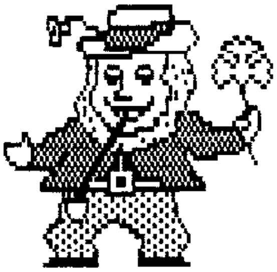
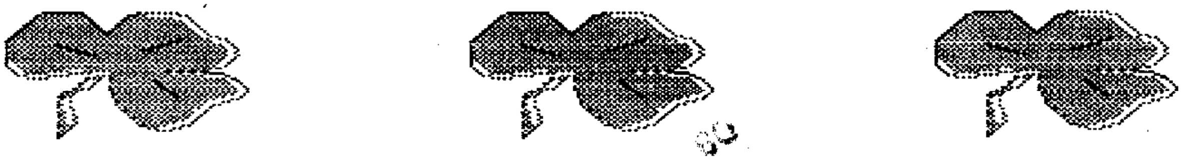
GUILFORD 99'ERS NEWSLETTER





SUPPORTING THE TEXAS INSTRUMENTS TI-99/4A COMPUTER

GUILFORD 99'ERS UG
3202 CANTERBURY DR
GREENSBORO NC
27408

GREENSBORO, NC 274
PM
20 FEB
1990

20¢ Universal Postal Congress

TO:

VOLUME 7 NUMBER 3 MARCH 1990

THE GUILFORD 99'ER NEWSLETTER

VOL.7 NO.3 MARCH 1990

Bob Carmany, Pres. (855-1538) Emmett Hughes, Vice Pres. (584-5108)
Mack Jones, Sec/Treas (288-4280) Bill Woodruff,Pgm/Library (228-1892)
BBS: (919)621-2623 --ROS

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:LF Jones, 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: March 6, 1990 Time: 7:30 PM. Place: Glenwood Recreation Center, 2010 S. Chapman Street.

Program for this meeting will be a demonstration of the RAVE keyboard for the TI. Basically, it is a larger keyboard that gives you single-press function keys and more. Come by and see Tony show you what it can do!

MINUTES

The February 6th meeting of the Guilford 99er Users Group was held at the Glenwood Recreation Center in Greensboro, N.C. There were 8 members and one visitor present.

We had three members to pay their yearly dues and unfortunately it seems we have lost two members. We have not heard from John Gollar or Andrew Small so they will be dropped from the mailing list.

The meeting was called to order by President Bob Carmany at 7:45 PM. The minutes were read and accepted as read.

OLD BUSINESS:

1. Bill Woodruff announced that the club's extra disk drive is all finished but it will not save data. He says that a friend has offered to provide a case for the drive when it is finally fixed. Bill also advised that he had received more information from the Chicago User Group on the TI/Writer.

NEW BUSINESS:

1. Bob advised that the BBS had a "crash" in January and all of the TI files were lost. They are being slowly uploaded but it will take some time to get it as it was. Bob asked that all who will to please upload files back on the board.
2. Bob announced that in two weeks (from the 6th) that he will be leaving for Australia. He asked that anyone wishing to donate for software to give the donations to him and he would pass it along when he gets there. He also says he intends to bring back some "goodies" from Australia.
3. Bob and George have invested in a MacInker to give to the Hunter Valley Group. Bob will take it as he goes there.
4. A motion was made and seconed to send the Miami Group an invitation to swap newsletters. We have not heard from 3 of our exchanging groups in 4 months so we will have to drop them from our mailing list.
5. Bob spoke on DISK REVIEW which is written in assembly language and will handle two files in memory at the same time. Bob says it will also read and print any type file.

Bill Woodruff demoed the TI/Writer Companion disk from the CUG and it is indeed a remarkable disk. Everyone enjoyed Bill's demo and he received a good round of applause from the group. Thanks Bill.

The meeting was adjourned at 9:27 PM.

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

RAMBLING BYTES

By 'Mac'

I guess each of you are enjoying the Spring-like weather we are having after the bitter cold we had at the start of January. This is indeed strange happenings. It hurts the "hacking" as you want to go out and walk in the sun shine instead of sitting looking out at it.

Our only out-of-State member, Ellen Kramer, has decided to join us again for another year. Thank you Ellen. Ellen lives in Ringwood, New Jersey and is very active in the TI. I would also like to welcome Larry and Gay Reid from Queensland, Australia to the membership. Larry's membership was donated by Bob Carmany. As you know, Bob will be on his way to see our "Mates" very soon now.

For those of you not attending the meeting, we received a flyer from William Chavanne in Ft. Meade Maryland who is vending TI-TAX templates. If any wish to do their 1989 taxes on the TI, this seems like a good bet. I find that they have made mine so simple that I have to pay through the nose to let a qualified CPA do them! If the Government gets them much more simpler, I doubt that the CPA will be able to handle them!

I know that you have more reasons for loving the USA than you have fingers and toes, but here is one more reason for me...a friend, Scott Copeland, who is the President of the East Anglia Region 99'ers in England was helping his children board their stopped school bus one morning and a speeding motorist came zooming around the bus and hit Scott. After a long period of recovery, Scott is now being sued by the motorist for damages to his car! Now can you believe that happening in the good old US of A? No way!! Not only do those folks drive on the wrong side of the road, they don't even give pedestrians the right-of-way at crossings. Makes you want to put on a powdered wig and say, "M'Lord...your justice stinks"!

There are quite a few TI's that are still gathering dust out there. Their owners probably bought them when they were going for $50. thinking they had a complete home computer in the console. After finding out they had to have the other components, they stuck them back or tried to get their money out of them. Anyhow, my son-in-law has a friend at his work that found that he and I are into the TI and asked him if he would like to have some "TI junk" he was going to throw in the garbage. Of course my SIL said he would love to have it and in a day or so the friend showed up with a version 2 gray console, a P-box with 32K, single drive and card, one of the early thermal printers that TI came out with (minus the power supply transformer, naturally) but no RS232. After reading about the printer in the manual that came with the printer, I found it plugs into the I/O port so I guess there is no need for a RS232. Wish it would tell how many volts it takes to run the printer, but there was no mention of this. Anyone out there know how many volts it takes to run the printer? It is an ALPHACOM 81.

I don't know how many of you are interested in Genealogy but I have recently become interested in it for myself. I know there are Jones' all over the Country, but I was wanting to find out my personal heritage. After talking to my mom who by the way is 83 and getting a little dense on things past, I wrote down most of what she could remember. We made a small trip to the country grave yard where most of her kin were interned and I found stones dating back to the 1790's but the information was limited. I borrowed a copy of the disk from the library from the Hunter Valley called THE GENEALOGICAL DATA BASE and thought it would be simple work. Was I ever Wroooooong! First, I realized I have waited too long in my life to start something like this. All of the important sources for information have passed on. If you have young children, try to get them interested in their family tree while they can gleen information from family members and their in-laws etc. The Hunter Valley disk is very helpful I found if you have all needed information. It is very thourgh even giving a menu of diseases you can list as the cause of death of deceased members. I just can't seem to find the time to go to the County Courthouse to look up all this information. Anyhow, if you think you might want to try this, start while you are young if possible. By young I mean under 50!! Since our groundhog here in N.C. never had a chance of seeing his shadow, I am still looking for some more bad weather before we can pull out the short sleeves and short pants for good, so until the April showers start the May flowers, enjoy the good TImes and see you at the meeting.

THE GUILFORD 99'ER NEWSLETTER

VOL.7 NO.3 MARCH 1990

Bob Carmany, Pres. (855-1538) Emmett Hughes, Vice Pres. (584-5108)
Mack Jones, Sec/Treas (288-4280) Bill Woodruff,Pgm/Library (228-1892)
BBS: (919)621-2623 --ROS

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:LF Jones, 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: March 6, 1990 Time: 7:30 PM. Place: Glenwood Recreation Center, 2010 S. Chapman Street.

Program for this meeting will be a demonstration of the RAVE keyboard for the TI. Basically, it is a larger keyboard that gives you single-press function keys and more. Come by and see Tony show you what it can do!

MINUTES

The February 6th meeting of the Guilford 99er Users Group was held at the Glenwood Recreation Center in Greensboro, N.C. There were 8 members and one visitor present.

We had three members to pay their yearly dues and unfortunately it seems we have lost two members. We have not heard from John Gollar or Andrew Small so they will be dropped from the mailing list.

The meeting was called to order by President Bob Carmany at 7:45 PM. The minutes were read and accepted as read.

OLD BUSINESS:

1. Bill Woodruff announced that the club's extra disk drive is all finished but it will not save data. He says that a friend has offered to provide a case for the drive when it is finally fixed. Bill also advised that he had received more information from the Chicago User Group on the TI/Writer.

NEW BUSINESS:

1. Bob advised that the BBS had a "crash" in January and all of the TI files were lost. They are being slowly uploaded but it will take some time to get it as it was. Bob asked that all who will to please upload files back on the board.
2. Bob announced that in two weeks (from the 6th) that he will be leaving for Australia. He asked that anyone wishing to donate for software to give the donations to him and he would pass it along when he gets there. He also says he intends to bring back some "goodies" from Australia.
3. Bob and George have invested in a MacInker to give to the Hunter Valley Group. Bob will take it as he goes there.
4. A motion was made and seconed to send the Miami Group an invitation to swap newsletters. We have not heard from 3 of our exchanging groups in 4 months so we will have to drop them from our mailing list.
5. Bob spoke on DISK REVIEW which is written in assembly language and will handle two files in memory at the same time. Bob says it will also read and print any type file.

Bill Woodruff demoed the TI/Writer Companion disk from the CUG and it is indeed a remarkable disk. Everyone enjoyed Bill's demo and he received a good round of applause from the group. Thanks Bill.

The meeting was adjourned at 9:27 PM.

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

RAMBLING BYTES

By 'Mac'

I guess each of you are enjoying the Spring-like weather we are having after the bitter cold we had at the start of January. This is indeed strange happenings. It hurts the "hacking" as you want to go out and walk in the sun shine instead of sitting looking out at it.

Our only out-of-State member, Ellen Kramer, has decided to join us again for another year. Thank you Ellen. Ellen lives in Ringwood, New Jersey and is very active in the TI. I would also like to welcome Larry and Gay Reid from Queensland, Australia to the membership. Larry's membership was donated by Bob Carmany. As you know, Bob will be on his way to see our "Mates" very soon now.

For those of you not attending the meeting, we received a flyer from William Chavanne in Ft. Meade Maryland who is vending TI-TAX templates. If any wish to do their 1989 taxes on the TI, this seems like a good bet. I find that they have made mine so simple that I have to pay through the nose to let a qualified CPA do them! If the Government gets them much more simpler, I doubt that the CPA will be able to handle them!

I know that you have more reasons for loving the USA than you have fingers and toes, but here is one more reason for me...a friend, Scott Copeland, who is the President of the East Anglia Region 99'ers in England was helping his children board their stopped school bus one morning and a speeding motorist came zooming around the bus and hit Scott. After a long period of recovery, Scott is now being sued by the motorist for damages to his car! Now can you believe that happening in the good old US of A? No way!! Not only do those folks drive on the wrong side of the road, they don't even give pedestrians the right-of-way at crossings. Makes you want to put on a powdered wig and say, "M'Lord...your justice stinks"!

There are quite a few TI's that are still gathering dust out there. Their owners probably bought them when they were going for $50. thinking they had a complete home computer in the console. After finding out they had to have the other components, they stuck them back or tried to get their money out of them. Anyhow, my son-in-law has a friend at his work that found that he and I are into the TI and asked him if he would like to have some "TI junk" he was going to throw in the garbage. Of course my SIL said he would love to have it and in a day or so the friend showed up with a version 2 gray console, a P-box with 32K, single drive and card, one of the early thermal printers that TI came out with (minus the power supply transformer, naturally) but no RS232. After reading about the printer in the manual that came with the printer, I found it plugs into the I/O port so I guess there is no need for a RS232. Wish it would tell how many volts it takes to run the printer, but there was no mention of this. Anyone out there know how many volts it takes to run the printer? It is an ALPHACOM 81.

I don't know how many of you are interested in Genealogy but I have recently become interested in it for myself. I know there are Jones' all over the Country, but I was wanting to find out my personal heritage. After talking to my mom who by the way is 83 and getting a little dense on things past, I wrote down most of what she could remember. We made a small trip to the country grave yard where most of her kin were interned and I found stones dating back to the 1790's but the information was limited. I borrowed a copy of the disk from the library from the Hunter Valley called THE GENEALOGICAL DATA BASE and thought it would be simple work. Was I ever Wroooooong! First, I realized I have waited too long in my life to start something like this. All of the important sources for information have passed on. If you have young children, try to get them interested in their family tree while they can gleen information from family members and their in-laws etc. The Hunter Valley disk is very helpful I found if you have ail needed information. It is very thourgh even giving a menu of diseases you can list as the cause of death of deceased members. I just can't seem to find the time to go to the County Courthouse to look up all this information. Anyhow, if you think you might want to try this, start while you are young if possible. By young I mean under 50!! Since our groundhog here in N.C. never had a chance of seeing his shadow, I am still looking for some more bad weather before we can pull out the short sleeves and short pants for good, so until the April showers start the May flowers, enjoy the good TImes and see you at the meeting.

CALL PEEK("PRESIDENT")

This is a column written "in absentia". I would like to thank Bill for the excellent program last month. I had seen most of the programs contained in the Chicago TI-Writer release but it was interesting to see them together in one offering. That is the way of some of this "fairware", a series of programs brought together that were written over several years.

The program this month will be a demo of the RAVE keyboard. It is a full-sized (ugh!) IBM-type keyboard that replaces the TI keyboard. The main advantage is that it replaces some of the multiple keypresses (ie. <FCTN>) and another key with a single key. Besides that, the keys are a bit larger and you can "hide" the console out of the way and use RAVE. There are a couple of models and if you like the larger IBM keyboard, you need to see this demo.

About 5 years ago, we got a plaintive letter from Adelaide, Australia from Mr. Larry Reid. Larry, it seemed, wanted a computer "pen pal" to write to. I figured it was worth the price of a stamp and we have been corresponding ever since. A fairware donation or two, and an introduction here and there and now I'm corresponding with just about the entire city of Newcastle. A "sometimes" column turned into an every month effort and plans were soon made to see the "wonders down under" firsthand. After a year of saving pennies, I finally got enough together to make the trip! I'll drink a cold one for ya, mate! Come to the meeting next month to see if I remembered to come back to the States!

XB TUTORIALS PART 7

By Tony McGovern

We now continue the Tutorials with detailed ways and means of scrunching program length. As I remarked before, it's a subject I'm not completely comfortable talking about because, while this series has been devoted to better XBasic programming, most things you can do to scrunch Basic programs make them less readable by ordinary mortals, given reasonable programming skill in the first place. The other reason for my reluctance is that this kind of discussion tends to degenerate into a collection of unrelated items, yet another set of "Tips", when I really want this series to be a gentle but systematic look at the workings of the machine and its language(s).

Anyway let's start at the small end of things and work up to the larger scale. Last time we looked at the space taken by simple variables. The most obvious thing is to keep variable names short. I don't recommend this until late in the piece because it is such a cheap and obvious way of gaining bytes that you might as well have the help of descriptive variable names until you are absolutely desperate for bytes. Absolute desperation has not occurred until you have had several rounds of byte saving already. The shortest variable name has only one letter character, but TI Basics also officially allow "@" (shift-2) and "_" (fctn-U) as variable names. It has to be a fairly long SUBprogram before you need more than 26 simple numeric variables but it can happen. On this console there are 3 other single characters which can be used as variable names. Experiment to find if they exist on your machine. The nagging problem is that they are not documented.

There is another way to use variable names to shorten a program. Remember from last time that a one digit numeric constant is treated as a string and takes 3 bytes, while a single letter variable takes only 1 byte. If a particular numeric value occurs frequently in a SUBprogram, 0 or 1 being common examples, then it may be worth the overhead, 14 bytes plus the defining statement, for a new variable of that value if you can then save 2 bytes on numerous occasions. A frequently used longer numeric constant, as might occur in CHAR or SPRITE manipulations, yields more bytes each time. It is a matter of doing careful book-keeping and byte counting in each SUBprogram. Once you start down this track be alert for further gains -- if you have defined S=7 and F=5 then it saves a byte to write S*F instead of 35. If you can reuse an already defined variable name then the investment is paid back faster, but this requires keeping very careful track of program flow. Go back to the example of a Key/Joystick routine in an earlier Tutorial and see if you can shorten it by reducing the number of variables used.

Replacement of numbers by variables has precedents in other languages. In TI-Forth the numbers 0,1,2,3 are not treated directly as numbers but are defined words in the language.

There is another little way that cunning entry of characters can shorten programs. This is in the entry of graphics characters with ASCII values above 127 in the upper color groups of XB by writing strings with DISPLAY AT instead of H VCHAR CALLs. Characters in this range can be entered in strings in program statements by use of the CRTL key, rather than by using the CHR$ function. It does tend to make the program incomprehensible as these echo as blanks to the screen. They will appear with their defined shapes if the line is called up for editing after RUNning the program. These codes are also used as XB tokens and can only be used within strings. I should add in passing that I am in total agreement with the TI designers' choice not to allow abbreviated (direct token) entry of Basic keywords. If you want that sort of thing you should be back on your Sinclair or Commodore, and you probably don't believe in relocatable object files either,

The use of arrays to represent small collections of numbers needs detailed working out. The gains from less variable table overhead and simplified parameter passing to SUBprograms have to be balanced against the extra bytes needed for each program reference. Let the program logic be your initial guide.

This idea of using fewer bytes to represent quantities leads on to the larger subject of data compaction. One byte can carry 256 different values, and one third to one half of those can be conveniently entered from the keyboard. It's sheer overkill to use an 8 byte floating point number to represent just a few values, or even just a logical (Boolean) variable which really needs only one bit. Some languages compact Boolean variables as bits in a word or words. The CRU single bit bus of the TMS-9900 provides an ideal mechanism for bit storage and testing, but as in so many other areas the 99/4a hardware does not do justice to its CPU. The later TMS-9995 in fact has a little on-board CRU memory for just this purpose.

Opportunities for data compaction are limited in XB both because of the structure of the language (it has only character strings, floating point numerics and arrays of these as data types) and the convoluted, slow way it is implemented via GROMs and VDP memory. Any scheme for coding or compacting needs computation to pack and unpack the data. At the machine code level the tradeoffs between memory use and speed are different from those in Basic, especially TI-99 Basics, because Basic is so much slower. In my experience the use of string variables to compact data in active parts of a program is almost always doomed to failure because of slow string handling by XB and pauses for garbage collection. Data compaction can be useful though in setting up initial graphics designs or for music data. There are only so many different notes, in pitch length and volume used in any given short musical piece, and since each note takes time to play and is handled by the machine on an interrupt driven basis, this time can be used to do the computations needed to unravel the data for the next note.

Let's have a look at the graphics screen example. Suppose that in setting up a game screen, either one of two characters, maybe the same pattern in two different color groups, has to be written to 20 locations in various parts of the screen. The simplest way is a whole succession of CALL HCHARs - assuming the display is not suited to generation with DISPLAY ATs - and that's the way you will find it done in many programs (just like long lists of CALL SOUNDs). What is totally unforgivable is to find incompetent magazine or commercial programs with inefficient coding that force inconveniences like CALL FILES(1) or (2) on the user when it could have been avoided.

```
1000 CALL HCHAR(23,12,105)
1010 CALL HCHAR etc etc
```

This takes over 600 bytes. How can it be shortened ? One way, a bit of a dead end in this example, is to use multi-statement lines. This would be shorter by 30 bytes or so, and marginally faster. The real improvement is to eliminate the repetition of CALL HCHAR - remember CALL is cheap but HCHAR is expensive - by using a loop and DATA statements.

```
1000 FOR I=1 TO 20 :: READ A,B,C :: CALL HCHAR(A,B,C):: NEXT I
1010 DATA 23,12,105, etc etc
```

Now all but one of those HCHARs have gone. The price paid is loop and DATA execution overhead and the increased possibilities for clerical errors since the DATA items have been divorced from their proper context. At this stage you may be feeling very pleased with yourself, but then you find that to add another feature to your program you need more space. Now is the time to reflect seriously on data compression. A column index for HCHAR can only have the values 1 to 32 and rows 1 to 24. One of these values can be expressed by 1 byte with possibilities to burn. Say you use 1 byte for each row or column value then. Expressing the bytes efficiently as DATA is the next problem - there are a few bytes of overhead for each item in a DATA list, and DATA lists of a lot of short items are notorious for causing a "line too long" error. So let's pack them in a single string and use SEG$ to unpack them, with ASC to turn a ASCII character back to a value for HCHAR. A minor problem is that characters 1 to 32 can't be entered directly in XB, so just use characters starting with "A" and subtract 64. The opposite problem may occur with the string for the character values if upper graphics sets are being used. Then just use lower values and add a correction. So now the code might look like

```
1000 READ A$,B$,C$ :: FOR I=1 TO 20 :: CALL HCHAR(ASC(SEG$,I,1))-64,ASC(SEG$( B$,I,1))-64,ASC(SEG$(C$,I,1)+32):: NEXT I
1010 DATA "W...  ","L...  ","I...  "
```

You could further pack the data into a single string and modify the SEG$ statements accordingly, but it might not be worth it. Remember now that the problem posed involved writing only two different characters and work out how you could compact things still further for this limited case. This example is based on one of methods that was used to squeeze TXB into console memory. An extreme example of data compression comes when the data is regular enough that it can be generated by a formula or procedure. This is something that has to be worked out in each case.

The use of loops as in the examples above applies in other situations, particularly in CHAR definitions. XB allows the use of multiple arguments in CHAR, COLOR, SPRITE and suchlike SUBprograms. This is better and faster than using individual SUBprogram CALLs for each item in the list. The real dilemma comes when you try to use a loop to compact the program further. Critical parts of the program may be slowed down unacceptably so that you may find yourself using compact slow code in some parts of a program and longer but faster forms elsewhere. Just in passing I should remind you to null out on exit from a SUBprogram, any string variables not required to keep their value till the next CALL. This particularly applies to string variables used for READ, INPUT, PRINT etc operations involving long strings. Remember that it is the length of a program while RUNning that really counts.

IX More Bugs in XB

Now funnelweb spiders aren't exactly the nicest critters to be found around Funnelweb Farm, but they do have the virtue that if you don't bother them they leave you alone too. Unfortunately the bugs that infest Extended Basic aren't nearly so accomodating in keeping out of the road in the first place, and insist on making their presence felt.

I have looked at a few in previous XB tutorials where they came up naturally in the subject matter, mostly in ACCEPT AT. This time we have two beautiful specimens. The first of these was exposed in the Spring 85 issue of TI*MES from the UK by John Bingham. To see it at work, enter the following little program

```
100 I=1 :: IF I=1 THEN J=1 :: GOSUB 200 ELSE K=1 :: J=2
110 PRINT K;J :: STOP
200 RETURN
```

Before you run this, predict what it is going to print out ! Then run it and see what you get. Next reverse the order of the statements between THEN and ELSE and run it again. Now it should work the way you expected, K=0 and J=1. If your XB gets it right the first time, it's different from the one I have. The presence of the GOSUB just before ELSE seems to have upset XB's mechanism for keeping track of ELSE, and the program has gone ahead and ignored the ELSE and the statement after it and executed the following statement which it should have ignored entirely while proceeding to the next line. Try substituting a dummy SUBprogram CALL for the empty GOSUB. XB then works just as expected. Yet another reason for using SUBprograms instead of GOSUBs.

The XB manual lays down a few prohibitions on what can go into IF seem, despite the warnings, that FOR..NEXT loops can follow the final ELSE without problems, but this usage is not to be recommended as it may not hold good for all XB modules.

I must admit that reading this news had me a little worried, as I have written long and thoroughly debugged XB programs with some tricky IF..THEN..ELSE footwork, and had never picked up this problem. How come ? The first saving grace is that the Tutorial advice I gave is for real, and I use very few GOSUBs and very many SUBprograms unless I am absolutely desperate for more bytes. This was frequently the case in the writing of TXB, and the central SUBprogram, one of 12 in the program, itself contains 12 subroutines written in to save bytes. Careful study of the code for TXB showed that none of the GOSUBs was written in a way that would let this evil bug loose. When you look back at something like this, you wonder whether you had scrapped particular pieces of code that never quite worked properly, for entirely wrong reasons.

The second bug mentioned has not yet been fully explored. It showed up in a CALL LINK from XB to an assembler routine which also passed in a string variable to be examined by the routine. It happened in the COLIST program, the all-singing all-dancing version of the SIMPLIST program which appeared in an earlier XB Tutorial, The symptoms were that the machine crashed utterly when it was printing out a line of its own listing, after it had already printed several hundred lines, many quite similar to the one that caused to the trouble. Not just an error caught and reported by XB, but a full blown paralytic seizure. It turned out after some TRACE work to be in the very line that was being processed for printing, and to be asociated with the LINK name being at a particular position in the line of text being passed in with STRREF. The problem seems to be CALL LINK extending its link name search over places it shouldn't orter, but more research is needed. Further reports in Entomology Corner in a future issue.

One disappointing discovery came up in the bug hunt. I disassembled the XB machine code utilities loaded by CALL INIT to see if the problem was in STRREF. The good news is that that code looks OK, but the bad news is that STRREF reads strings out of VDP in the same slow way that the console does, 1 byte at a time, resetting the VDP addresses each time. This is the tortuously slow way GPL does it because the console has hardly any CPU RAM, but it scarcely seems necessary in STRREF which can only be used when there is expansion RAM present.

Another bug in CALL LINK has been reported in the US of A in some older XB modules, I suspect prior to the models of V110 sold in Australia. This comes when the link name is supplied as a string variable, as in CALL LINK("A$"). If a garbage collection is performed in VDP RAM by the XBasic interpreter in between assigning A$ and using it in CALL LINK, this routine would lose track of where A$. I have not encountered this bug myself, but I will try to stir it up in the XB modules I have. This reported bug brings to mind the strange state of affairs in XB where it is possible to DELETE a file by string variable reference but not to RUN a program file by similar reference, leading to a small cottage industry of ways around this deficiency.

X A Few Last Little Items

As you are now no doubt aware not all Basic programs will run correctly in XB, and some fail altogether, usually because XB supports fewer color groups than console Basic. Suppose you want a program that can run under either Basic or XB, and determine from either without crashing which interpreter is in control. If it can run under either, then you should be able to edit it in either! Some suggestions have been made to use PI which is a reserved word in XB giving the usual math value. This runs afoul of edit processing though. A better way is to look at the first value returned by RND. The initial seeds have quite distinct values. Just don't call RANDOMIZE first.

If you are writing such a program it will also be necessary to take care in using colons in PRINT statements to cause multiple line scrolling. XB will enter two colons in a row in a statement as a statement separator token. It is necessary to use semi-colons in between to stop then being run together when a line is being called up for editing as in

```
200 PRINT "A":;:;:"B"
```

XI In Retrospect

These Tutorials were originally written about a year after TI orphaned the 99/4a. They have even been translated into Swedish since then. Now it is over 5 going on 6 years since Black Friday in late 83 and the old 99/4a is still going strong, and the industrial strength of the expanded system has allowed all sorts of new developments, RAMdisks by the megabyte if you want, a hard disk controller, and best of all now 80 column displays using the Yamaha development of the original TI video processor. The machine's potential is by no means exhausted and it remains a fascinating one to work on, and even the original console/XB remains a viable and now very low cost introduction to computers for new beginners.

(Ed Note: This completes the XB tutorial series. If you missed any of the articles, the complete series is on ROS.)

TONY'S CORNER

Confessions of a Small-TIme User. by Tony Kleen. Guilford (NC) Users Group.

Okay, you want to create a TI based product, and use TI-BASE as your file management system. What will you do, what willlll you do?? Well, first, I've got several questions for you. What do you want the system to do for you? What are your entities, your 'critical' collections of data? How are your entities related to each other, ie., a relational database system? What are the attributes of your entities, the fields in your records? Do you know how to normalize a relational database? DO you know how to optimize a database system?

Lots of questions in the above paragraph. I must admit that I didn't answer but one or two before implementing my first system on TIB (TI-BASE). Consequently, I have a very segmented process. BUT, I did get the darn thing up and running quite quickly, which was the critical factor! Now that I have time to do it right, according to my own theories, that is; let's get cracking on a new system for the TIB..

To start off, I'm going to answer my first question; what do I want my system to do? Well, what I need is a system that will make my life easier in the area of finance. I have volunteered to be the treasurer for a local recreational club. We have monthly dues, initiation fees, special activities; the old accounts receivable/payable stuff. I would like my system to print the periodic (monthly for me) statements for a client, or the entire group of clients. I would like it to post the monthly dues or other receivables; for an individual or group of individuals. Of course, it needs to be able to post the client payments, then prepare the deposit slip and post to the bank account. Posting the periodic balances for an individual or group would be necessary. Reconciliation of the bank account, with a printed record would certainly help out. Every quarter, I would want the client's transactions summarized into a single balance transaction; in other words, I wouldn't want more than 1/2 year of detailed transaction history to print on the statement. Oh yeah, since I have addresses, label printing by individual or groups would be handy!!

Quite a nice wish list, wouldn't you say?? A system to handle all of the above isn't as far fetched as you might think.. Wouldn't it be nice if this system had its own HELP and ERROR analysis? How about it being user friendly, with the pull down menu approach; or commands for the expert operator. Gosh, have we got our work cut out for us..

The next topic we need to address are the critical entities (databases, files, collection of records; whatever you want to call them). Then we will talk about their relationships to each other. Finally, we'll begin to populate (list the attributes for each row, the field for each record; again, whatever) each entity. In SQL (a structured query language), what we call a file can be likened to table; what we call a record can be likened to a row; and each field would be called an attribute. Since I'm currently working in SQL at my business, I may from time to time use table/row/attribute for file/record/field. Please bear with me. I might also mention that entities most likely will become tables/files, depending upon the outcome of normalization and optimization..

Back to our entities. It appears to me that the CLIENT is critical to our system. The organization that the clients belong to seems necessary. The clients detailed transactions are definately critical to our system's success. We also need client balances, and organization balances. In the sentences that precede this one, I've listed five entities necessary for our system; the organization entity, which I'll refer to as OWNER; the CLIENT; the client's ACTIVITY, his detailed receivables/payables; the client's periodic history, which I'll refer to as CLNTHIST, which is the client's periodic balances; and the orgaization's periodic balances; which we'll call MSTRHIST. SO, as an initial delve into our design, we have defined five entites, MASTER, CLIENT, ACTIVITY, CLNTHIST, and MSTRHIST.

Let me define the purposes of each of these entites. The MASTER is the entity which describes the organization for which we are doing the accounting. In my case, this will be the Brookcliffe Club. I plan to use this same system for many organizations, so Brookcliffe won't be the only organization I need to keep track of. The CLIENT will be the family who has membership in the club. Each month, the member is billed for his monthly dues, and normally, each month, he pays those dues. These are examples of what will be included in the client's ACTIVITY entity. The client's periodic balances (monthly) comprise the CLNTHIST entity. Similarly, the organization's balances comprise the MSTRHIST entity.

Now that we know the entity names, and a little bit about them; I now need to describe how each entity is related to the others. Each MASTER entity will have 1 or more CLIENT entities related to it. Also, the MASTER will have many MSTRHIST entities, one for each month of the year in my case. Each CLIENT entities will have zero, one, or more ACTIVITY entities (transactions) related to it. Also, the CLIENT will have one CLNTHIST entity per month.

It may not seem like we've done much yet, but we've conceptually designed a relational database that will allow us to fulfill that wish list we expressed in the first paragraph. Our next endeavor will be to populate the entities. This will give us a list of the columns/fields that are to be contained in each row/record of our table/file.

I might mention this from the beginning. I am not planning this column to become a tutorial on TIB. I would just like to share with you my findings. See ya' at the next meeting.

(Ed Note: The first of a series?)