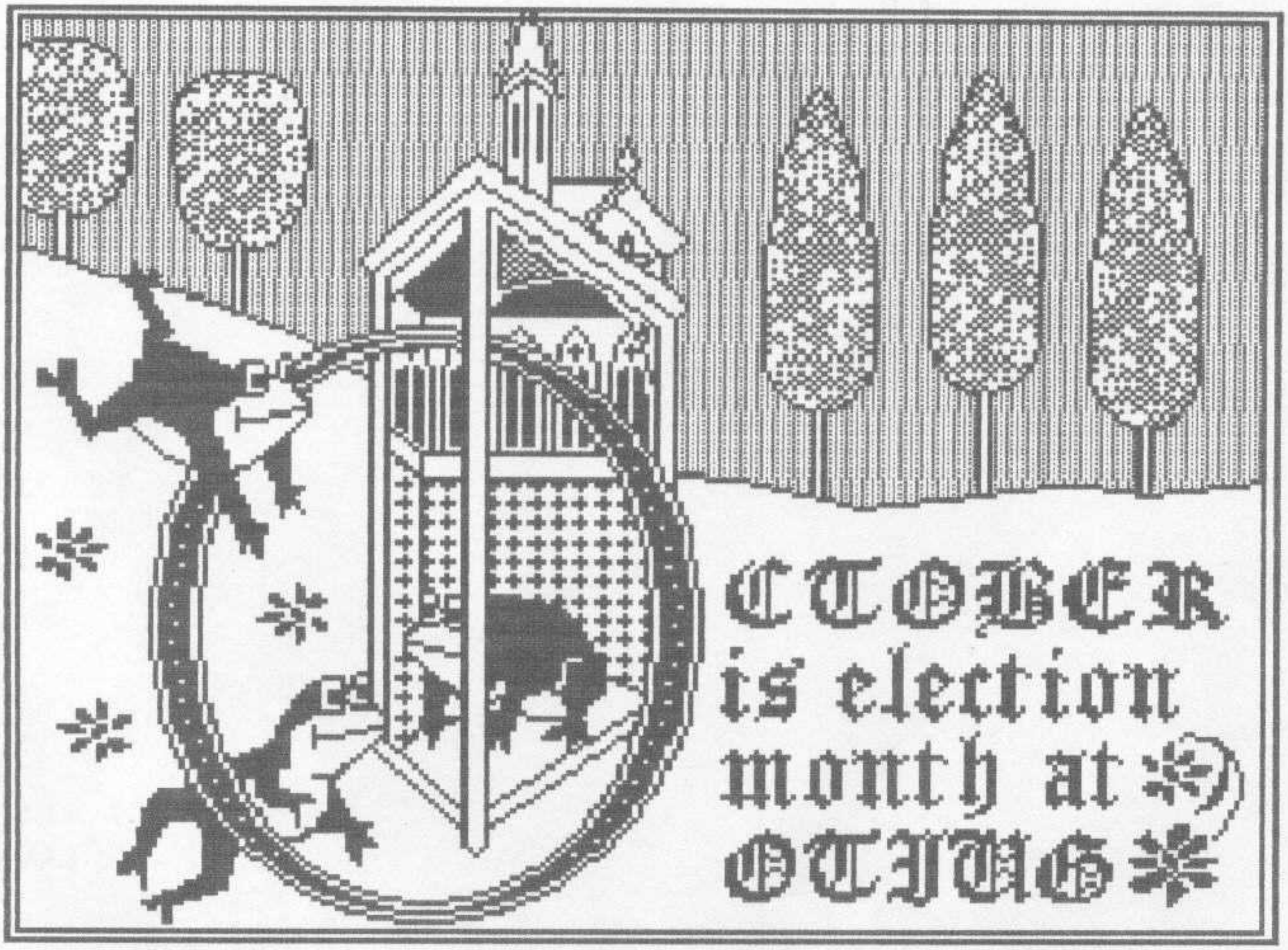
The Ottawa TI-99/4A
User's Group

NEWSLETTER

Volume 13, Number 8 ... October 1994



OTIUG: 222 Guigues Ave, Apt 603
Ottawa, Ontario CANADA K1N 5J2

COMING EVENTS

October meeting: 6 Oct 1994 at Lucie Dorais' apartment - 222 Guigues Ave Apt 603, Ottawa, ON at 7:30pm

November meeting: 1 Nov 1994 (tentative date)

Newsletter deadling: 20 October 1994. Submissions may be sent to the TIUG at the address on the cover care of the editor, or e-mailed directly to me at mbrent@proton.com

TWO CENTS WORTH by Lucie Dorais

I am just back from a lunch with our new editor... and what I have seen (part of the new cover page design, with a video digitized picture of a TI!) is very promising. Mike, who will use his Amiga to do the newsletter, has got many ideas of articles, starting this month if he has the time to write them (he is also moving at the end of September); he already pestered Jeff Brown, so there is probably an Assembly article in here as well. I feel so relieved that I have decided to write an XB column again this month, after saying no yesterday, but really it is about Visual Basic for Windows and how it relates, or not, to XB.

So, there might be life in the Ottawa UG Newsletter after December 31st, at least for a few more months. Philip Harris stays on as Treasurer of course, and it is a good time to thank him for the good job he did as editor.

OK, more TI news: last month, I mentioned that Cecure Electronics had received some hardware from Lou Philips, but that was only part of the story: the company, expertly managed by Don Walden, with the help of Tim Tesch (a GEnie regular) also got the contract to repair ALL TI hardware, making Cecure the ONLY Authorized TI Service Centre; yes, Don got all the spare parts, service manuals and related docs, so he can repair about anything. The only difference is that now you will need to phone in advance and get a Repair Authorization number before sending him the part to be repaired.

Contact: CECURE ELECTRONICS, INC.
P.O. Box 132
Muskego WI 53150
1-800-959-9640 or 414-679-4343

You might notice that Don managed to get the Geneve number into his 800 number... clever thing, but he does not discriminate on good ol' TI.

And we heard from Ed Swartz, the author of the PC TI Emulator. In a public message on the TI-Echo (reprinted in THE HUGers, the Hoosier UG newsletter), he discloses "the truth about the ROM scandal". I summarize: Ed got a letter from TI asking to remove the ROMs from his module dumps until he get a license from them, and to remove all copies of the Emulator from public BBSs, Internet, etc. He quickly complied to that letter, and as quickly wrote to TI asking about getting a license. But as of August 10, he had not received a reply from TI, and was thinking of ways to release the Emulator with instructions to users as to how they can transfer their ROMs over and how to patch them for Emulator use. So, as he says, there is a Future for the program. As a happy user, I must say I am relieved, I was afraid that Ed had been so discouraged that he had abandoned the TI world. You can write to Ed at this Internet address: swartze@ralph.txswu.edu

A note on the cover picture: It is adapted (greatly simplified!) from an initial letter in the "Missal of Saint-Denis", a manuscript written near Paris around 1350 (fol. 261); the original pic shows a church, a deer inside, dogs jumping around the big O, with figures, the subject being "Saint Denis Aids Dagobert". I got the idea last Winter, but had to wait for a month starting with "O" cause it was easier to work with the same initial as in the model than to redesign the picture. Now how did a TI-Artist pic got on the Amiga? via my PC: the program PaintShop Pro 2.0 converted it from RLE (MAX-RLE on the TI converted it from TI-Artist) directly to IFF, the Amiga format; OK, not directly: I first did an interim PC BMP version to add the sky shading...

Words From the Editor

Well, as Lucie mentioned, I'm the new editor. Name's Mike Brent. Some (a lot, perhaps), know me as Mike Ward. Long story, short result. 'Nuff said. Anyway, Lucie mentioned that I'm doing this on an Amiga. As some of you probably know, the Amiga's an orphan now, too. I'm starting to feel responsible for the death of computer lines... every time I buy one, they go out of production.

Anyway, to bridge the gap, I'm working on a TI emulator for the Amiga. It's coming slowly, as I learn all about Amiga architecture. It's a good deal more complex than the direct hardware programming which is permissible on the TI. But, with Jeff Brown's assistance, it's nearly complete. He just has to get that last bug report to me so I can fix a few op-codes, and add video support. Then it can be tested.

Of course, after that, I still have to add all the graphics modes, sprites, sound, and DSR routines. But I think it'll be very stable. I have, of course, taken note of TI's desire to protect their ROMs, and will have to release it without them. However, I'll include psuedo-ROMs which will contain compatible routines at the correct addresses, and allow you to load and run MOST program image files, as well as a TI program to copy the ROMs and GROMs to a disk file for use with it.

Enough of that, on to the newsletter. It's VERY rushed this month, between a late changeover and my running around to find a new place to live. But, I hope you enjoy it. Lucie and Jeff both have their regular articles (although both went in a slightly different direction than usual). In addition, I've put together a little article with comments from the TI-Echo, plus I'm reprinting Beery Miller's predictions for the coming year in the TI world, with my own comments.

My apologies to both Beery Miller and Tim Tesch if they object to my reprinting of their messages.

At any rate, I'll be editor until March, at which point I will likely be leaving Ontario back to the sunny Okanagan Valley in BC. But I hope to provide a decent newsletter to everyone at least until that time, and then, well, we'll see what happens.

Enjoy!

FAST EXTENDED BASIC

by Lucie Dorais

No new XB program this month, absolutely no time to write one. I was too busy playing with another Basic so, as I did last Spring with QuickBasic, I will just ramble a bit about... Visual Basic for Windows.

If you have played a bit with Windows on a PC, or seen pictures of Windows programs, you know that it looks very different from a TI XB program: you got buttons to click, nice pictures, lots of nice colors. It all seems like magic, until you start writing programs for Windows. There are many ways to do that and, given Bill Gates' love for Basic (he apparently wrote our own TI BASIC, one of his first contracts), there is of course a Windows Basic: MicroSoft Visual Basic for Windows (there is also a DOS version; I have seen programs written with it, but never saw the program itself, so will not comment about it).

Why is it called Visual? Well, because the visual aspect of the program is very, very easy: you get a collection of tools and a page, called a "form" (could be as big as a screen, or as small as you need). To design the program, the first thing you do (after you got some idea of what the program should do of course!) is to click on a tool to pick it, click again on the form to place and size it. These tools, called "Controls", can be Command buttons, Scroll Bars, Labels for text to read, Text Boxes for text that can be modified (and this comes with basic word processing tools like deleting, cut and paste, all automatically done by Windows, no need to program anything!), Shapes and Lines to add some decoration, Frames to hold check boxes (any box can be checked in one frame) or option buttons (where only one option per frame can be checked at a time), etc. If you decide that the program needs a visual modification, all you need to do is move or resize the control on your form while in design mode, using the mouse. This part is really, really fun, and saves a lot of programming.

Now, how does a VB program work? In plain Basic, especially the kind with line numbers, the programmer decides what the program does, and when; usually, the user is left with a menu choice, a joystick to move, etc, not much. But when the user can click on ten buttons or scroll bars at any time, which in Visual Basic is called an "Event", the programmer's task is a bit more complicated, since he/she must anticipate any thing that the user might do.

Below are some examples taken from a program I am working on, a kind of Intertactive Reference book for VB. The Visual Basic programming environment is very similar to that for MicroSoft QuickBasic: remarks are preceded by a "'", there are no line numbers, and all the coding is done in subroutines. Even the starting code, to initialize variables, fill arrays, etc, is itself a subroutine, automatically executed by the program upon loading:

```
Sub Form_Load
    nl = chr$(13) + chr$(10)
    subshown = 1
End Sub
```

Programming in Visual Basic is called Event programming: if you click on a button, it triggers a "Click Event", and you can write a subroutine to respond to it. For example, the subroutine optSize_Click below responds to a mouse click on an option button, a choice to make pictures Bigger, Normal, or Smaller; curhit/wid and defhit/wid are size values (height and width), expressed in Twips (this is the measuring unit used by Windows to adapt itself to all video formats... it is very small, about 1/12 of a pixel in VGA resolution). The option buttons were put in a Frame (a rectangle) and named as an array to facilitate coding, Index being the array position automatically passed by the "event" as parameter, same way we pass parameters to user-defined subs in XB. The "Select Case" statement was used because it is easier to read than a series of IF... THEN... ELSE:

```
Sub opt
Size_Click (Index As Integer)
```

```
   Select Case Index
   Case 0              ' make pic smaller
      curhit = defhit - 240: curwid = defwid - 240
   Case 1              ' make pic normal size
      curhit = defhit: curwid = defwid
   Case 2              ' make pic bigger
      curhit = defhit + 240: curwid = defwid + 240
   End Select     ' now do the three steps in succession
   ResizePics 0
End Sub
```

ResizePics is a CALL to another subroutine, one that does not respond directly to an Event (note that it does not have a "_event" suffix), but that is put in a distinct subroutine because it is called by more than one Event. The parameter passed, "0", is explained in the ResizePics sub itself. Which brings me to why I decided, when I woke up this morning, to write this article: this routine has caused me lots of problems because it is a good example where "Event programming", versus sequential one, can cause headaches to the programmer.

The sub started life as plain ResizePics, with no parameter: in that sub was all the code needed: first empty the "containers" of the current demo picture they are holding (these "containers" are Picture Controls that can Load a bitmap picture), then resize the containers, and finally reload the demo picture.

[This part of my program shows how the loaded pictures resizes themselves or not into the container, depending if the "AutoSize" Property was set to True or False. Each Control has a list of Properties, like color, size, position on the form, etc, that can be adjusted at design time or changed at run time. For example the text in a label or on a Command Button, the equivalent of "DISPLAY AT" in XB: you place a label (or button) on the form, then change its content, or Caption Property, when and if needed.]

The problems started when I decided to add a "tutorial" aspect: by clicking on a button labeled "Show Me", I wanted the program to display part of the code, then ask the user to click the same button, now re-labelled "Do It", to trigger the part of the ResizePics sub that dealt with that part of the code (empty pic, resize container, or reload pic). Since, in VB, the code responding to the Button_Click event had to be totally separated from the ReSizePics code, with no way to relate them (I could have copied the code, which is not efficient programming; I could not have put the ReSizePics code only in the ShowMe_Click button sub, because ReSizePics also has to be done when an option button was clicked).

The solution was to add a parameter to the ReSizePics sub, and put a counter in the "ShowMe_Click" routine, since at least there was a definite number of times (four) that the user should click that button: once to start the tutorial, then once for each step. As a precaution, I disable the Size option buttons mentioned earlier when the "ShowMe" button is clicked.

When all three resizing steps need to be done at once, the parameter is "0" (see Sub optSize_Click above); when only one step needs to be done, as when ReSizePics is called from the "DoIt_Click" event, the parameter is equal to the current value of the counter, 1 to 4. Oh yes: the counter variable as to be declared as "Static" at the beginning of the sub, so that it keeps its value when the program exits the sub (unlike user-def subs in XB, which keep the variable values, in VB all variables are reset to zero when you Dim then upon entering a sub, unless you declare them as Static as here: if you need to reset a static var, you do it yourself, as here when counter has run its course, i.e. is equal to 4). Here is part of the code: the program comes in here each time the user clicks on the button, so ReSizePics is called four times:

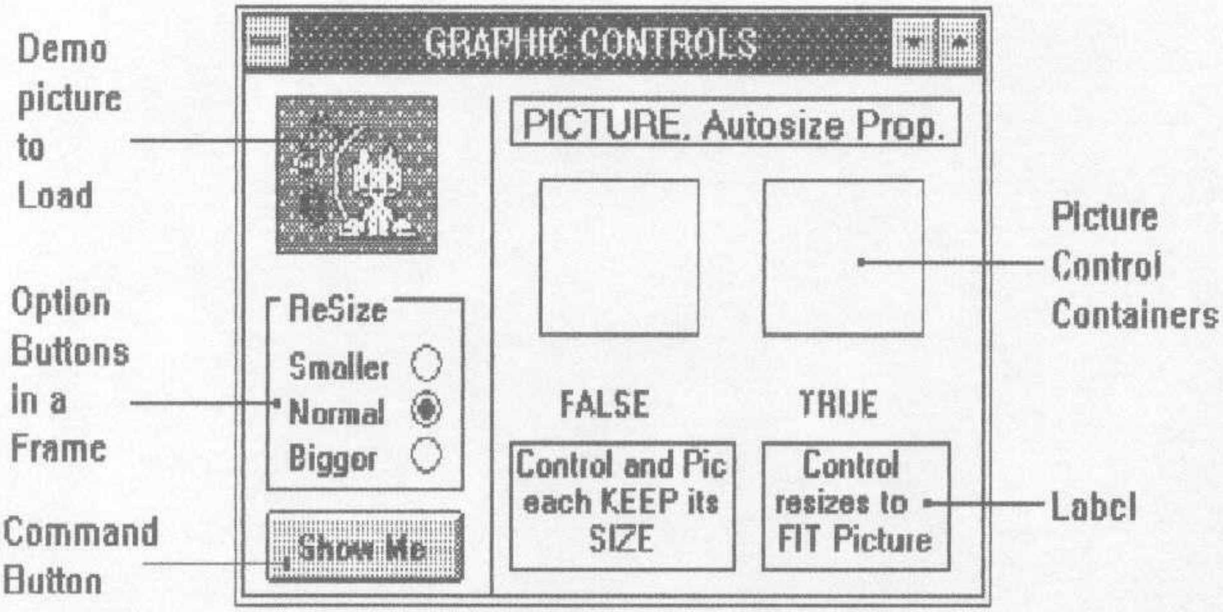
```
Sub btnShowMe_Click ()
    Static ctr As Integer
        If ctr = 4 Then ctr = 0            ' reset counter
    If ctr = 0 Then                    ' prepare tutorial
       Frame.Enabled = False           ' disable option buttons*
       btnShowMe.Caption = "Do It..."  ' change button caption
    End If
        ResizePics ctr      ' call to subroutine
     ' if ctr=0, do all 3 steps: here, we need to first reset the
     ' demo pics to normal state before starting, therefore we
     ' need to ReSize 0, then ReSize 1, 2 and 3
       Select Case ctr
    Case 0        ' display the code for first step
     Case 1        ' ditto for step 2
     Case 2        ' ditto for step 3
     Case 3
       Frame.Enabled = True           ' re-enable option buttons*
        btnShowMe.Caption = "Show Me"  ' reset button caption
    End Select
    ctr = ctr + 1                  ' ready for next step End Sub
```

* Instead of using a FOR...NEXT to disable the three option buttons, VB allows you to do it in one sweep by disabling the Frame Control (really a rectangle) that "contains" them.

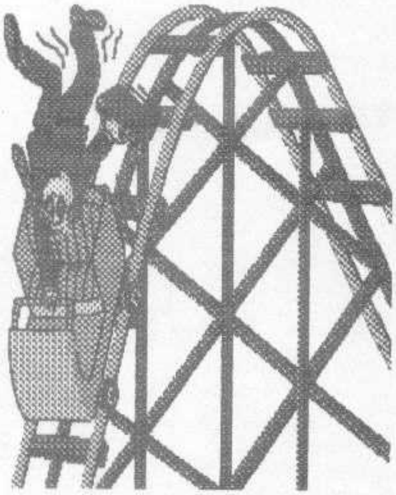
Now the ReSizePics code; "nostop:" is a label used with GOTO (like a line number in XB):

```
Sub ResizePics (ByVal dostep)
    ' passing the parameter By Value means the sub will NOT send
    ' back the new value, just in case...
    ' resize according to step
    ' do all 3 steps in succession if from optSize_Click
    ' do one step at a time if from btnShowMe-Click
    Dim ctr As Integer, all3 As Integer
    If dostep = 0 Then all3 = True: dostep = 1
   ' all3: a flag
nonstop:
    Select Case dostep
    Case 1                    ' code to erase the pics in containers
       If all3 Then dostep = 2: GoTo nonstop
    Case 2                    ' code to change size of containers
       If all3 Then dostep = 3: GoTo nonstop
    Case 3          ' code to reload pic into containers
    End Select
 End Sub
```

I hope you enjoyed this introduction to Visual Basic, and that I did not confuse you: I include a picture to illustrate it:



This is a Form



The Trials of an Assembly Language Programmer

by Jeff Brown

As many of you may know by now... I have been working on a project to take full advantage of my spiffy 14.4kbps external modem on the TI. I have left a number of messages on the TI FidoNet Echo to that effect. In this article, I would like to let you take a look into the trials and tribulations of doing hardware-direct programming on a TI. (as an aside from the Assembly article... I'm sorry I wasn't able to do the last one up... I was busy with this project.)

I had originally hooked up my high-speed modem to the TI using Telco. Unfortunately, it turns out Telco is too slow to access the modem at that rate without skipping characters. As well, since I don't have a fancy 80 column video chip, I was limited to 40 columns. Telco also doesn't do full ANSI, which means, NO visual editors on bulletin boards, nor did it do colour. I set about to create a better modem program, which would fix these shortcomings.

Ok... I started this project some time in the winter I think. I wasn't going towards the same goal then. At the time, I was satisfied with Telco, and only wanted another protocol... that attempt, failed miserably! I was getting in above my head! I immediately killed the idea... and deleted the files. [Ack! never delete your own source code! -Ed]

Hmm.. I can't remember when I started this particular project... I figure about mid-August... I found an article with a little example of how to directly access the modem port. I was overjoyed! I quickly put together a small terminal program (well... it got rather big!) with a few techniques I'd learnt. This became the basis of my program... here, let me show you.

1) I took an 80 column display routine (in bitmap mode) which I'd designed. It displayed a black and white image in which each character was 3 pixels wide, by 8 pixels high, with no spaces in between. It's squished, but readable.

2) I then designed a modem read routine, and a modem write routine. To avoid missing characters, the read routine was called every 6-10 instructions and stored incoming characters into a 24k buffer in high memory. Outgoing characters weren't buffered.

3) I added a direct keyboard access routine which is MUCH faster than TI's SCAN routine!!!

4) I put coding to treat line feeds, carriage returns, backspaces...

5) I put them together with initialization routines for the bitmap mode and a clearscreen (no scroll yet! or cursor...)... and tried it.

There were a few bugs at first, but soon after I managed a 2400 bps connection.

Ok... that done... I increased the modem speed (in the init routine) to 9600 baud. The modem returned a computer rate of 9600 bps and worked nicely! Then I tried 14400 baud... which worked, and finally 19200 bps, modem to computer!

That was a bare bones terminal... next to come was routine to clear the screen and a cursor.... until I got bored and wrote a proper scroll routine. All fine and well, but boring... I wanted MORE! I got documents on how to do ANSI cursor and colour control. This got buggy... I had to recode a lot of stuff to handle colour, plus 4 control keys to modify the screen colour (in case a bad colour combo arised). My first attempt at parsing ANSI codes failed for the most part. It was slow and clunky. It had relied on multiple compare instructions (IFs). I revamped the ENTIRE ANSI parser, using a table-oriented methodology. This seemed to work well. (I'm still using it now)

Ok... so I had ANSI (although slightly buggy still), I had 80 columns, I had 19.2kbps connections, I had a lot... but not proper interface.

Anyhow, I won't stick a large flowchart here... since it is growing... and I would be liable to take a page or more! Needless to say it is very well organized, and liable to be my best program yet! I has shown no sign of clutter due to growth (which are caused by inefficient coding...) which is a trademark of most programs that start small and G-R-O-W, since it has been constantly under improvement, debugging, and efficiency checks.

To give you an idea of complexity, the modem READ routine calls the keyboard scanning routine... does the control key interpretation (for things like arrow keys, clear-screen...) sends the data to the modem... all while it buffers characters to be used by the display subprogram (or program... it's 80% of the program with all its ramifications!)... it takes into account holding down keys and VDP priority of calling routines. As to size... the parser is indexed and huge! (about 600 lines). The low memory holds the main subroutines and all displaying. High memory has buffers, the ANSI parser, the 80 column character set. The rest is buffering space for the READ routine.

At time I am writing this article it stands at 2100+ lines of 100% assembly. It does colour ANSI and near full ANSI. It allows using ANSI arrow keys instead of TI's CHR$(8-11). It also will send data directly to the PIO port, for which I found a use in getting various text files).

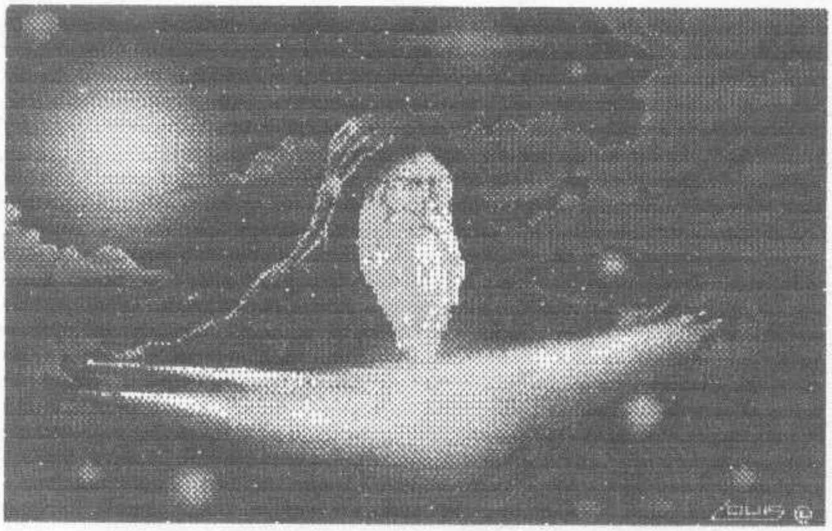
To come: I have the documents for Xmodem, Ymodem, Ymodem G, and TIbbs Xmodem. I am going to incorporate these and Zmodem into the program. An autodialer and menu system is coming... it is not all that easy!

Anyhow, I would just like to mention to those who might like a copy that it will be distributed... just ask for it, I'll upload it to Texlink. It is by NO means complete, but at the point it is, it leaves all TI telecommunication programs I have seen, in the dust! Or will leave... I use this program every day to get my mail from FreeNet (Ottawa FreeNet, bb737) and from Texlink and a board run by Mike Ward called The FlipSide.

As a last note, questions and comments are appreciated... send them via mail or GEnie (through Lucie if she doesn't mind), Texlink, or The TI Echo... Even FreeNet if you like.

Bye!!

[Editor's note: Jeff's Internet Email address is: bb737@freenet.carleton.ca
Texlink BBS is (613) 738-0617
FlipSide BBS is (613) 236-6717 and (613)236-0177]



TOP PSYCHIC'S EIGHT TOP PREDICTIONS FOR THE TI WORLD IN 1995!!

====================

by M.Brent (and captures from the TI Echo)

(sorry, Beery. Couldn't resist a tabloid-style header! Captured direct from the TI Fidonet Echo:)

from: Beery Miller
on : Tue 20-Sep-1994 5:43p

Several things are happening within the TI Community, this year being the poorest of them all.

I have the following predictions for the TI Community............

1) GEnie and Delphi will remain for 1995. GEnie will outlive Delphi, primarily because it's TI File database is the most extensive, next to my BBS and doesn't close areas as long as their is a volunteer someplace (and I'll be there <g>). I expect Delphi to remain in the game almost as long as GEnie.

2) This will be the last year for the Chicago Fair and will leave the Chicago User Group in a financial bind.

3) I predict the SCSI project will be abandoned officially, unless someone

pulls a rabbit from their hat (which they have been claiming they could do for the last 2 years).

4) Fewer than 6 new TI programs will hit the TI Community next year, including commercial and fairware offerings outside of any "bug" or "maintenance" releases of programs.

5) Many TI BBS's will shut down, leaving only a few to allow users to communicate.

6) Intemet access for TI'ers will continue to increase as access becomes available to them.

7) We see the loss of many renowned TI'ers in the community to other platforms and activities (i'm not going to name names, though I know quite a few "big" names that are moving that direction).

8) That's it. This is the TI's last year, future years are strictly for people just holding on (though it seems many of us have held on for quite some time).

Later

---
* Origin: -9640 News BBS- MidSouth 99'er BBS **1-901-368-0112** (1:123/50)

* * *

[I left the origin line in there, in case someone wants to call the system! Free Advertising!!]

Anyway... I certainly hope that Beery is NOT the top psychic that I billed him as at the top. <grin>

The TI community will exist for as long as there are truly interested people. It's been nearly 10 years since the machine was out of production, and a good 15 since it was designed, and it still has it's niche. This is a very impressive record.

Even though people are leaving the TI, it's my hope that they won't all abandon it. With the emulators available, people can pick one that they like, get it going, and continue to relive the TI as long as they wish. My own intent is to do some developing once my Amiga TI Emulator is finished. I know there is still software that people will use on their TI, if it's only available. (What, exactly, I'm still looking at.)

The big problem is one of support. But I don't mean just BY the programmers, I also mean OF the programmers. Consider my next capture from the TI-Echo:

from: Tim Tesch
on : Sun 18-Sep-1994 12:21a

Indeed, there IS a color ansi terminal program (PORT) in the works for the Geneve. I have already released one program to demonstrate the color ansi, called VCLR, which I got little to NO response from. While people have expressed an interest in PORT, I never seem to get much reply from the users. The only time I seem to hear anything is when a program is released with a small bug. Maybe I should but bugs in all of my programs to elicit responses?

In essence, I can truly understand why there are not many programmers left in the TI community. NO ONE EVER RESPONDS when an author releases a new program or a new version of a program. (oh, perhaps NO ONE is a bit too strong; almost no one is more appropriate). I program because I enjoy writing for hte Geneve and the TI, but when people complain and say this program or that program is not released fast enough, I have the mindset to just drop all of my efforts and give up on the community.

I am not saying you or any other particular person has been ungrateful, but when other people start deciding for ME when MY programs will be released, I can't help but become a bit frustrated - this is why I try not to promise release dates. If anyone wants to take over, I'll gladly pass the code onto them. Sound drastic? Maybe so... but as you can see, I'm getting sick and tired being criticized for items not being released when people feel they ought to be public.

Perhaps if I was a big company with 10 or 20 employees, I could release programs more efficiently and in a timely manner. This goes for other programmers in the TI community also. I'm sure many of them (of those that are still around) have many of the same concerns and opinions I've just voiced.

Perhaps a bit more support of the authors is necessary?? We like to hear about bugs, but we also like to know when our programs are used and if the effort was truly appreciated, maybe a note saying "thank you, your program works for me". Monetary gifts are not the only support items we look for.

Perhaps you were not expecting this response, but I feel it is about time to speak-up on this matter. I suppose that if it weren't for the programmers, no one would have anything to complain about. Maybe that is what the TI community....all programmers to quit their efforts. Sound bad? It could happen....

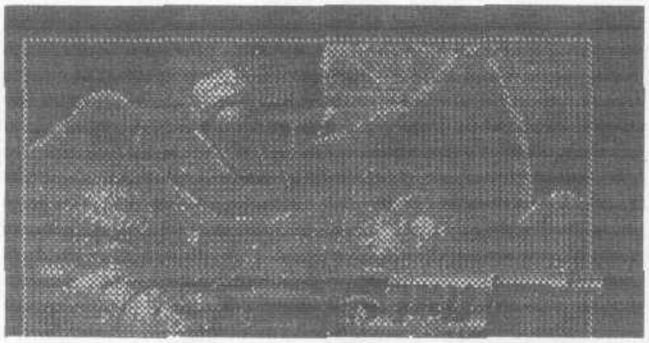
Tim

--- QuickBBS 2.76a
 * Origin: 9640 NEWS BBS * Memphis,TN! (901)368-0112 * (1:123/50)

* * *

This is a COMMON complaint from programmers. The users demand the software, they complain when it doesn't work, but nobody ever seems to write or call just to say "Good work".

Many programmers are not in it for money. I think at this point that it's safe to say MOST TI programmers are not. But feedback is essential if you want them to keep going. It's time now for the users to bear as much of the responsibility for supporting the TI world as the programmers.



Naturally, sometimes the problem comes down to one of distribution. The gentleman to whom Tim was replying to obviously didn't know that there was anything related to a colour ANSI-compliant terminal program for the Geneve. Once again, though, the brunt for distribution must come down to the users.

If you call several TI BBSs, and find a new program on one of them, either that does something very well, or even just an early version of a program that has not been done before, the programmer wants feedback. Send the program to all the other systems that you call... it doesn't really take that long in the grand scheme of things. Contact the programmer, if he left a means to do so in the docs, and tell him what you thought of the program. Tell him it was really good, or that it was horrible. But then tell him WHY it was horrible. What was wrong with it? What could improve it? The more feedback of this nature the programmer gets, the more likely he will incorporate your ideas into his next release.

Mind you, not everything you suggest may be incorporated. The programmer has his own ideas, too, and may disagree with some of your ideas. I've done it, I'm sure every programmer has. Don't take it negatively, but accept it. Realize that you've now done your part, and the programmer knows that someone looked at his program and was at least interested enough to comment.

EVERYONE must take SOME level of responsibility to keep the TI world going. Admittedly, if you do not have a modem, keeping in touch is getting more difficult. But perhaps you can start a mail exchange with a TI'er somewhere else in the world? A pen-pal on floppy, so to speak? There's always a way.

But now is the time. The TI has always had a lot going for it, but it's best aspect has always been the dedication of it's users. Time for everyone to do their part to help the TI world stay alive. It can be done, and it's not as hard as you may think.

I apologize for the length of this, but I agree with what Tim said, and fear what Beery said, so I felt this would be a good place to voice my comments.

Besides, I'm the editor. So there. <grin>