

8710
(965)



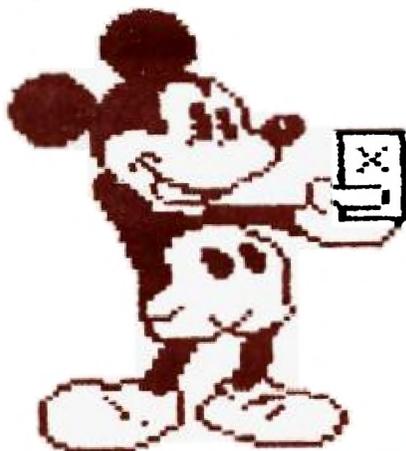
The Ottawa T.I.99/4A Users' Group



VOLUME 6 NUMBER 07.....OCTOBER 1987

DOUBLE FEATURE

OCTOBER 6



ELECTIONS

AND THE AUCTION GOES ON



P.O. BOX 2144, STATION D, OTTAWA
*** ONTARIO, CANADA K1P 5W3 ***

EDITORS NOTES

by Marg. O'Connor

I am a little sad to be stepping down as editor but as things stand I think it wise to let someone else try their hand at it and maybe they will bring fresh new ideas to the newsletter.

Ruth O'Neil has said that she will take over in November and I am sure she will succeed in doing a bang up job. I hope that you, the members, will support her as you have done in the past for me. Please bring her lots of articles for her to print, she needs them to make a good newsletter.

Ruth has a modem so you can download to her. Her phone number is 234-8050. Lets make it better than ever.

I will say goodbye now it has been fun and welcome Ruth and hope she has as much fun as I have had in the past.

Don't forget the deadline for next month.

Jane has asked me to tell everyone to keep MARCH 5, 1988 free for our THIRD TI-FEST and ask for volunteers as it takes a lot of hands to make a show a success, so get out and help where you can. WE NEED YOU. See Jane at the meeting.

NOTES FROM THE VC

by Lloyd Galanzoski

This month our meeting will see the election of a new slate of officers. They will need our support for the many activities and projects that the club conducts. An extra hand is always welcome. Don't hesitate to offer whatever assistance you feel can be provided. Our club needs you!

At our last meeting we sold about half of Berry's inventory. The sale exceeded my expectations and I'm therefore looking forward to the next auction. The popularity of this method of giving members access to TI software and hardware suggests to me that periodic auctions would be a good thing.

You are reminded to check last month's Newsletter for a listing of the meeting dates, through to June 1988.

WRITER-WROUTES

by Jane Laflamme

This month's article will be just a little different. I'm throwing a few little tips about TI-Writer in Editor mode. (Hopefully, it will help those of you that are using that great Australian creation, FunnelWriter, but do not have full documentation on TI-Writer. And may I take this opportunity, on behalf of the Ottawa TI-99/4A Users' Group, to thank Will McGovern and his fellow Australians for such a wonderful utility.)

Did you know?

That in PRINT FILE - in command mode, function 9, type in PF (cr):
You can "print" a file to disk? DSK(n).FILENAME
You can strip all control codes from a file? C DSK(n).FILENAME
You can write a DIS/FIX file? F DSK(n).FILENAME
Print line numbers with your file? L DSK(n).FILENAME
You can write a partial file AND strip control codes?
C 345 655 DSK(n).FILENAME

(In this example, lines 345 to 655 have been printed to disk and stripped of all control codes.)

Did you know?

That rather than deleting many lines within a document, you can save partial files and then merge them together?

In command mode, type SF (Save File) (cr) then:

1 145 DSK(n).NAME1

500 E DSK(n).NAME2

In command mode, type LF (Load File), (cr), then:

DSK(n).NAME1

In command mode again, LF (cr), then:

145 DSK(n).NAME2 (or "E DSK(n).NAME2")

where after line "145" of the first file, (or "E" for end), is where you wish the second file to be placed. When this file is saved under its original name, you have in effect, deleted 355 lines much faster than the delete function itself.

Did you know?

To get to the top or bottom of a large document in a split second, in command mode, type S for Show (cr), then 1 (cr) for the first line, or E (cr), for the end of the document. (Good for the Replace String search when you wish to start at the top of a document.)

You can replace control codes in the replace string function by entering control U?

You can replace string with a "null"; eg: RS(cr), /HELLO//. Here, "HELLO"s will be deleted.

You can replace or search string within specified columns; eg: RS(cr) or FS(cr).

Example 1: 1 2 /Hi//. In this instance, it will search columns 1 and 2 for the string "Hi" and delete it.

Example 2: 34 39 /H/, it will search columns 34 to 39 and find the first instance of an "H". It will only find both strings AFTER the cursor. To search the whole document, place cursor in row 1, column 1.

If you have a document without carriage returns, you can use the replace string in fixed mode (Control Ø). That way, TI-Writer will not make a jumbled mess of your document.

You can "cut and paste" with the best of the word processors. Place your cursor at the beginning of the block you wish to move (can be mid-line), and then press "Insert character", Function 2; move your cursor to the end of the block, and again, press "Insert Character". Then you can enter Move from command mode and place the block AFTER any line you wish. (Should it be in the middle of the line, break the line by once again using "insert character".) Reformat (Control 2) and voila!

And finally, after 2 years of publishing our great Newsletter, Margaret O'Connor has felt it is time to step down. Thank you Margaret for putting up with our continuous late deadlines, complaints, and headaches associated with a major undertaking such as this. And thank you also to husband John for picking up my article which was just about always too late or I was too busy to drop off, and promised myself it wouldn't happen next month but did!! You both have been great. Thank you again.

And to Ruth O'Neill who is about to undertake our Newsletter, it's great to have another woman on board, and best of luck. (I promised to try and be better...)

GAME SCORES
BY
SCOTT BALDWIN

GAME	POSITION	SCORE	PERSON(S)	TIME LEFT
ANTEATER	1st	016,848	DOUG BOONE	(3)
ANTEATER	2nd	000,000	-----	
ANTEATER	3rd	000,000	-----	
BARRAGGE (DOUBLE)	1st	810,000	DAN HOPKINS	
BARRAGGE	2nd	536,090	STEVE WOODARD	
BARRAGGE	3rd	000,000	SCOTT BALDWIN	
BARRAGGE			PATRICK MURRY	
BARRAGGE (SINGLE)	1st	889,250	CHRIS GILLES	
BARRAGGE	2nd	155,250	DOUG BOONE	
BARRAGGE	3rd	000,000	-----	
BLASTO	1st	039,110	CHRIS GILLES	(3)
BLASTO	2nd	000,000	-----	
BLASTO	3rd	000,000	-----	
BUCK RODGERS	1st	079,500	DOUG BOONE	
BUCK RODGERS	2nd	000,000	-----	
BUCK RODGERS	3rd	000,000	-----	
BURGERTIME	1st	284,250	CHRIS GILLES	
BURGERTIME	2nd	129,400	DEVIN JOHNSTONE	
BURGERTIME	3rd	104,700	GREGORY SIMMONDS	
DEFENDER	1st	773,900	CHRIS GILLES	(3)
DEFENDER	2nd	000,000	-----	
DEFENDER	3rd	000,000	-----	
DIG-DUG	1st	037,620	DOUG BOONE	
DIG-DIG	2nd	023,070	KATHY BERLINGUETTE	
DIG-DUG	3rd	000,000	-----	
DONKYKONG	1st	810,200	DANIEL HOPIKINS	
DONKYKONG	2nd	042,600	MIKE BERLINGUETTE	
DONKYKONG	3rd	000,000	-----	
GUARDIAN	1st	155,410	KEAT MADDISON	
GUARDIAN	2nd	100,500	GREGORY SIMMONDS	
GUARDIAN	3rd	012,470	MIKE BERLINGUETTE	
HENHOUSE	1st	056,320	DOMINIQUE LAFLAMME	
HENHOUSE	2nd	000,000	-----	
HENHOUSE	3rd	000,000	-----	
HUSTLE	1st	019,900	TANYA GRAIG	
HUSTLE	2nd	000,000	-----	
HUSTLE	3rd	000,000	-----	
JAWBREAKER	1st	012,470	AMANDA MCALLISTER	
JAWBREAKER	2nd	008,100	GREGORY SIMMONDS	
JAWBREAKER	3rd	005,320	JEANNIE LAFLAMME	
JUMPY	1st	179,000	PHILIPPE TANGUAY	
JUMPY	2nd	130,700	JIM JONES	
JUMPY	3rd	000,000	-----	
MICRO- PINBALL	1st	220,000	BENOIT TANGUAY	
MICRO- PINBALL	2nd	175,400	SHAWN MILLAR	
MICRO- PINBALL	3rd	148,400	SCOTT BALDWIN	
MINER 2049'er	1st	065,350	DANIEL HOPIKINS	
MINER 2049'er	2nd	021,850	DOMINIQUE LAFLAMME	
MINER 2049'er	3rd	010,535	DOUG BOONE	

MOONMINE 1st 079,810 CHRIS GILLES
MOONMINE 2nd 000,000
MOONMINE 3rd 000,000

MOONPATROL 1st 030,950 DANIEL B. MCORMIC
MOONPATROL 2nd 000,000
MOONPATROL 3rd 000,000

MOONPATROL 1st 039,250 SHAWN MILLER
CHAMP
MOONPATROL 2nd 026,850 SHAWN MILLER
CHAMP
MOONPATROL 3rd 020,100 GARY MILLER
CHAMP

MUNCHMAN 1st 025,000 GREGORY SIMMONDS
MUNCHMAN 2nd 017,850 DOUG BOONE
MUNCHMAN 3rd 000,000

FACMAN 1st 030,510 DOUG BOONE
FACMAN 2nd 030,360 GREGORY SIMMONDS
FACMAN 3rd 000,000

PAR-FORE 1st 000,061 TIM SLACK (3)
GOLF
PAR-FORE 2nd 000,000
GOLF
PAR-FORE 3rd 000,000
GOLF

PARSEC 1st 532,800 PETER ARPIN
PARSEC 2nd 369,000 BENJIT TANGUAY
PARSEC 3rd 000,000

POLE POSITION 1st 408,800 KEAT MADDISON
POLE POSITION 2nd 234,470 PHILIPPE TANGUAY
POLE POSITION 3rd 090,920 PAUL BERLINGUETTE

SLYMOIDS 1st 032,155 DOUG BOONE
SLYMOIDS 2nd 024,935 DANIEL LAFLAMME
SLYMOIDS 3rd 000,000

TI INVADERS 1st 025,682 GREGORY SIMMONDS
TI INVADERS 2nd 007,860 KATE MCALLISTER
TI INVADERS 3rd 002,400 PHILIPPE TANGUAY

TI RUNNER 1st 062,600 SHAWN MILLAR
TI RUNNER 2nd 028,400 DAN LAFLAMME
TI RUNNER 3rd 012,200 MIKE BERLINGUETTE

TOPPER 1st 032,570 GARY MILLAR
TOPPER 2nd 030,190 SHAWN MILLAR
TOPPER 3rd 000,000

WORD RACE 1st 016,380 BOB BOONE (3)
-HARDEST
WORD RACE 2nd 000,000
-HARDEST
WORD RACE 3rd 000,000
-HARDEST

IF ANYONE HAS ANY QUESTIONS, COMMENTS, COMPLAINTS OR EVEN A SUGGESTION PLEASE LEAVE A MESSAGE FOR ID#54 ON THE CLUB BBS OR GIVE ME A CALL AT (613)523-5573.

NOTE: IF SINGLE SCORES SUCH AS WORD- RACE-HARDEST ARE NOT CHANGED WITHIN 3 MONTHS THE SCORE AND THE GAME WILL BE DELETED FROM THE LIST I ALSO WILL NOT ADD ANY GAME TO THIS LIST. THE AMOUNT OF TIME LEFT WILL BE PLACED IN BRACKETS AFTER THE NAME.

BASIC TI INPUTS AND OUTPUTS PART 5

(conclusion)

BUILDING AN EXTENSION KEYBOARD

by David Caron

NOTICE: See second paragraph concerning the reliability of the keyboard pin numbers.

The Table presented in the last newsletter (September 1987) is very beneficial for input from external devices (not necessarily keyboards). By simply connecting two wires together manually or through a transistor, a simple BASIC program with CALL KEY can detect that connection.

Incidentally I should warn you that the keyboard pins numbers are correct in comparison with the connector of my keyboard, BUT are REVERSED in comparison with the keyboard mapping schematic of the TI-99/QI (the beige computer). Because there are numerous other errors in the schematics, I suspect that this may simply be another one, therefore my pin numbers are probably right and the schematics wrong (thats the way it usually is). If I have not yet convinced you, I will also point out that the "Q" line (pin 3 on schematics) and the "A" line (pin 10 on the schematics) are REVERSED. THIS HAS BEEN EXPERIMENTALLY TESTED, THE HARD WAY. If you are still in doubt as to which pin system is correct, then I suggest that you manually connect two of the pins yourself and see what CALL KEY returns (to do this of course you will have to separate your console into five pieces or more to get at the keyboard bus). Should you decide that I am wrong and the schematics correct, you can simply "fix" the keyboard bus table by substituting 15 for 1, 14 for 2, 13 for 3, etc... Enough of that.

If you have not already noticed you might notice that CALL KEY detects things like CTRL 1-7, and also things like FCTN J,K,L,Q etc... If you find yourselves running out of FCTN's and CTRL's here are some more. The ascii codes for these undocumented special keys are given in the keyboard bus table.

Another great application using the keyboard bus is an additional keyboard so you can push the console to the back of your desk, giving you more room for "other things". (take care not to push the console off the end of your desk in your enthusiasm to make your workspace look uncluttered). This operation involves a male and female 15-pin connector and at least 6 feet of cable, (ribbon or coaxial), a spare keyboard, a flat type 15 pin connector (Active has these with about 26-30 pins which can be broken off into proper lengths). All these components can be obtained at Active with the exception of the keyboard which can be obtained from ADDISON for less than \$3.00.

DIRECTIONS:

1. Take the computer apart to the point where only the keyboard remains in the casing. If you feel you need advice on this operation, call me at 745-4618. However the only trouble you should have is with the ON/OFF switch on black and silver consoles. The switch must be pulled out directly away from the console.
2. Drill, cut, melt or punch a hole in the region left and up of the joystick port where the surface of the console levels out, making sure the hole is barely larger than the size of the male connector. Do this in small steps to avoid "overshooting" the size. Remember that the connector must be mounted on the outside of the console NOT THE INSIDE. A file should be used near the end of the operation.
3. Drill two holes on either side for nut and bolt type of screws. When mounting the 15-pin MALE connector, the bolts go on the outside and the nuts on the inside.
4. This is the tricky part. If you bought a non-solder type FEMALE connector for easy connection to a ribbon cable then you might as well solder individual wires from the internal keyboard bus to the MALE connector in the same fashion as the ribbon cable

on the FEMALE connector. HOWEVER if the condition is not true then you should solder individual wires between the internal keyboard bus and MALE connector, pin for pin.

5. Connect the cable to the FEMALE connector in the same fashion as above using the conditions. Try to make the odd colour wire pin 15.

6. Solder the other end of the cable to a flat type 15-pin connector similar to the one on the mother board.

7. Reconnect the mother board and powersupply and test the extention keyboard. If problems are apparent use the CALL KEY routine and compare the ascii codes with the chart from the September 1987 issue.

And that's it folks! I've depleted everything I know about INTERFACING BASIC with the outside world via the keyboard bus, joystick connector, and cassette connector.

ARE YOU T.I.R.E.? (TI RUNNER EXPERT)

by Henri Monat, M.R.A. (Master in Runner Administration)

This article is an analysis of TI-Runner, by EB Software and gives our family evaluation of what we consider to be the best action game ever made for the I.I.99/4A.

RULES OF THE GAME

In this game, our heroe (the Runner) must collect keys, treasures and other goodies, without being caught by some funny joggers. To do so, our Runner must manage to find the best route between bricks, walls and ladders. But be carefull: the joggers are also collecting the same and if they touch you, you are dead. Fortunately, you may defend your-self: by pressing the fire button, you make a hole in the bricks for few seconds and any joggers who falls into it looses the treasure until it be picked it up again. When you have picked up all the goodies for that level, new ladders appear on the screen which you must climb up to the next level. Each level is in fact a new screen, having new bricks, new walls, new ladders, having also new treasures and goodies but having still the same good old joggers. You so proceed to run and run again, and climb and climb again, from level to level, from screen to screen, from ladders to ladders until such time as you find out that you have reached ... your level of incompetence.

PETER'S PRINCIPLE REVISITED

We are all familiar with the famous Principle of Laurence Peter:

"In every hierarchy, whether it be government or bussiness, each employee tends to rise to his (which ought to be read "his or her") level of incompetence."

TI-Runner questions now this well accepted Principle. Discover it by your-self: load the program and play; when you have reached your level of incompetence, just press FCTN 5 and see what happens: your favourite game skip over this level and jumps to the next one. No bonus is given to you though, but no damage was done.

Pressing FCTN 5 is not just a mere routine gesture. It is a whole revolution! To give you an idea of how powerfull it is, here are some practical applications:

- you can save and accumulate lives to concentrate your efforts on a target screen;
- you can skip over boring screens;
- you can satisfy your curiosity by seeing what is after level 50;
- you can search for nice screens because you like that;
- you can search for easy screens to get experience;
- you can look for tough challenging screens;
- how about non accountable benefits such as mental health, relief

from stress, relaxation, or even pretending, when you socialize, that you cannot reach your level of incompetence.

Since TI-Runner is teaching us a new fact of life, I would reformulate Peter's Principle and call it Runner's Principle:

"In every hierarchy, whether it be government, bussiness or leisure activities, people tend to rise to their level of incompetence, unless they can jump to a higher level."

OTHER POINTS OF INTEREST

Don't fool around by pressing FCTN 6 or 8, unless you need to. FCTN 6 makes you begin the same screen again; FCTN 8 returns you to level 1. The "P" key is for pause. It is a good strategy to make many pauses: it permits you to re-asses the situation and change your plan accordingly.

Before beginning a new screen, make a route plan; in some cases, there is no "come-back". Find safe places; sometime, there are places where you can safely stay and observe how joggers behave. Take risks; you may have pleasant surprises.

Be patient; for some difficult screens, it took me up to 50 lives to solve them; a good game may take 2 hours or more.

Thanks to the authors, SCOTT EMERY and JON BURT of EB SOFTWARE for having thought of such a wonderfull and terrific game.

QUIZ

Since I pretend to detain the record (hurray for my humbleness) of TI-Runner, I claim a right to ask a little question to see if there was an interest in writing this article.

There are screens that have some kind of meaning: there is for example a boat, or a castle, or an I.B.M. logo. The question is: HOW MANY SCREENS CAN YOU RELATE TO T.I. AND WHICH ONES?

If you find none, try the Runner's Principle; if you can find 1 or 2, you are becomming a "gameaucrat" (a "gameaucrat" is somebody who plays video games just for fun and likes to talk about it); if you can find 3, you are a Runner addict; if you can find 4, I would like to know if you can beat 199,800 points for a game of 3.30 hours and I will give you an M.R.A. (Master in Runner Administration); if you can find more than 4, either you have a lot of imagination or you have gone back to Peter's Principle...

EVALUATION

The following chart is our family evaluation of each of the screens and is self explanatory. Anybody can get in touch with me for a supplementary clue or hint, but unless you twist my arm (in which case get ready for a legal action), I will not give the solution for any screen.

TI-RUNNER EVALUATION CHART

GRAPHIC	LEVEL OF DIFFICULTY	PLEASURE	TIPS/HINTS/COMMENTS
01 FAIR	1 EASY	FLAT	
02 FAIR	1 EASY	FLAT	
03 NICE	2 EASY	FLAT	
04 FAIR	3+MEDIUM	FLAT	KEEP ONE JOGGER OUT
05 NICE	3 MEDIUM	GOOD	
06 VERY NICE	2 EASY	GOOD	
07 FAIR	2 MEDIUM	GOOD	
08 FAIR	3 MEDIUM	GOOD	
09 FAIR	3 MEDIUM	FLAT	
10 NICE	3 MEDIUM	GOOD	
11 NICE	4 HARD	VERY GOOD	HAVE JOGGERS FOLLOW THE GUIDE
12 NICE	3+HARD	VERY GOOD	

13	VERY NICE	4+HARD	VERY GOOD	HURRY UP! JOGGER IS COMING DOWN
14	NICE	3 MEDIUM	GOOD	
15	NICE	5+HARD	BORING	BE THE GUIDE BUT GET A MAP
16	NICE	2 EASY	GOOD	
17	NICE	3+MEDIUM	GOOD	
18	NICE	2 EASY	FLAT	
19	VERY NICE	4 HARD	VERY GOOD	RELAX NEAR THE BORDERS
20	VERY NICE	3 MEDIUM	GOOD	
21	VERY NICE	5+VERY HARD	VERY GOOD	STAY IN THE CENTER
22	VERY NICE	4 HARD	VERY GOOD	
23	VERY NICE	3 MEDIUM	VERY GOOD	
24	VERY NICE	5 VERY HARD	BORING	STAY UP AND BE PATIENT
25	NICE	5 HARD	VERY GOOD	
26	VERY NICE	4 HARD	VERY GOOD	
27	EXCELLENT	3 MEDIUM	VERY GOOD	SEND COMPANY TO LONELY JOGGERS
28	VERY NICE	5+VERY HARD	VERY GOOD	
29	NICE	5+VERY HARD	EXCELLENT	
30	EXCELLENT	5 HARD	EXCELLENT	OPEN THE DOOR FIRST
31	NICE	3 MEDIUM	VERY GOOD	
32	VERY NICE	3+MEDIUM	EXCELLENT	
33	EXCELLENT	5 HARD	EXCELLENT	MATE THE MUTINY AT THE STERN
34	EXCELLENT	5+VERY HARD	EXCELLENT	LET JOGGER TRY TO EAT HEART FIRST
35	VERY NICE	5 HARD	EXCELLENT	
36	VERY NICE	3 MEDIUM	VERY GOOD	
37	NICE	5+VERY HARD	VERY GOOD	
38	VERY NICE	5+VERY HARD	?	RUNNER'S PRINCIPLE SO FAR
39	NICE	5+VERY HARD	?	RUNNER'S PRINCIPLE SO FAR
40	VERY NICE	3 MEDIUM	VERY GOOD	
41	VERY NICE	5+VERY HARD	EXCELLENT	
42	EXCELLENT	5+VERY HARD	?	RUNNER'S PRINCIPLE SO FAR
43	NICE	3 MEDIUM	VERY GOOD	
44	VERY NICE	4 HARD	EXCELLENT	
45	VERY NICE	3 MEDIUM	VERY GOOD	
46	VERY NICE	5 HARD	EXCELLENT	DAMMAGED BRICKS WILL BE REBUILT
47	VERY NICE	3 MEDIUM	EXCELLENT	
48	VERY NICE	5+EXCELLENT	?	RUNNER'S PRINCIPLE SO FAR
49	EXCELLENT	5+VERY HARD	EXCELLENT	KEEP THAT JOGGER OUT OF THAT HOLE
50	VERY NICE	5+VERY HARD	EXCELLENT	

```

*****
*                                     TRADING POST                                     *
*                                     -----                                     *
*                                     FOR SALE                                       *
*                                                                              *
* HARDWARE:- TI 99/4A, STANDARD DISK DRIVE, COLOUR MONITOR *
* JOYSTICKS, SPEECH SYNTHESIZER, P. E BOX, 2 1/2 HEIGHT *
* DRIVES, TAPE RECORDER, ROLLER CONTROLLER, WIDGET, MEN. *
* EXPANSION CARD, SIGNALMAN MODEM. *
*                                                                              *
* CARTRIDGES:- WITH DOCS AND MANUALS TI INVADERS, PARSEC, *
* CAR WARS, ALPINE, MOONMINE, MUNCHMAN. *
* TEACH YOURSELF BASIC, HOUSEHOLD BUDGET MAN., DISK MANAGER, *
* MULTIPLAN, EX. BASIC, TERMINAL EMULATOR II, TI WRITER, *
* TI LOGO, EDITOR/ASSEMBLER. *
*                                                                              *
* DISCS:- 4 - BASIC AND EXTENDED BASIC GAMES, GENEALOGY, *
* HANDI-DISC, TI- ARTIST + ENHANCEMENTS, NUTRI-GUIDE, *
* FUNNELWEB, GRAPHX (3), TI FORTH - 4.5 *
*                                                                              *
* BOOKS:- FUNDAMENTALS OF TI ASSEMBLY LANGUAGE, 101 PROG. *
* FOR TI, PROG. AND GAMES FOR TI, GUIDE TO THE TI, COMPUTE *
* MAG., TI USER'S GUIDE, COMPUTERS BOOK OF TI GAMES, PROG. *
* BASIC TUTORIAL, STIMULATING SIMULATIONS FOR THE TI, TIPS *
* AND TRICKS FOR THE TI WRITER, INTRO. TO ASSEMBLY LANGUAGE *
* FOR THE TI, BEGINNER'S BASIC, BACK ISSUES OF OTUG NEWS *
* LETTER: JAN. 86 - CURRENT. *
*                                                                              *
* PRICE:- $1000. COMPLETE *
*                                                                              *
* CONTACT:- ANN GONNEAU AT 835-3344 *
*                                                                              *
*****

```

TI BASIC continued from September

by Steven Shaw

Just as with English, your ability to use BASIC increases with use. Examine as many 99/4A programs as you can and look to see

- i)WHAT each part of the program does
- ii)HOW it does it
- iii)WHY it does it.

Then try to improve the program!

COLOUR DEMONSTRATION PROGRAM:

This program has been provided to give a practical demonstration of some of the features of TI BASIC described earlier.

The program has been written in small blocks, and each block will be described separately.

The first section, lines 100 to 420 form the start of the program. The first block is intended to display the colours available, and by making them cross over each other, show the relative contrasts.

Because some random patterns are created later, RANDOMIZE has been used to provide different patterns each time the program is run. The array F contains frequency values for use with CALL SOUND later on. As it will contain 100 values, DIM is used to instruct the computer to allocate memory to hold the values.

The DEF function is used to create a new random function, which will provide integer (eg no fraction) numbers from 1 to the figure used with the new function in the program: watch out for the new function RAN(X) in the program.

M\$ has been set to "0" (a string with a zero in it) for use in defining all the characters as blanks (eg spaces) in the following loop in lines 180 to 200. The string "0" could have been placed in the definition function in line 190, instead of the string variable.

NB: Although the TI99/4A distinguishes the number 0 from the letter O on screen by squaring the O, in listings a slashed 0 usually represents the number.

Each group of eight characters is in a separate character set, and each set may be a different colour. Lines 210 to 240 change the colour of each set: TI Basic has 16 sets, and 16 colours. The foreground colour of every set has been set to WHITE (code 16).

Line 230 places a number of vertical stripes on the screen, each a different colour. As the screen is 32 columns wide, the stripes have been set to 2 columns wide each, and therefore each CALL VCHAR uses 48 characters (2x the 24 rows).

So that we know which colours are which, they are labelled by the routine in lines 250 to 340.

Lines 350-360 give a small delay.

It is not possible to provide 16 horizontal bands with a space between each, as the screen only has 24 rows, but lines 370 to 390 cross the screen with as many rows as can fit.

In lines 430 to 500, random characters are placed on a blank screen, and in lines 510 to 590, random stripes and columns are placed on screen.

Lines 600 to 690 redefine the characters for the purposes of the following sections of the program. Remember that the strings used with CALL CHAR can only contain the numbers 0 to 9 and the letters A to F.

700 to 720 again place random characters on screen.

Lines 730 to 770 fill the F array with values to be used with subsequent CALL SOUNDS. Line 730 sets the lowest possible frequency, and line 740 sets the basis for the tones: the formula used creates what is known as 'microtonal' music, with very little difference between adjacent tones.

Then, accompanied with random tones, the character colours

are varied at random in lines 780 to 830.

Lines 840 to 960 provide sound, colour changes, and small bars of random characters.

Lines 970 to 1040 use the PRINT routine to place random characters on screen, and the colour is varied in 1050 to 1070.

In lines 1080 to 1200, characters are given a random definition and in 1220 to 1260, the colours are varied, random characters placed and tones generated.

1270 to 1310 again varies the colours, and uses a different random tone generation method (line 1300).

The remainder of the program again uses PRINT to provide a random display and the colours are varied.

Note in particular the usefulness of the DEF statement in this program.

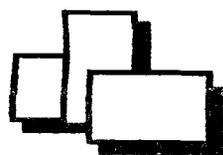
There are many 'loops', and some loops contain other loops: see for example lines 840 to 960. These loops are 'nested', with the COUNT loop inside the Z loop.

An ARRAY is used to store frequencies.

The loop counters (eg CHARR, SET and so on) also function as numeric variables in the loops. Their value increases by one for each cycle of the loop until the maximum value (set by the TO X in FOR TO NEXT) has been reached.

In line 1120, SEG\$ is used with the string variable M\$ (set in 1080) to create a random definition of a character. A letter is chosen at random from the string variable M\$, and used to create the definition in the variable P\$.

Note that P\$ is reset to a 'nul' (empty) string after each character has been defined. Then it is reused with different letters to form a new definition.



GENEVE 9640

NOW IN STOCK

CONTACT A DEALER

FOR MORE INFORMATION

COMPUTER DOWNLOAD
UNLIMITED

CANARIA DATA INC.

25 Ottawa St.
ARNPRIOR, Ontario
K7S 1N7

264 Weber St. W.
KITCHENER, Ontario
N2H 4A6

(613) 623-7841

(519) 578-3873

LAFLAMME & WRIGLEY WHOL. CAN. DISTRIBUTOR

TI BASIC GLOSSARY

A list of all the commands and functions available in TI BASIC, with brief descriptions.

ABS(X)	
APPEND	the + or - sign.
ATN(X)	Used to extend files when using the disk system or mini memory module.
BREAK	Provides the arctangent : this is a trigonometrical function. ATN(X) provides the angle in radians whose tangent is X.
BYE	Used in a program to temporarily halt execution. A message is placed on screen. The program is continued by typing in CONTINUE
CALL CHAR(N,STRING\$)	Used to return to the master screen. Preferable to the use of the QUIT key, as only BYE will properly close any open files, and ensure no data is lost.
CALL CLEAR	Used to define character N by means of a hexadecimal string STRING\$.
CALL COLOR(SET,FG,BG)	Used to clear the screen.
CALL GCHAR(ROW,COL,CH)	Used to defing the foreground (FG) and background (BG) colours of each of 16 sets of characters. Sixteen colours, including transparent are available.
CALL HCHAR(ROW,COL,CH,NO)	Places in the variable CH the ASCII code of the character on the screen in ROW, COLUMN.
CALL JOYST(NO,X,Y)	Used to display a character CH in ROW, COLUMN. When a number or variable in the position NO is used, the character is repeated horizontally NO times.
CALL KEY(NO,CODE,STATUS)	Returns X and Y values for Joystick NO.
CALL SCREEN(NUMB)	Interrogates the keyboard. If a key is depressed the ASCII code is placed in variable CODE. STATUS indicates if a key is pressed, and if it is the same key when two adjacent CALL KEYS detect a key pressed. NO indicates the key unit, and is used to split the keyboard and to switch the codes returned by the Function and Control keys.
CALL SOUND(TIME,F1,V1,F2,V2,F3,V3,N,V4)	Used to change the colour of the screen.
CALL VCHAR(ROW,COL,CH,NO)	Used to generate sound. Up to three tones may be used with an optional noise channel. TIME is in milliseconds and F1,F2 and F3 are the frequencies in cycles per second.
CHR\$(CODE)	As CALL HCHAR, but the character CH is repeated vertically NO times.
CLOSE	Used to make a character CODE available as a string.
CONTINUE	Used to close a data file.
COS(X)	Used to resume a program when execution has been halted with the CLEAR key or a BREAK command.
DATA	Provides the cosine of angle X, where X is in radians.
	Used as a heading on lines containing values to be READ.

DEF	Used to DEFine a function of your own.
DELETE	Used with the disk system to DELETE a file.
DIM	Used to DIMension an Array.
DISPLAY	<ul style="list-style-type: none"> i. Same effect as PRINT ii. One of two file storage formats. DISPLAY uses the same codes and format as the computer uses for screen displays.
EDIT	One method of entering EDIT mode.
END	An optional marker for the end of your program.
EOF	'End Of File' used with disk files.
EXP(NO)	The inverse of the natural logarithm function LOG. Thus $X=EXP(LOG(X))$
FIXED	Used to define data files. The alternative, not available with cassettes, is VARIABLE.
FOR...TO...(STEP)	Used to establish loops which execute until the counter reaches the value following TO.
GOSUB	Used for a line transfer when the program is to RETURN to the line following the GOSUB line after the section transferred to is completed.
GOTO	Used to make a simple line transfer.
IF...THEN...ELSE	Used to make conditional line transfers, with an optional alternative transfer if ELSE is used.
INPUT	<ul style="list-style-type: none"> i. Used to fill a variable from the keyboard, or other device if a data file is used. ii. Used to specify a file is to be used for INPUT only.
INT(NO)	Used to provide the INTeger of a number NO. eg any fraction is removed.
INTERNAL	Used to specify the format of a data file. INTERNAL specifies the codes used by the processor internally. The alternative is DISPLAY.
LEN(String#)	Used to provide the LENgth of the string String#.
LET	Optional. LET A=2 and A=2 are both accepted.
LIST	Lists a program on screen or other device.
LOG(NO)	Provides the natural logarithm of number NO.
NEW	Used to clear the console memory in preparation for a new program.
NUM / NUMBER	Provides line numbers automatically when a program is to be keyed in. Starting number and increment may be defined. Default is to start at 100 and increase each line number by 10.
OLD	Used to load a program from cassette or other storage device.
ON...GOSUB	Used for GOSUB transfers when the value following ON determines which of the line numbers following GOSUB are to be used.
ON...GOTO	Similar to the above, but for simple line transfers when it is not wished to RETURN to the line following the transfer.
OPEN	

need not be.

As the "Extended Basic Product Information" booklet (inserted into the XB manual) says, "careful planning is required"; and it then sets up five rules, all dutifully followed in the above program:

1. "Enter your first DATA statement within the pre-scan". In order to keep the pre-scanning to the minimum, and leave the data lines where I wanted them, I invented a dummy DATA (line 140), that I then read (but do not use further) in line 170. You could also frame your first data line by the pre-scan statements: here, we could add a line "195 !%P+", on, and "205 !%P-", off.
2. "Include the first use of each variable and/or array (also the OPTION BASE if used)". But what happens if you do not use all your variables at the beginning of the program? Well, XB provides you with the possibility to put them all in one line or two at the beginning; but you MUST precede that line with a GOTO to the next program line for it to execute properly. In our example, all the variables that are not used before the computer encounters the pre-scan line (which is line 160) are thus declared in line 120; since the variables T and V, and the NOTE() array, were used in line 110, they are not repeated in line 120.
3. "Include the first reference to each CALL statement of any subprogram", whether they be TI's or the ones you created yourself, like CALL WAIT. This is done in lines 120 and 130. You will notice that I have not declared the CALL KEY statement in line 400, since it will be pre-scanned with the SUB (see 5).
4. "Include all DEF statements for user-defined functions", as I did in line 150; the definition itself is used in line 280.
5. "Include all SUB statements and SUBEND statements in the pre-scan". That of course applies to the user-defined subs; since as a rule they have to be put at the very end of your program, the easiest thing to do is to put the pre-scan off statement on a line before your sub, as I have done in line 380. Since we learned last month that the variables in our own subs are totally independent from the ones in the body of the program (I could have used A and B instead of K and S), Tex needs to know them if you want it to save memory space for them too; the CALL KEY statement will also be pre-scanned here.

Now you know what is a PRE-SCAN; the best way to implement it is first to make sure that your program works, then gather all you variables and CALL statements. A long task, so I have written the following program; it will print a convenient form that you can use for your gathering of information. Note line 110, another way to put the "pre-scan off" statement in your program.

```
100 L$=RPT$(" ",34):: LL$=RPT$(" ",73):: S$=" " :: V$=" VARIABLES"
110 CALL CLEAR :: GOTO 120 :: PGM$,X :: !%P-
120 OPEN #1:"PIO" ! or your printer
130 PRINT "PRE-SCANNING"&V$: "FOR PROGRAM (title): " :
    :: INPUT ">":PGM$
140 PRINT #1:CHR$(27)&"@": " PRE-SCAN FOR: "&CHR$(14)&PGM$: :
    " will be on line " : :
150 PRINT #1:TAB(12);"NOMERIC"&V$;TAB(52);"STRING"&V$:S$&LL$
160 FOR X=65 TO 90 :: PRINT #1:S$&L$&"! "&CHR$(X)&"! "&L$ ::
    NEXT X
170 PRINT #1:"":":": " CALL STATEMENTS:":": " :: FOR X=1 TO 4 ::
    PRINT #1:S$&LL$ :: NEXT X
180 CLOSE #1 :: END
```

What happens if you forget to include a variable or a statement in your pre-scan? Well, Tex will inpolitely stop and warn you with a "* SYNTAX ERROR IN nn". Just go back to your pre-scanning line(s), and include it.

As for the extra goodies, they concern simpler ways to use the graphic CALL statements in XB; I will leave that for next month, and we will use the same program, so please keep it.



- Quality printing at low prices
- Special prices on photocopies

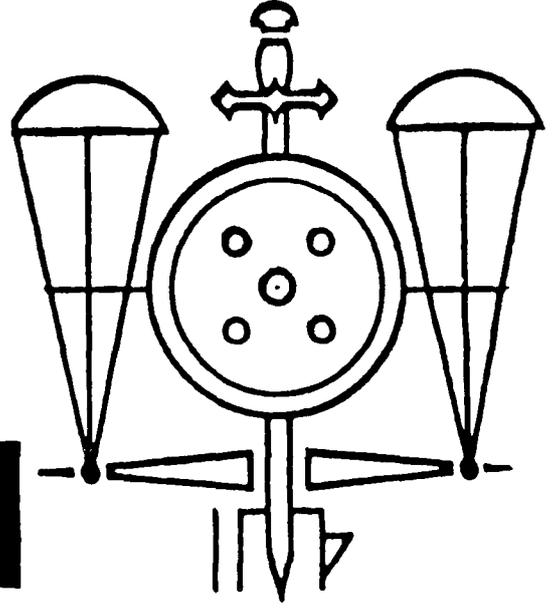
837-1844

2297 B St-Joseph Bl Orleans
 St-Joseph Plaza
 ORLEANS, Ontario K1C 1E7

OPEN
 MON 9:00 A.M.
 TO
 SAT 8:00 P.M.

OFFSET PRINTING • LAYOUT • PHOTOCOPYING

momin PRINT



FROM

P.O. BOX 2144, STATION D, OTTAWA
 *** ONTARIO, CANADA K1P 5W3 ***

EDMONTON USER'S GROUP
 BOX 11783
 EDMONTON, ALBERTA