THE GUILFORD 99'ER NEWSLETTER

VOL.7 NO.1 JANUARY 1990

Bob Carmany, Pres.(855-1538) Emmett Hughes, Vice Pres.(584-5108)
Mack Jones, Secy/Treas(288-4280) Bill Woodruff,Pgm/Library
BBS: (919)274-7000(OPUS) ROS: (919)621-2623

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (George von Seth, Ed.:292-2035)

OUR NEXT MEETING

DATE: Jan 2, 1990 Time: 7:30 PM. Place: Glenwood Recreation Center, 2010 S. Chapman Street.

A party! That is the program for this meeting! Bring your favorite "munchies". The Users' Group will provide soft drinks. Celebrate the New Year (our seventh) with good cheer and the companionship of one of the oldest TI Users' Groups around!!

MINUTES

The December 5th meeting of the Guilford 99er Users' Group was held at the Glenwood Recreation Center in Greensboro. There were 8 members present.

The meeting was called to order at 7:40 P.M. by Vice President Emmett Hughes.

Old Business:

1. There were no nominations from the floor as to the choice of officers so the standing list of officers will start at the January meeting. Those are as follows:

President...Bob Carmany
Vice President..Emmett Hughes
Sec. Treas. .L. F. "Mac" Jones
Newsletter Editor..George von Seth
Library ...Bill Woodruff

New Business:

1. The book that Herman donated to the club was auctioned off and was bought by George von Seth. George also made a $3.00 donation for using the inker.
2. Bonnie Jones made a motion that the club have a New Years party at the January meeting. The motion carried and the secretary was asked to furnish the soft drinks from the treasuary. Members will bring goodies.
3. Two members paid their 1990 dues. George von Seth and Bonnie Jones who also gave a donation over her dues.
4. Bill Woodruff showed a booklet and disk that he had ordered from the Chicago users' group. The articles were pretaining to TI/Writer.

The program was a demo by Bob Carmany on his PROGRAM WRITER that he has written. The program saves many many key strokes when entering program lines in extended basic. A very good program Bob and thanks.

The meeting was adjourned at 9:00 P.M.

Respectfully submitted,
L.F. "Mac" Jones Sec./Treas.
Guilford 99er Users' Group

PRES NOTES

By Bob Carmany

As we start yet another year of the Guilford 99'ers, it is well worth noting that our little band of TI fanatics is one of the oldest UG's in existence. This issue of the newsletter marks the 7th calendar year of publication --- all without missing a single issue. Somewhere in mid-August, we will actually enter our 8th year. There are 't many of us left (George and I, I think) who remember the very beginning at the TV place on Battleground Ave and Cone Blvd. During the years, we have been through a couple of name changes, several different meeting places, and several slates of officers. Throughout it all, our interest in the TI has survived! Amazing for a machine that has been orphaned for 5+ years!!

There have been some interesting developments in the software arena lately! I was one of the fortunate few who got a glimpse of the "beta-test" version of the latest effort of the McGoverns --- a slightly early release of F'WEB 4.20. To be sure, there was the promise of outstanding new utilities like DISKREVIEW but the bugs in the "beta-test" version were significant! Undaunted, Tony quickly squashed all of the offending "beasties" and released F'WEB 4.21 (all bugs fixed). A truly amazing piece of software!!! It is really astounding how a single programmer can produce innovative additions to an already excellent program with such regularity. I thought long ago that all that could be added to the program had been. Hats off to Tony for another outstanding revision of an excellent program.

As many of you know, I am headed for the "Land Down Under" next month for vacation and maybe to "plow the froth off a couple". I hope to spend a couple of days examining the "cookery" that Will and Tony use to produce all of these fine programs. The fact is, the HV99 UG has some of the most innovative programmers around. Besides the McGoverns, there is Richard Terry, MD (and Forth programmer extrordinaire), and Ron Kleinschafer (of GED program fame). At any rate, I hope to meet as many of my "mates" as possible in the three weeks that I'll be there. There will be a full report in the April newsletter.

RAMBLING BYTES

By "Mac"

Whoever the gentleman is that writes the predictions for the Almanac sure must have had his map turned upside down when he predicted that all the rough weather this winter would be to the West and North East! Man are we getting it! Seems nice tho' sitting here by the fire with nothing to do but hack on ol' faithful. Just hope I don't fall asleep before I get this to the board so Bob can get it downloaded.

It seems that Tetris is taking the country by storm. My grandson who has a Nintendo just got his version and believe it or not it is an issue from the Soviet Union. The one I have for the TI was written by a gentleman in Germany. I honestly think the TI version is better but that is just a personal opinion. I had to get Bob to translate the docs for me as my German is very limited but basically, the game is very user-friendly. I think one could get by without reading the docs.

The group will hang in there for another year it seems. I was hoping we would but you never know until that December meeting just how the members feel about going on. There is one good thing that has come out of the decision to continue. There has been offers to help the Editor by submitting articles and also offers to give demos. This is what holds a club together gang. Thank you in advance for the input.

Can you "expanded" users remember the hassel we used to have to put up with to load a program? The "OLD CS1" bit? Then, if it was a very large program, we would have time to walk around the block while the tape loaded. Then, sometimes, having the console "beep" and the hateful message "NO DATA PRESENT" flash on the screen. Egad how I used to hate that message! Then in 1984 I was offered a box with a disk drive and memory expansion at a reasonable price (for 1984 that is) so I robbed my piggy bank and jumped on it. I remember the astonishment at the speed my programs would save and load with that wonderful disk drive. Then came the knowledge of just what I was doing a little better so a pair of half highth Teacs were added to replace the single drive. I thought that was also wonderful since you didn't have to change the disk while copying texts. Now with the ram-disks being added, you wonder just where this little orphan is going to wind up.

I realize there are others who just have to have something faster with more enhanced keyboards, but this little bugger pleases me to a T. I would be afraid to open up an IBM or clone, but I have had my TI open so many times I have thought about adding little hinges on the RF shield.

In closing I would like to wish each and every one a very, very MERRY CHRISTMAS and a very safe and wonderful NEW YEAR. There have been a lot of programs for Christmas and New Years Eve, but the very best program I can wish for is to see all of our members at the January meeting for the goodies and good fellowship. Until then, enjoy the good TImes.

XB TUTORIALS PART 5

By Tony McGovern

VII. ACCEPT AT and other RAMBLINGS

TI Extended Basic is a very substantial language. The XB cartridge contains 12K of ROM and 3 and a bit (the 4th one isn't full) GROMs at 6K apiece. This is on top of the 8K of console ROM and whatever parts of the 3 console GROMS are still used in XB. The tragedy of the TI-99 is that GROMS and GPL were ever invented. I guess it was TI's way of trying to keep the software market sewn up. The end result as we all know is that they shot themselves in both feet with uncanny accuracy. Instead of using the TMS9900 CRU addressing to bank switch plain ordinary ROMs or even just using GROMs only as sources of code to load into RAM (as I believe is done in the p-code card), they could have had a machine that did justice to its CPU, a real home minicomputer ..... that's all past history now.

I have been pondering on what TI should have done way back when the 99/4 was first designed, that could have been easily done at the time (or even when it was updated to the 99/4a). My conclusion is that the machine should have been given 4K of fast 16-bit CPU RAM instead of a measly 256 bytes. There would have been plenty of room with a little rearrangement and/or better decoding of memory-mapped devices (VDP, sound, speech, GROMs). This would have meant that Basic and XB system areas, sprite tables, full screen buffers, string buffers, value stack, and so on could have been in fast RAM, and even console Basic could have had full scope for character and sprite definitions (as in TI-LOGO for instance). Their cartridges could then have easily been a lot better, and let's face it, many of the earlier ones were pretty hopeless, and the later ones are all limited by lack of honest CPU RAM. The only cartridges which have stood the test of time are those that use the 32K RAM expansion. TI would then have never been dragged into that marketing war to the death (TI's that was) with that vastly inferior machine, the VIC-20. I have a suspicion that the 256 bytes happened because part of TI management wanted to protect their existing evaluation board and smaller minicomputer business.

The immediate improvement really needed in XB sub-programs is a means of examining variable values in any sub-program when program execution is halted by BREAK or errors. TI should have done it in XB by retaining the EDIT command of console Basic, allowing it to access user subprograms by name. Anyone listening out there? If so add single command array operations, full syntax checking on entry, 80 column display capability with formatting power to match, bit-map screen functions, fast program execution and anything else will then be gravy. Then TI-99/4a owners will be most pleased to join in. The bad news is that TI is starting to cut back on support for the 9900 family despite its excellent qualities, and so it is becoming less attractive for new designs. In retrospect we still don't have these things in Basic, as the Geneve Basic remains incomplete and buggy (mid-89).

Enough ramblings and back to the tutorials ! What then is the most powerful feature in XB after SUB and CALL? A good candidate is the file system, but as this is already built into the console I will stick with commands specific to XB. The prime candidate is ACCEPT AT and its qualifying clauses (even just plain ACCEPT has some interesting improvements over INPUT but that has been treated elsewhere). This was emphasized by the recent appearance (mid-84) in a computer magazine of a long article on machine code for adding this function to IBM PC Basic (which doesn't have sub-programs either). ACCEPT AT is very useful and powerful, but has some undocumented features as well as some subtle and treacherous bugs, and is well worth talking about in this series.

The simplest level of ACCEPT AT combines the INPUT routine with its access to editing features, with cursor positioning on the display screen by the AT clause. So far this is just the input version of DISPLAY AT. The difference from INPUT is that there is no provision for prompt strings, but a DISPLAY AT soon fixes that. It also accepts input to a single variable only, and not to a whole variable list. As ACCEPT AT and DISPLAY AT do not scroll the screen, their repeated use can give a much better effect than INPUT when graphics elegance is important. Construct your own examples here or work the XB manual examples. Remember that the cursor is in XB color group 0 if you are trying to dress up the graphics.

BEEP allows an audible prompt with only one program byte (we'll talk about program length later on if this series keeps going long enough). Of course constant repetition of beeps can get a little wearing. The ERASE ALL clause provides an alternative to CALL CLEAR for clearing the screen. As compared with CALL CLIAR, ERASE ALL is slower to execute, (it seems to be line at a time) but takes less program space. Its effect is slightly different also. This little program which uses ERASE ALL with DISPLAY will make both speed and screen effects easy to see.

```
100 CALL CLEAR :: CALL COLOR(0,3,3)

110 FOR I=1 TO 100 :: CALL CLEAR :: NEXT I

120 FOR I=1 TO 100 :: DISPLAY ERASE ALL :: NEXT I

130 CALL SCREEN(11):: FOR I=1 TO 1000 :: NEXT I
```

That's the simple pieces of ACCEPT AT -- now it starts to get interesting. VALIDATE allows the programmer to decide what characters are acceptable in a response. The computer honks (that's the word in TI-FORTH) at unacceptable inputs. Three predefined types are available. UALPHA accepts only upper-case alphabetic characters -- very useful for filenames and suchlike. This is not quite the same as depressing the alpha-lock key as it it only accepts letters, and so is incompatible with input to a numeric variable. If you are in the habit of verifying wet paint signs by touch, try that for a change. The DIGIT type does just what its name implies, and NUMERIC allows the input of any floating point number as well as plain positive integers. As with INPUT, all numbers are acceptable to a string variable, but numeric variables are fussier.

Now what if these predefined types aren't right for what you want ? Suppose only digits 1 to 4 are acceptable, as in a menu choice of 4 items labelled 1 to 4. In console Basic extra lines of code would be needed to check the input, but ACCEPT AT handles this with the clause VALIDATE("1234") or VALIDATE (I_LIKE_IT$) where the string variable has previously been set to "1234". To put it more formally, only the characters in the string argument of VALIDATE can be entered at the keyboard to be ACCEPTed.

The SIZE clause allows ACCEPT AT to be used with almost no interference to screen displays. It blanks out the specified number of characters, providing an input window of finite length, and if the length specified is negative, the characrers already in the window are not erased, and form an immediate input for ACCEPTance. This is very handy for making default choices obvious to the user. Let's enter a little program to get at the essentials.

```
100 CALL CLEAR :: DISPLAY AT(12,1):RPT$("_",28)

200 ACCEPT AT(12,2)SIZE(3):A$

300 DISPLAY AT(15,2):A$;LEN(A$)

400 CALL KEY(0,K,S):: IF S>0 THEN 100 ELSE 400
```

You most likely have the Alpha-lock depressed. If so let it off, and RUN our little program. Just press ENTER the first time round, next time hit <space> first, and finally <space> first before hitting another key. This shows that <space>s after the last honest character entered are ignored. Try some VALIDATEs here too, if you wish. Now with the program as given, alter SIZE(3) to SIZE(-3). It now ACCEPTs whatever is in the was or is placed in that 3 character input window.

Now that's all very simple, but it brings us to the edge of the undocumented wilderness. Alter the CALL KEY(0,K,S) in the last line to CALL KEY(3,K,S) and RUN the program again, this time entering letters. Observe what happens the second time around. This answers the question of what keyboard mapping ACCEPT AT uses -- like CALL KEY(0,K,S) it uses the last one, whatever that was. Try split keyboard units in the last line. At the machine code level, a particular byte in the CPU scratchpad RAM has to be set to the key unit before calling the SCAN routine. I interpret the behaviour as showing that in the XB modules of my experience that ACCEPT AT does not alter this byte. The XB manual however does not document this behaviour at all. If XB weren't a dead language that would be a caution signal. It does need to be watched in your programs, if your last CALL KEY wasn't the key unit you want for ACCEPT AT. On the positive side you can control ACCEPT AT with a prior dummy CALL KEY to ease input for the user. An example is when a program requests input of a filename, setting the key unit to 3 makes letters come out as upper-case while still allowing other characters. Brian Rutherford of HV99 first brought the anomalous behavior to my attention and has turned up other minor undocumented variations in the use of ACCEPT AT. .

Now that's not too bad, but there is worse to come. Insert a VALIDATE("123") clause in the ACCEPT AT and RUN the program. No problems there with SIZE(3), but SIZE(-3) is trickier. You can't enter invalid characters from the keyboard but unaltered "_"'s slip through. The VALIDATE appears to be exercised as characters are entered from the keyboard, and not as the edit buffer contents are transferred into the target variable. The decision to ignore trailing blanks in the input window is taken then however. Presumably a negative SIZE pre-loads the edit buffer with the screen window contents without doing a VALIDATE check. Ultimately this is not a real problem since the programmer can control what is on the screen before ACCEPT AT is invoked. Once again, the XB manual does not bind ACCEPT AT to work this way.

This behaviour does leave a weak spot in ACCEPT AT which can only be considered as a bug, but not an intractable one. Suppose you have a menu choice of items, say 1-4 by number, with default 1 pre-loaded in the SIZE(-1) window, and a VALIDATE("1234") clause to ensure proper entry for a numeric variable . What can possibly go wrong ? An evil-minded program tester would immediately delete the default using FCTN-1. An attempt to enter the blank will then cause the screen to scroll with a WARNING message. This is not a fatal error, but might as well be if your background is a carefully composed graphics screen. The workaround for this problem is not difficult, but the best one also resolves an even worse bug, so I will leave it for a little while. I do consider suppression of error trapping or warning messages by global ON ERROR or ON WARNING to be poor programming practice. The best safety net is one that is never used, only tested.

Now go back to the original sample program and change every every A$ to an array element A$(2). Default dimensioning will do. Nothing changes. Next alter your A$(2) in the ACCEPT AT to A$(1+1). Now it works only if there is also a VALIDATE clause, but the SIZE window is disabled and input can even spill over into the next line. No, it's not useful as a multiline ACCEPT ! The solutions to this and the previous problem are the same --- always ACCEPT into a temporary simple string variable, and then process the return, and do not ACCEPT a numeric directly or ACCEPT into an array element with computed index. Both of these problems were turned up by my testing crew during the writing of TEX-BOUNCE, and served as a reminder that program testing should never be left to the author of a program. The same holds true for writers of languages!

Might as well keep on going with the entomology lesson. The sub-program CALL ERR fails to clear errors when the DSR routine cannot find the external device, as in attempts to access an empty disk drive. The work-around this problem is to have a second bash at CALL ERR after further trying for a file on the device which failed to OPEN. The OPEN cannot be CLOSEd without crashing the program or invoking this extra step to flush out the Peripheral Access Block.

The instruction ON BREAK NEXT is useful, particularly in games, for disabling the FCTN-4 (BREAK) key action. However a CALL SOUND with duration greater than 33 over-rides that. Just why is not so far obvious to this outside observer.

PROGRAM WRITER

By Bob Carmany

At the December meeting I gave a demo of a "beta-test" copy of Vn 4.0 of my PROGRAM WRITER software package. After I got it home, I thought of a couple (?) more things that I could add to the menu. There was some additional error trapping added and a general re-organization took place over the next few days. The results was a smoother-running and more comprehensive package.

After the configuration prompts, WRITER now displays the following menu:

1) DISPLAY AT
2) ACCEPT AT
3) CALL STATEMENTS
4) CALL STATEMENTS II
5) CALL LOADS
6) MISCELLANEOUS
7) SAVE AND EXIT

There has been no discernable change in the first three menu options. All of the DISPLAY AT and ACCEPT AT options are unchanged and CALL STATEMENTS presents the usual 11-choice menu.

The real changes start with the next section --CALL STATEMENTS II. The new menu looks like this:

1) CALL MOTION
2) CALL COLOR
3) GENERIC CALL I
4) GENERIC CALL II
5) GENERIC CALL III
6) EXIT TO MAIN MENU

The new additons are options 3, 4, and 5. Option 3 now allows for the entry of a customized CALL that requires no parameters. For example, a user-defined subroutine like CALL WAIT. Option 4 allows a "wildcard" entry with two numeric values added in the form CALL WAIT(10,20). Option 5 allows the entry of acustomized CALL with two numeric variables. An example is the well-known CALL ERR(ES,ET). The user need only follow the on-screen prompts to generate a syntactically correct line of code. The adition of these "wildcard" generic CALLS should allow the generation of virtually any CALL statement that is desired.

The next program segment is the CALL LOADS segment. The menu is as follows:

1) DISABLE FCTN=
2) ENABLE FCTN=
3) QUIT & RUN DSK1.LOAD
4) GENERIC CALL LOAD I
5) GENERIC CALL LOAD II
6) EXIT TO MAIN MENU

The first three options generate the appropriate CALL LOAD to accomplish the task shown --- just by choosing the option from the menu. A single key-press generates the line of code. Option 4 allows a user-defined CALL LOAD with a single numeric parameter and Option 5 does the same except that two numerica values can be added.

Has this little overview piqued your interest a bit? Well, the program is now up on ROS for downloading. I would ask that you respect the fairware attribution at the end of the documentation and mail a donation to the address listed. None of the proceeds of this program go to the author, by the way. That's the way I wanted it.

SHORT BYTES

Here is a short program that appeared in the Hunter Valley Newsletter that might be of interest.

```
100 !SAVE DSK1.HIGHLIGHT

110 !By Joe Nolan, Tacoma 99ers UG Newsletter Dec 1987

120 ! Modified by E Raguse UGOC 1987

130 CALL INIT::CALL LOAD(161128,2,224,38,0,2,0,8,17,2,1,63,36,2,2,0,34,4,
    32,32,36,2,224,131,192,3,128)

140 CALL LOAD(16164,244,244,244)::CALL LOAD(-31804,63)

150 PRINT "TURN IT OFF? PRESS SPACE, ELSE ANY"

160 CALL KEY(0,K,S):: IF S=0 THEN 160 ELSE IF K<>32 THEN END ELSE CALL
    LOAD(-31804,0)
```

FINAL NOTES

There are still a few TI programs making their appearance from time to time. Not enough to justify a fulltime column like the old "TI SHOPPER" that used to appear here but still a few of interest.

One recent entry is a program called PAGEPRO. Basically, it will allow a person to combine TI-Artist instances and text on the same page. The major difference between PAGEPRO and the other offerings is the number of pictures allowed (28) and the fact that the text is created by loading a font --- a practice similar to TI-Artist. In fact, the cover page on both the November and December issues were done with PAGEPRO. Just as its name implies, the program will let you create a single page at a time but no multipage documents. There are two versions contained on the disk. One is for the 4A and the other for the GENEVE and the package has gotten some mixed reviews with praise from some and disappointment from others. The biggest disappointment seems to come from GENEVE owners and those with 80-column capability who bemoan the fact that the GENEVE version doesn't take advantage of the 80-column capability of their systems. It seems that is simply scrolls like the 4A version 40 columns at a time. Still, if you are a graphics "nut" it is worth looking into.

BROWSE is another recent software offering but of only marginal value. It will let you look at text files and, as the name implies, browse through them. With the advent of the latest version of F'WEB and the DISKREVIEW program, BROWSE is simply a program without a purpose. Everything that you can do with BROWSE can be more easily done with the latest version of F'WEB.

Speaking of F'WEB, it is now in Vn 4.21 (as mentioned in PRES NOTES) and is a dynamite package. For several months, there has been an ongoing discussion in the HV99 Newsletter about the workings of the TI-WRITER editor and how it works internally. With all of the preliminary discussion, there was little doubt that something was "in the works" from Will and Tony McGovern. Version 4.21 has the editor buffer reworked with the result of a noticeable increase in speed. It used to be possible to lose a keystroke to wordwrap from time to time. I haven't been able to beat this one yet ---even with auto-repeat!! The reworked buffer also speeds up functions dealing with line and string handling. (F)ind(S)tring and (R)eplace(S)tring are now much faster and all of the functions with Lines are speeded up. Moving lines of text and deleting lines of text are now much more rapid. If you haven't stepped up to a copy of F'WEB 4.21, you are still in the "Dark Ages" of word processing.

One of the most celebrated and ballyhooed software packages was conspicuous by its absence. PRESS was supposed to have been introduced shortly after the 1988 Chicago TI Faire. Then, as the planned release date passed, there were suddenly no more announcements to be made. Rumors were spreading that it would be released at last at this year's Chicago TI Faire in November. No such luck!! The fact is that the program is still so full of bugs that any release date is sheer speculation. The cold, hard fact is that it might never be released in a usable form. At the present time, the TI-WRITER clones are the best programs going.

This has been a brief look at some of the software packages that were released in the past few months. There are certainly more than these few but the once steady stream of programs has been reduced to a trickle. We will try to cover some of the better software and hardware items as they are released throughout the rest of the year.