

SPACE Newsletter Feburary 1995

Newz From The Prez
by Nathan Block

Nathan has failed again to submit his article in time for this publication!

The Editor

DOM Info by Terry Streeter

Terry has failed to submit DOM info to me in time for the newsletter.

The Editor

Treasurer's Report by Greg Leitner

Although the January meeting got off to a slow start. by the time we set down to have our fornal meeting there were a good representation of our membership present. By the end of the meeting we even had a visit from the former President of MAST who offered his vast library of ST disks to our Club which we gladly accepted. I hope we will see more material on the ST at future meetings as I would rather use my ST over my IBM any day.

This January meeting was very interesting in that most of the discussion centered around the using of modems to connect to various services, especially the Internet. I can't wait for future meetings where we can see from current users just how these services work and how to negotiate them. I for one have never had the time to sit down and learn about my modem, so I welcome the help from anyone who knows the ropes. This is exactly the kind of help I was looking for when I joined the Ateri Club and now I plan on taking advantage of this opportunity.

Thanks to the attending members who continue to support our Club. We had very good receipts for January which included four memberships (two new ones), disk box and blank disk sales along with our regular Dom sales. All together we had receipts of

\$122.00, and with only the December BBS phone bill and the January Newsletter expenses for the month our total gain in our account was about \$87.00. This brings our balance up to \$465.75 for the end of January 1995.

Plan on attending the upcoming meetings and see what your Club has in store for you. With the addition of an ST disk library and tutorials on using your modems, Ateriwriter etc. exciting times are in store for all of us.

Seek & Find

ZEALOT

Z W X E E R V T B T B Y N M U V P Q
Q Z E A L O T V E E W X Q Z S Y N M
P L M Q E T E O L J G B A E E B M R
Z D H E L W V E X R O V C Y Y B M R
M N B V L X O D T O L T E B M W R A
G C L P V P R D S O A X H R D R T E
J X H E W Q I T A R V X V H J S W G
L M W A F V E C I W F E F E I W L P
W Y T U M R X A S F O A D M M Y A R
N T A D G P W O U I I I E E B M C V
W O D T H L I B H A D R C V N D I I
L G J P Q D W O T A T U N I U E D I
D I H O V U Y I N X W Q L X F H A R
X B W R B V C W E B T C I D D A R E
X G Y Q J L D G T W P E M L U P A E

Fanatic	Enthusiast	Bigot	Sectarian
Extremist	Radical	Devotee	Champion
Disciple	Aficionado	Fan	Buff
Nut	Booster	Addict	

News From The Publisher:

I tried to download the SPACE minutes from the New SPACE BBS, but it seems there is a upload/download ratio now? I havn't uploaded anything yet so I couldn't download the minutes from the last meeting or another article that was there waiting for me to get!

This will have to be addressed at the next meeting.

Larry's ACTION! TUTORIAL

#3 The First Mistake

Let's begin with your computer on, and the HELLO WORLD program in the Editor. Add or type in the follow text;

```
PROC stop()
PrintE("End of program.")
RETURN
```

```
PROC hello()
PrintE("Hello World")
stop
RETURN
```

Now press <CNTRL> <SHIFT> 'M' to get to the Monitor. Enter 'C' to compile the program. You will then be exposed to a compilation error. At this point, the program will not run. Note the error number and open your manual to the Error Code Appendix. You can try to guess what the error is if given the correct line, however, sometimes you are not given the correct line and you must decide what to look for by the description supplied in the Error Code explanation.

If text was displayed in the Monitors' message area, the cursor would be moved to the start of that same line. If no text was displayed, the cursor is right where it was when you left the Editor. In my case, I had no text line displayed in the Monitors message area, only the error number. Your cartridge may be a different version, and act slightly differently.

The manual explains this error as an Illegal assignment. It is also an Illegal function call, for which there is no other error number. In my manual I have added in the words "Jump should be Jump()" to remind me to look for this type of error too. Other than this ommission, I believe all the error code explanations are right.

To correct the error, add the parenthesis after the word "stop" ("stop()") in the 'hello' procedure, then return to the Monitor to compile it again. It should compile correctly now, if it doesnt, recheck your text to find any mistakes. If everything is correct, go ahead and run it. Everybody makes mistakes, especially when first learning the language. Here

are the typos you should be aware of in the August 1983 Action! manual:

- = This is the line to look for.
- + = This is how it should be.
- > = Correct the notes for this line

MONITOR / RUN - Program Execution

>RUN "<filespec>" only works when used on programs saved from the Editor. (When it is still in text form.)
-RUN PrintE() <RETURN> does not work.
+EXECUTE PrintE() <RETURN> is correct.

LANGUAGE / Arithmetic Expressions

>TECH NOTE Using '*', '/', or 'MOD' will not work correctly on large CARD values.

LANGUAGE / Record Manipulations

-TYPE idinfo=[BYTE level,
+TYPE idinfo=[BYTE level (omit comma)
LANGUAGE / Advanced use of Extended Types
>newrecord=idarray+(reccount*recordsize)
The pointer will point to the next record not the end of the array.

>In the main procedure;

-"mode=InputB()" won't work correctly.
+"mode=GetD(7)" works fine.

LIBRARY / The PUT Procedures

-PROC PutDE(BYTE channel, CHAR character)
+PROC PutDE(BYTE channel) will compile.

LIBRARY / BYTE color

>For Graphics(0) and text windows:

+Chr. luminance (color number = 1)

+Background (color number = 2)

LIBRARY / PROC Fill

>col and row must be the numbers of the lower LEFT corner of the box.

LIBRARY / PROC SCopy

>The description is wrong, DEST should be dimensioned larger than source, or else identify an array with enough space.

While in the Monitor, enter the line:

? \$B000<RETURN>

The '?' requests the Monitor to display the contents of a memory address. My cartridge displays:

45056,\$B000 = 6 \$0136 54 310

This means I have version 3.6 which is the last version I am aware of. If yours is 7 or greater, LET ME KNOW!

Until next time!

Larry's
ACTION! TUTORIAL

#4 Sing-a-Long!

One more aid in correcting errors is the use of the List option during compilation of your program. To use this option, you can go to the Monitor and enter 'O' for the Option Menu. Continue pressing RETURN (4 times) until the prompt "List on?" is displayed, then enter 'Y' <RETURN>. This option tells the compiler to list the program in the Monitor message area, as each line is being compiled. If it does encounter an error, the listing of the program stops in the approximate location of the error. Whatever caused the error may be listed on the screen, plus all the preceding lines that compiled before the error. This can narrow your search down to about 22 lines of text, instead of how ever many lines there are in your program. Using the List option does slow down the compilation process. You can use it dynamically (you can turn it on and off) during the compile process by including a 'SET List=1' (on) and 'SET List=0' (off) in the text of your program. Try this out to see how it performs. Here is a short program to finish off the section on the Action! Editor and Monitor;

```
MODULE
PROC versel()
    PrintE("You say good-bye,")
RETURN

PROC verse2()
    PrintE("and I say ""hello!""")
RETURN

PROC song()
PROC main()
BYTE ch=764

Graphics(0)
PrintE("PRESS ANY KEY;")
song=versel song()
do until ch<255 od
ch=255
song=verse2 song()
do until ch<255 od
ch=255
RETURN
```

This example gives an indication of why the parentheses are needed in any call to a procedure or function. (Remember Error

10 from the last lesson?) Type it in just as it appears, compile, and run it. In this example, the word 'song' is used both as a procedure name and as a variable. The only distinction between the two uses are the parentheses following the variable name:

```
song=versel      ; as a variable
song()           ; as a procedure
song()=versel()  ; will not compile
```

Here is another example program:

```
PROC main()
BYTE ch=764,atachr=763,tmp

PrintE("KEYBOARD TEST PROGRAM")
PrintE("Press any key;      Esc=exit")
do
    while ch=255 do od
    tmp=ch
    GetD(7)
    PrintF("ch=%U atachr=%U (^[%C)%E",tmp,
        atachr,atachr)
until tmp=28
od
RETURN
```

The above example illustrates one method of testing for a keypress. In the main procedure, the BYTE variable 'ch' was used to test for a keypress. The operating system from ATARI uses one address to hold the keyboard value of the most recent key pressed. From this location the computer can tell when you have pressed a key, and will then translate it to ATASCII. The address of this location is 764 or \$2FC. Action! allows the assignment of addresses to variables, arrays, and procedure names. By declaring the variable to be residing at address 764, the program can watch for any keypresses, just as the computer does. When no key has been entered, and after each translation the address will be set at a value of 255.

Location 763 holds the ATASCII value of your keypress, after it has been entered and translated.

Because the GetD(7) statement will reset ch to 255, tmp was used to temporarily store the value so that it may be used in the PrintF statement. You can use this example to compare the ATASCII values with the actual keyboard values.

Next month, Vo-cab-u-lary!

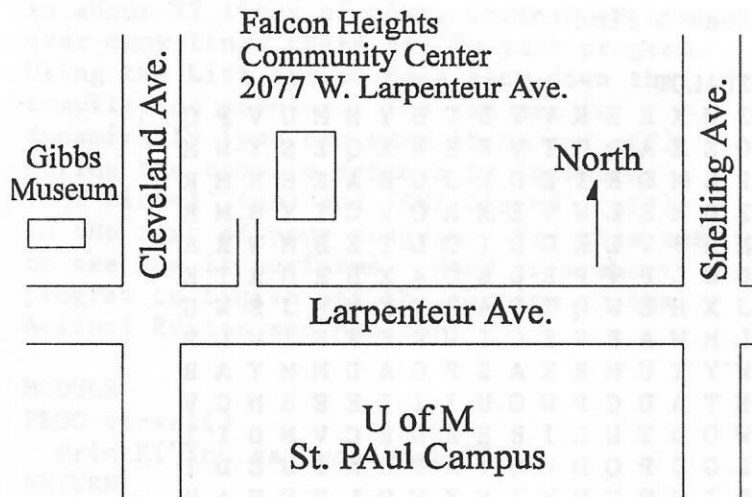
CLUB OFFICIALS

President:	Nathan Block	922-8012
Vice President:	Sherm Erickson	434-5254
Secretary:	Mike Fitzpatrick	XXX-XXXX
DOM Librarian:	Joe Danko	777-9500
Paper Librarian:	XXXXXXXXXX	XXX-XXXX
Software Librarian:	Earline Fitzpatrick	xxx-xxxx
Membership Chairman:	Glen Kirschenmann	786-4790
SPACE Treasurer	Greg Leitner	429-4909
SPACE Newsletter Editor:	Michael Schmidt	757-4192

DISCLAIMER

Published by the Saint Paul Atari Computer Enthusiasts (SPACE), an independent organization with no business affiliation with ATARI Corporation. Permission is granted to any similar organization with which SPACE exchanges newsletters to reprint material from this newsletter. We do however ask that credit be given to the authors and to SPACE. Opinions expressed are those of the authors and do not necessarily reflect the views of SPACE, the club officers, club members or ATARI Corporation.

Saint Paul Atari Computer Enthusiasts (SPACE) meets on the second Friday of each month at 7:30 PM in the Falcon Heights Community Center at 2077 West Larpentour Ave. Doors open at 7:00 PM.



SPACE BBS

BBS Phone: (612)-427-4317

Sysop: Mike Fitzpatrick

Co-sysop: Sherm Erickson

Supporting:

ATARI 8-Bit Computers

S.P.A.C.E.
P.O. Box 120016
New Brighton, MN 55112

