsheet for the Atari I would choose the new Synapse version, "Syn-Calc".

BASIC XL

by OSS

review by Mark Maxham

(EDITOR'S NOTE: Yes another review of BASIC XL. #1 A member thought enough of the product to take the time to write this review. #2 No one should miss out on the power of BASIC XL if their "into" BASIC.)

Weary of the reliable but cumbersome Atari BASIC, I was glad to hear of a new, upwardly-compatible version from Optimised Systems Software, OSS. For BASIC programmers, whether weekend or hard-core, this is a fantastic combination of language and debugging tools.

To begin with, BASIC XL will accept any sort of keyboard input, meaning that you can type your lines lower case, inverse, or even both. After changing back-and-forth from PRINT statements with lower case or inverse text, or accidentally hitting CAPS-LOWR, I find this to be one of my favorite features. The program, when listed, is in lower case with the first letter of each word or variable capitalized. There is also program formatting, with FOR-NEXT loops being indented a few spaces.

For those of you who are tired of looking up what those error codes are time after time, BASIC XL gives a brief description of the error, such as "ERROR- 170, FILE NOT FOUND" or "ERROR- 9 AT 20, DIM ERROR". Despite the brevity of some messages, I find them to be helpful memory-joggers.

BASIC XL has many invaluable debugging tools. Here are a few:

RENUM x,y--Lets you take program and renumber, with X being the new first line and incrementing by Y. Especially useful for combining two programs that occupy the same line numbers in BASIC.

FAST--precompiles the program and changes every line number to an address in memory. Then, when a GOTO or GOSUB or FOR-NEXT loop is in use, there is no line searching necessary: the program goes directly to that address, which is useful in speeding program operation.

LVAR--Gives a listing of all variables to a specified device. LVAR defaults to E:, but a LVAR P: will display all variables on hardcopy. Much better than saving a program to the editor to look at the variable name table.

NUM x,y--This is the auto-line numbering utility, with X and Y like the parameters in RENUM--first line and increment. Very handy for writing subroutines up in the 10000's lines.

TRACE/TRACEOFF--A feature from Microsoft BASIC which prints the line number of a program as it executes. If your program is so heavily TRAPPED that you can't find the

errors that are plaguing your program as they happen, then this is for you. This is another handy feature, despite the fact that it does garble the screen display a bit.

Another pair of commands I believe are also in Microsoft BASIC are DPEEK and DPOKE. These allow you to manipulate 2 bytes at a time with one peek/poke where two would have been necessary from Atari BASIC. Example:

```
Atari:?" Screen RAM at ";PEEK(88)+PEEK(89)*256
XL: ?" Screen RAM at ";Dpeek(88)
```

Much better. DPOKE is even handier due to the amount of math required under normal conditions to get a 16-bit number into a couple of 8-bit bytes.

Ever get tired of going back-and-forth to DOS to do a small but important function, and the possible hassle of MEM.SAV? Hate XIO's to the drive? Well, the folks at OSS obviously felt the same way and included several DOS functions. DIR acts just like DOS function A, with name masks and destination of directory. Default of course is "D:*. *" and "E:". Also included are ERASE, PROTECT, UNPROTECT, and RENAME.

Two new commands that aid I/O operations are BGET and BPUT. BGET is in the format BGET c,l,b. This will take B bytes from channel C and store them beginning in location L. This is a very fast way of loading machine language routines, character sets, and even player/missile graphics data. Or just a large text file, as shown below:

```
OPEN #1,4,0,"D:REVIEW.TXT"
DIM R$(FRE(0)-1024)
BGET #1,ADR(R$),20000
```

This mini-program would open up to "REVIEW.TXT" (which I happened to name this) and get 20000 bytes from the file and store them in R\$. To look at what I had done, simply ? R\$.

BPUT works the other way around: it takes a number of bytes and stores them to a channel. This could be saving a graphics 8 screen as shown:

```
OPEN #2,8,0,"D:PICTURE"
BPUT #2,DPEEK(88),40*192
CLOSE #2
```

This would find the beginning of screen RAM and dump the appropriate number of bytes (40 bytes/line*192 vertical) and voila. You've got a graphics screen on disk, editable by most graphics programs.

BASIC XL (cont.)

BASIC XL uses three functions that appear in Microsoft BASIC for string handling that I feel are a waste of program. LEFT\$, MID\$, and RIGHT\$ are functions that could be done quite simply in Atari BASIC. An example of one of these functions using LEFT\$ would be:

```
? Left$("Atari 800",5)
```

This would return the 5 leftmost characters of the string in question, in this case "Atari 800". The result of this command would thus be "Atari". I say this is a fairly useless function because it would be no harder to take a string and:

```
? A$(1,5)
```

Which would get the same thing. Of course we'd have to put "Atari 800" in A\$, but in the example I used you could tell by looking that you'd get "Atari" out of that statement. The same type or rules apply to MID\$ and RIGHT\$, in that you just pick out the areas of strings you wish to use. This is why I consider these functions to be useless. However, I am not all-knowing in this field, so if I am mistaken, please correct me.

One innovation I do like when string-manipulating is FIND. This command will search a string for another substring or character. The format for this is:

```
? Find(string, substring, begin search at)
```

So: ? Find("ABCDEFGH","EF",X) would return 5 if X=1. The number returned is relative to the start point, X. If X were greater than 5, a zero would result, indicating that no matching string was found.

One facet of BASIC XL I have mixed feelings about is the fact that strings do not have to be dimensioned. If BASIC XL finds a string not previously dimensioned, it automatically sets its maximum length to 40. This has caused me problems in writing programs that are meant to be compatible with Atari BASIC, as I load a previously working program from XL into Atari and get "ERROR- 9 AT LINE.." which means there is a string error. However, it is possible to turn off the auto-dimensioning part of BASIC XL, (although I usually forget to do so) and I shall discuss it next.

Configuring BASIC XL may also be accomplished through the SET command. SET X,Y stores register X with number Y--similar in nature to a POKE. With sets you can alter functions such as:

BREAK key--on, off, causes TRAP-able error. Default is ON.

INPUT character--default is 63, ASCII for '?'. Can be changed to anything, simply use the ATASCII code for that character.

Multiple INPUTs--When entering more than one set of things, such as 'INPUT A,B', another prompt comes up, '??' if only one thing is entered. This is the default, but it can be turned off, ala Atari BASIC to result in an error.

FORMAT--indentation of FOR-NEXT loops can be turned on and off, and the uncapitalized program listings can be made all upper case, so when LISTing to a disk file, the program will be compatible with Atari BASIC when ENTERed.

ENTER--Normally, after ENTERing a program from disk you get the READY prompt. A SET will make ENTER, upon completion, cause a TRAPPable error (136), so a properly TRAPPed program can enter lines to itself and continue operation.

COMMA TABBING--the distance a ',' causes the cursor to move when the comma is encountered in a PRINT statement. In addition to the SET, you can check the various current parameters of a SET with SYS, where SYS X,Y will return the current X register to variable Y.

IF-THEN loops have a new twist added to them, something I find powerful: the ELSE and ENDIF commands. The 'THEN' part is left out when using ELSE for some reason; it doesn't really matter, though. The format is as follows:

```
IF (condition):(statements)
  (more statements if you desire)
ELSE (statements)
ENDIF (end of IF block)
```

Another command-type loop is the WHILE-ENDWHILE loop. Again, the format is this:

```
WHILE (condition is true, <>0)
  (statements)
ENDWHILE (causes jump back to WHILE as long as condition
is true)
A very powerful set of commands indeed.
```

Controllers plugged into the joystick ports have been given a few more commands to work with. To improve on the clumsy STICK(0) readings, BASIC XL has HSTICK and VSTICK. HSTICK(stick #) returns a +1 when the stick in question is moved to the right, a -1 when moved left, and a 0 when horizontally centered. VSTICK acts the same way, with +1 returned for up, -1 for down, and 0 for centered. A sketchpad routine in graphics 8 moves far faster with HSTICK and VSTICK than any cumbersome routine involving STICK. Also included is PEN() which returns the value of the lightpen value of the pen indicated.

BASIC XL (cont)

To those of you who have found player missiles mind-bogglingly confusing from Atari BASIC with endless numbers of POKEs and PEEKs, BASIC XL offers quick and easy PMG handling. Some of the player-missile commands are:

PMCLR X--clear player/missile X any data already stored there. Equivalent of taking all locations of that player and POKEing with zero.

PMCOLOR X,Y,Z--color of player/missile, where X is the player, Y is the luminance, and Z is hue. Again, another sort of auto-poke.

PMGRAPHICS X--Sets PM graphics mode, zero for off, one for single-line resolution, and two for double-line resolution.

PMMOVE will allow you to move a player/missile anywhere on a screen instantly. You give the player, movement left/right, and movement up/down. What could be easier?

The BGET and BPUT mentioned above can be used to bring a player straight from disk into RAM, enabling instant p/m loading.

The final comment: should you shell out the \$60 or so bucks for BASIC XL? I did, and have not regretted it once. I've heard some say that they're so accustomed to Atari BASIC that they'll stick with what they've got. However, anyone who works with BASIC XL and prefers Atari BASIC has gotta be nuts.

Note: If you buy this, don't sell your old BASIC cartridge! A few machine-BASIC hybrids such as Jupiter Mission and Temple of Apshai don't seem to get along with XL, as well as some autorun routines.

A Programming Quickie

Here's a neat trick that I recently ran across. If you poke a 1 (or any other non-zero value) into location 580 (decimal), the system will perform a "cold start" on the next pressing of RESET.

In BASIC this looks like

```
POKE 580,1
```

then press RESET.

The effect of this is to completely reload DOS and start from ground zero, just like powering up the Atari for the first time. This is even better, because it doesn't wear out the power switch!

Try it sometime.

THE FACEMAKER

Spinnaker Software, \$18.95 KMART

Review by Jeff Golden

If you have a 5 to 8 year old child who is still learning to use a computer, then this may be the program.

To an adult, the program seemed so simple-minded, that I never expected it to occupy an above average child's mind for more than about 15 minutes.

Much to my surprise, my seven-year old and her friends played with it for over a week, and thoroughly enjoyed it. So much for adult thinking when it comes to the ways of children.

The program comes in a cartridge, which is perfect for the kids. It should work on just about any variety of Atari home computer including the XLs. It is menu driven and is controlled from the keyboard.

The program is broken up into 3 functions that are activated by entering a 1, 2, or a 3.

Function 1 allows the child to construct a face using a limited selection of eyes, ears, noses, mouths, and hair styles.

Function 2 allows the child to make the face smile, frown, wink, cry, and wiggle its ears, by using one letter commands. It is also possible to program the face to perform a series of expressions.

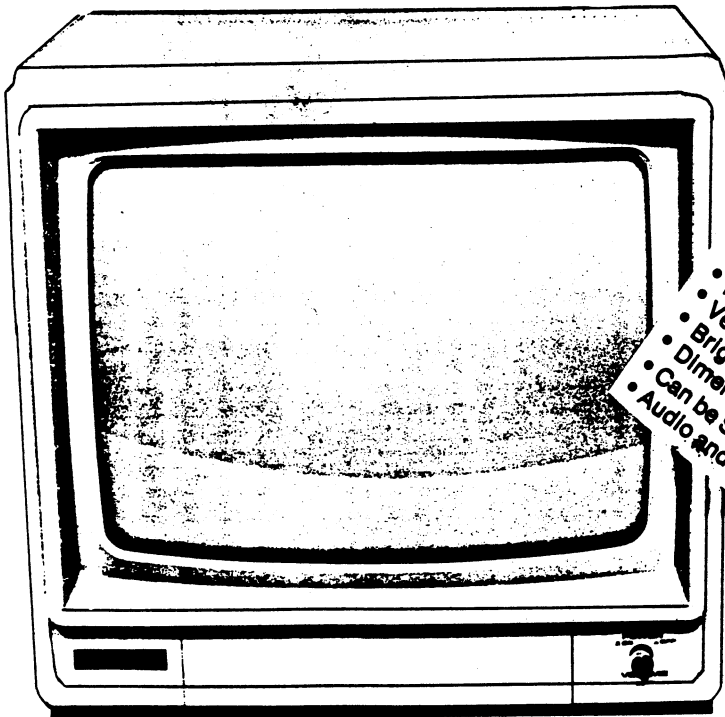
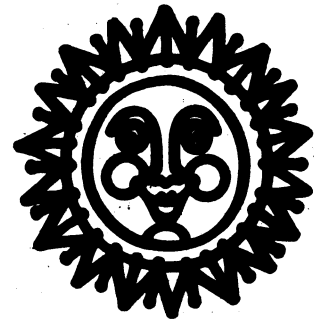
Function 3 is a game whereby the computer displays a series of expressions, pauses, and then asks the child to repeat the sequence. The sequence increases in length by one step each time the correct answer is typed in. With a little practice, my kid was able to handle up to 12 steps. I won't tell you where her old man dummed out.

It would have been nice if they had provided a little more variety in the selection of face parts, and a little better representation of the expressions. (A frown consisted of a downturned mouth). It also would have been nice if the child could have drawn in some face parts with a joy stick, but then, would they have enjoyed it as much as they did the simple-minded program?

At any rate, \$18.95 for a cartridge program is hard to beat.

WOTIP

FOR SUMMER!



**COMPUTER MONITOR
14" COLOR, COMPOSITE
MODEL THT-4501C**

- Power On-Off Volume Switch
- Vertical Hold
- Bright, Tint, Color Contrast, Focus Controls
- Dimensions: 14 1/3" x 13 1/3" x 14 1/3"
- Can be Switched from Color to Green
- Audio and Video Inputs and Outputs

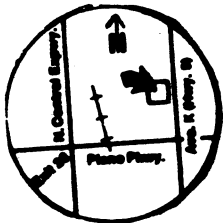
\$239.95

MCP

METROPOLITAN COMPUTER PRODUCTS, INC.

801 Ave. K, Suite 4 Plano, Tx. 75074

(214) 578-8504



Open M-T-W-F 9-6 p.m.
Thurs. 9-9 p.m. Sat. 10-4 p.m.



NEXT MEETING SEPTEMBER 1, 1984

---> SIG <---

"SIG" is the acronym for "Special Interest Group". These groups are composed of members having a common interest in a specialized area of the general computing field. These specialized areas range from languages, to applications and special features of the ATARI. Join in with one of the groups listed below and get more out of your computer! If you would like to start a new SIG, send your name, phone number, and a description of the subject to Jim Chaney, DAL-ACE, 916 E. Berkeley, Richardson, Tx 75081. Deadline for all SIG information is the 15th of the month.

---> FORTH <---

The next meeting of the FORTH SIG will be held in September. Contact Eric Weeren at 245-7429 or Fred Sagor at 424-0291 for further information.

FORTH SIG meetings usually consist of two parts: (1) an exchange of information about the FORTH computer language, especially on ATARI computers, and (2) a general show-and-tell discussion about anything relating to ATARI computers. Everybody is invited!

---> BUSINESS <---

The BUSINESS SIG is interested in applications such as Word Processing, Bookkeeping, Spreadsheets, Accounting, and Other business applications. Call Rich Greenlee at 267-7428 (Metro Number) for more information on this SIG.

---> EDUCATION <---

Information about the next meeting of the EDUCATION SIG was not available at "press time". Contact one of the EDUCATION SIG members for further information.

---> GRAPHICS <---

Members interested in the GRAPHICS SIG, please contact Sandra Stephens, 827-0493.

---> CP/M <---

The next meeting of the CP/M SIG will be held Tuesday, AUGUST 7th, at 7:30 P.M. in the home of Jim Chaney, 916 E. Berkeley, Richardson. Any questions you might have, please contact Ed Bohneemann at 495-1803. Hope to see some new faces at this next meeting.

--->TELE-COMMUNICATIONS <---

Contact Ernie Runyon (817)485-0871 for additional information on this SIG.

RICHARDSON CIVIC CENTER - RICHARDSON, TEXAS
DIRECTIONS: From LBJ (635), take Central Expressway (Highway 75) North to Arapaho Road in Richardson. Turn left (West) on Arapaho Road. The Civic Center is on the left between the Hwy 75 service road and the Richardson Public Library. Guest are WELCOME!!

MEETING AGENDA

- NOON TO 2:00 SALES (VENDORS & CLUB) DEMONSTRATIONS NEWSLETTER DISTRIBUTION NEW MEMBER REGISTRATION GUEST REGISTRATION SOCIALIZING
2:00 TO 2:30 VENDORS CLOSE AND CLEAR SET-UP CHAIRS FOR MEETING
2:30 TO 3:00 BUSINESS MEETING CLUB SALES
3:00 TO 3:30 GENERAL QUESTIONS CLUB SALES
3:00 TO 4:00 TECHNICAL QUESTIONS CLUB SALES
4:00 TO 5:00 DEMONSTRATIONS CLUB SALES

VENDOR RESERVATION

Vendors may reserve table space prior to the meeting by calling Jim Chaney (231-4402). Fee collections and table assignments will begin at 11:45, after which the vendors may begin to set up their areas. Those tables which have been reserved but not yet claimed by 12:30 may be purchased by other vendors at that time. Prepaid reservations will not be released. Space assignments will be on a "first come, first served" basis, unless prepaid. The current meeting facility allows for approximately 30 vendor tables.

NEWSLETTER ADVERTISEMENT

Personal Classified ads will be published free of charge for current members. Commercial rates are \$35.00 per full page (7 1/2" horizontal by 9" vertical), \$25.00 per half page (7 1/2" horizontal by 4 1/4" vertical), and \$15.00 per quarter page (3 1/2" horizontal by 4 1/4" vertical). Commercial ads must be camera ready. The deadline for all ads is the 15th of the month. Mail or deliver copy to DAL-ACE Newsletter, 916 E. Berkeley, Richardson, Texas 75081.

** DISCLAIMER **

The articles and advertisements contained in this newsletter reflect the opinion of the respective author. Members are encouraged to offer opposing opinions on any subject (relevant to computing) at any time. We will not, knowingly, publish fraudulent or malicious material. The purpose of this newsletter is to present information for your consideration ... you, the reader, are the final judge on any product or advice presented.

--EDS

**
** V V I DDD EEEE OOD L AA N N DDD ! **
** V V I D D EEE O O L A A N N D D ! **
** V V I D D E O O L AAAA N N D D ! **
** V I DDD EEEE OOD LLLL A A N N DDD ! **
**
** LOCATED ON PRESTON ROAD ACROSS FROM VALLEY VIEW **
**

DAL-ACE OFFICERS

PRESIDENT..... JIM CHANEY 231-4402
VICE-PRESIDENT..... DOUG VERKUIL 462-0843
V.P. COMMUNICATIONS .. ERNIE RUNYON 817-485-0871
SECRETARY..... SUSAN HENDERSON 817-497-3877
TREASURER..... ED KOBUS 492-2922
MEMBER-AT-LARGE..... BOB DAIN 350-7171
MEMBER-AT-LARGE..... DAVE GILLEN 245-2732
MEMBER-AT-LARGE..... JOHN OLSON 278-0814
MEMBER-AT-LARGE..... JEFF RUTHERFORD 214-727-2945

DAL-ACE VOLUNTEERS

NEWSLETTER EDITOR..... MYRON WALTERS 691-3401
EDUCATION..... TANDY PENN 276-8796
BULLETIN BOARD..... ERNIE RUNYON 817-485-0871
DISK LIBRARY..... JOHN CANEDY 790-4425
PUBLIC RELATIONS..... SANDY PARK 680-3120

DAL-ACE BULLETIN BOARDTEMPORARILY DOWN

DAL-ACE DALLAS ATARI COMPUTER ENTHUSIASTS

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

DAL-ACE membership is \$16.00 per year. This newsletter is written, edited, and published by club volunteers. Its availability and/or distribution may, at times, be subject to circumstances beyond the control of the club officers. Members will note that their membership renewal month appears as the first three (3) letters on the address label.

Other ATARI user groups may obtain copies of this newsletter on an exchange basis.

ALL MATERIAL IN THIS NEWSLETTER MAY BE REPRINTED IN ANY FORM, PROVIDED THAT DAL-ACE AND THE AUTHOR, IF APPLICABLE, ARE GIVEN CREDIT. LIKEWISE, PORTIONS OF THIS NEWSLETTER ARE REPRINTED FROM OTHER NEWSLETTERS AND ARE SO NOTED.

DAL-ACE
916 E. BERKELEY
RICHARDSON, TX 75081

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

