CO-ORDINATOR'S REPORT

How do you use a computer? Do you use one at work, at home, for pleasure, for profit, to save paper, or just to have fun?. What sort of a user are you? Do you spend just a few minutes occasionally, or are you one of the large number who turn it on as soon as you get home? Do you get withdrawal symptoms, if you are stopped from having your daily fix? Can people talk to you while you are using it? Do you hear them? Is your computer an indispensable part of your life? Is your computer as important as your television, video, stereo? If your home was threatened by fire, where would your computer come in the list of things to be saved? Would you risk your life to save your TI? Is it a tool, or is it simply a piece of furniture?

Where do you keep your computer at home? I have seen computers used in attics, garages, bedrooms, living rooms etc. I know someone who uses a portable, a notebook, and even takes it sailing with him. He plugs it into the boat's 12 V battery. Some users even take their notebooks to bed with them, just to complete those outstanding reports you understand. Is your computer a status symbol? Obviously if you only use a TI, status is not an issue, but some users must have the newest, dearest, latest, computer development or accessory, to show off to their business colleagues. Some executives in the Dept. I work for, have quite sophisticated machines allocated to their offices. Most of them never even use them. Those who actually do the work, have to make do with outdated machines, if one is available. The possession of a computer is a status symbol for these officers. One of our members told me of a company executive who just had to have the fastest machine in the west, just to run windows in front of his staff and clients. He was willing to pay big bucks for the privelege.

Psychologists believe, that as we achieve our basic needs and wants, other needs and wants develop to take their place. It is believed that the area where our needs can develop most, is that area where we can grow though testing or challenging ourselves. As we discover more about ourselves, we extend our boundaries, and seek greater challenges. Perhaps this is why computers are so addictive for people at all ages. Childen at an early age, are challenged to succeed in games, to achieve higher levels, to beat the previous score, or to do do it faster, or to try a more difficult task. I suspect this might apply to some adults I know also. Like young children, adolescents become addicted to computer game playing. While many only play games, others are challenged to extend their mastery over the computer by writing programs, or by communicating with othe users on a BBS, or even hacking into forbidden areas. I watched a TV program recently, which showed the fantastic lengths, one teenager went to, in using his computer to break into some of the largest systems in the USA. The greater the difficulty, the harder he tried to beat it, to the point where he faces years in jail when the authorities find him.

Perhaps adults find computers challenging, because of the range of new experiences available, and the mastery of new skills. I spoke to an acquaintance this week, a graphic artist of some fourteen years, who told me that in her industry, people now spend most of their time on the computer, being creative in a totally different way. She doesn't have to draw all day. She now can manipulate colours, shapes and designs on screen. She can edit, store or change all her creations electronically. I gained the impression that the new opportunities have opened up a whole new series of challenges for creative artists, in ways only dreamed of in the past.

In my own small way, I have been challenged by my computer. It has opened up my world much further than I ever expected. It has introduced me to some understanding of electricity, electronics, communication etc. I have learned which end of a soldering iron to use, and why some wires are red or black. There are times now when I am tempted to look at other electrical appliances which break down. (I only said tempted). Members can relax. I have been challenged trying to write programs which get the computer to do what I want. Unfortunately, this challenge has taught me that I don't really have the talent to go much further. However, I am being challenged more and more to apply computer use in my work, to be more productive, and simply to do things better. There is little doubt in my mind that perhaps the major reason why so many people respond so positively to computers, is that almost everyone can be challenged in some way. There are levels of achievement and challenges for everyone who tries. Computers can now be used in almost every aspect of our lives, whether we are talking about a spastic child with no muscle control, or a young child learning to count, or at the other extreme, a banker working with millions of dollars, operating on the international money market.

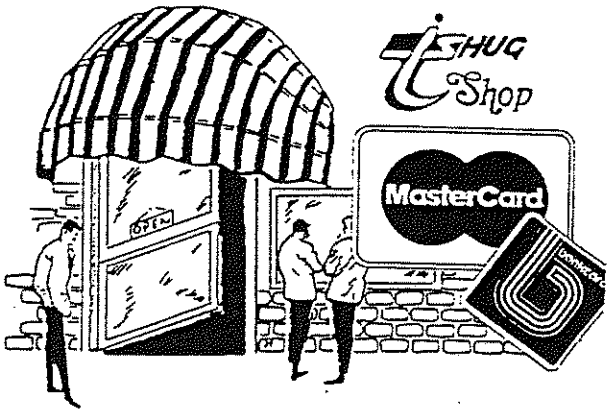
To come back to my original question. How do you use your computer? In what ways are you challenged? Some unkind people say I am a mentally challenged person. Certainly I am challenged by a computer, even by my trusty TI. I still have so much I can learn to do with it. Perhaps the bottom line is that with computers there is no limit to the challenge and achievement. We are limited only by constraints of intellect, creativity, anxiety and time.

Have you noticed how some people, usually older adults, are terrified of computers and associated technology. Anxiety, fear of the unknown and ignorance, rob them of them of the opportunity to be challenged, and to find the incredible enjoyment, and sense of achievement which can be had using a computer. I might add that they also miss the frustrations.

Do you want to be challenged mentally, to have your horizons extended, and find new areas of achievement? If you do, then come to the next meeting.

See you there,

Dick Warburton



TIsHUG SHOP.

with Percy Harrison.

Many thanks to Ross Mudie who has explained that the diode that was inserted in the keyboard mentioned in my article last month, was placed there to obviate the necessity to release the Alpha Lock key when using joysticks. Unfortunately the diode was inserted in the wrong place, it should have been placed in the Alpha Lock part of the circuit not in the Function Key circuit. It would appear that whoever inserted the diode selected the wrong wire on the ribbon cable connecting the keyboard to the motherboard and hence the problem with the Function Key. The morale of this story is that if you don't know what you are doing don't do it leave it to the experts in your club.

The April meeting saw a further reduction in our shop stocks with the sale of the Stand Alone Modems, Stand Alone Drive and six non-working Drives.

The non-working drives were purchased for $1.00 each by one of our younger members who is doing a project at University involving Robotics and he is using the Step Motors from the drives in the project. After pricing step motors from a retailer and being told that they would cost over $40.00 each he reckons that he has made a good buy through the club. Well Malcolm, we are very pleased that we were able to help you out with the drives and still have a few more if you need another lot.

While I think of it, I had a letter from one of our members in WA who purchased a Music Kit from us at the TI Faire back in 1992 asking if we could supply any disks of programmed music for the Music Card. I wrote to the States and was most surprised and disappointed to be told that there was nothing available from that side of the world. As we sold a number of these kits to some of our members I thought that some of you may have been musically gifted and written your own music programs to run on the Card in which case you may like to send a copy of your work to me so that it can be distributed as club software to other members of our club who invested in the Kit. How about it, all of you music Maestros, if you haven't already produced some music programs for the Music Card now is the time to put your talents to work and produce some music that can be shared by your fellow members. Any programs received will be appropriately acknowledged and made available to other members at our nominal software cost.

At the April meeting a group of our members started to assemble an IBM compatible computer for the club. The only real snag they struck, I believe, was that the hard disk would not work so we will have to acquire another one and hopefully will get it up and running at our next meeting. While on the subject of IBM compatibles, I note that the sales of IBM compatible programs through the shop is almost non- existent. In fact only two of our members are buying these disks. With almost half of our members now owning IBM compatible computers this is a most disappointing result and certainly will not help to keep our club in operation unless there is some worthwhile income generated from this side of our club. It is certainly not our intention to have the TI members subsidise the IBM newcomers so what about growing long arms and short pockets so that you can reach that elusive commodity called money and spend a little each month on software and blank disks from your club.

We have now decided that we will offer a cartridge refill service for Inkjet Printers commencing immediately. The price for a refill will be $16.00 so bring along your empty cartridges and have them rejuvenated, it's much cheaper than buying new cartridges and saves you the trouble of storing your own refilling equipment and getting ink all over your furniture while trying to fill them at home. Keep mum happy, let someone else do the refilling for you, that's what your club is for.

PRICE LIST.

5.25 in. DSDD Disks (Box of ten)	$6
5.25 in. HD Disks (Box of ten)	$9
3.5 in. DSDD Disks (Box of ten)	$9
3.5 in. HD Disks Formatted (Box of 10)	$12
5.25 in. DSDD Half Height Drive (New)	$35
12 Volt AC Transformer	$4
13 Volt Arlec Transformer	$12
8.5, 17 Volt Transformer	$25
60 VA Transformer	$20
MFC Printed Circuit Board	$30
MFC Kit (Disk Controller)	$103
Music Kit with PCB	$65
32K Memory PC Board	$7
Horizon Ram PC Board	$45
Horizon Ramdisk Basic Kit	$35
Funnelweb Eprom Set (3 Eproms)	$36
32K Static Ram IC (62256)	$10
8K Static Ram IC (6264LP)	$5
Exchange Console	$30
ROS Version 8.14	$12
Mini Master 99	$70
Mini PE RS232/PIO PC Board	$30
Modulator UHF or VHF	$15
TI Power Supply	$25
TI 32K Memory Card	$40
Modem PE Card (300 Bd)	$20
PE Ramdisk (184k Eprom)	$120
PE Ramdisk (248k Eprom)	$160
PE Ramdisk (320k Horizon)	$200
Printer (Serial)	$120

Postage extra on all items.

Bye for now.

MAY SOFTWARE

By Larry Saunders

Diskname P073
Total Sectors 358 Free Sectors 18
Date MAY 1994 Files 6

Page Pro Pictures.

P073

ALARMCLOCK	55 I 13	BAND	48 I 13
BOOKMARK	96 I 13	BOOKS	84 I 13
BUTTERFLY	30 I 13	COOK1	27 I 13

Diskname P074
Total Sectors 358 Free Sectors 24
Date MAY 1994 Files 6

Page Pro Pictures.

P074

COOK2	30 I 13	CRICKET	31 I 13
DRAGON	52 I 13	FEATHER1	53 I 13
FEATHER2	52 I 13	HEAD	116 I 13

Diskname P075
Total Sectors 358 Free Sectors 34
Date MAY 1994 Files 5

Page Pro Pictures.

P075

HOURGLASS	59 I 13	ICECREAM	93 I 13
KEYS	70 I 13	MONEY	65 I 1
MONEYBAG	37 I 13		

Diskname AV076
Total Sectors 358 Free Sectors 4
Date MAY 1994 Files 8

Adventure games.
All run from Extended Basic.

AV076

PS*ADV	39*Prog	PYRAMID	44*Prog
QUEST/4	39*Prog	QUESTSWORD	46*Pro
RAISEQUEST	46*Prog	REMZAK!	68*i254
SHERLOCK	37*Prog	SHIPWRECK	35*Pro

TENNIS

PROGRAM FOR THE TI-99/4A

This tennis program features most of the actions of a real tennis match, including:

- serves
- forehand and backhand shots
- lobs, volleys
- balls out or in the net
- defensive or offensive play
- tie-breaker
- scores announced by the referee via the speech synthetizer

And even better, you can play against a wonderful partner: the TI-99/4A COMPUTER, or against a second player.

Three different levels allow you to select a perfect partner, really adapted to your training and skill, from beginner to pro.

Furthermore, a live demonstration game between two computer players will show you how realistic the action is, and perfectly illustrate all the capabilities of this program.

USER'S INSTRUCTIONS

I- LOADING THE PROGRAM

Required configuration:
-TI-99/4A
-Peripheral expansion system with disk-drive and memory expansion
-Joysticks
-Speech synthetizer (optionnal)
-Editor/assembler cartridge

Select the Editor/Assembler option 3 (Load and run)
-File Name: DSK1.TENNIS
-Program Name: TENNIS

The program is now ready for use.

II- SELECTING GAME OPTIONS

The introduction screen appears, announcing the program. After a few seconds, a demonstration game starts automatically, showing live action.

Press BACK then any key to get the option selection screen.

```
--------------------------------------------------
                    TENNIS
              OPTION SELECTION

                     1         2
                  PLAYER    PLAYERS     DEMO

    NOVICE         --O
    AMATEUR
    PRO
```

Select the level and the number of players (or a demonstration game) by moving the small racket shown in the chart by using the joystick or the arrow keys (S,D,E,X).

Press ENTER or FIRE once your selection is made.

You are then prompted for the name of the players. you can also give a name to the computer champion. If you do not enter a name, the computer will just assign a standard one to allow distinguishing the two players on the score board. Note that a colored player indicates the color of the player to which the name is assigned.

Then the following message appears at the bottom of the screen:
"REMOVE THE ALPHA-LOCK THEN PRESS ENTER"

You are now ready to start your tennis match.

III- PLAYING A TENNIS MATCH

Move the players with the joysticks. Press the FIRE button to swing the racket in order to hit the ball.

You can position the player to receive the ball either in forehand or in backhand.

When you press the FIRE button, the racket starts moving. the direction of the shot is determined by the relative position ball/racket when the coincidence is detected.

SERVING:

When it is your turn to serve, use your joystick to give the direction of the ball, relatively to the serve area (left, center or right) but also the strength of your serve (up or down for fast serve, center position for normal serve). Then press FIRE while keeping the

joystick in the selected position. If your first serve is out, you are naturally given a second chance. the probability of success is related to the direction and strength you selected as in a real tennis game.

POSITIONING THE PLAYER TO RETURN THE BALL:

Moving your player to the right (resp. left) results automatically in a positioning of his racket for a forehand shot (resp. backhand shot). However, in order to allow a fine positioning of the player, this one can move a few steps left or right before the racket gets actually positioned. In any case, hitting the FIRE button results in moving the racket from backhand to forehand and vice versa.

RETURNING THE BALL:

The ball speed control can be achieved by the player motion when this one hits the ball.

- if the player moves towards the net, the ball will be accelerated.
- if the player moves backward, the shot will be a lob if the opponent is close to the net.
- if the player does not move vertically, the ball will be hit at normal speed.

The ball direction is also affected.

SCORING:

All the TENNIS rules are respected. the players change side after every odd game. The referee announces the score.

The match takes place in five sets.A TIE-BREAKER game takes place when necessary.

LEVELS:

The three levels are characterized by the pace of the action and by the increasing aggressiveness of the computer champion.

At NOVICE level, the computer champion returns the ball in your direction and is not aggressive. At PRO level, the champion becomes merciless: he alternates fast and normal shots, executes lobs and volleys. He won't let you breathe a second. At AMATEUR level, the computer champion plays at an intermediate level, but be careful: sometime he will play as a real pro.

At the end of a match, the level and the number of players is displayed allowing to testify the level of a performance.

SPECIAL OPTIONS:

- PAUSE: pressing the SPACE BAR results in stopping temporarily the action. Press any other key to resume the game.

- SPEED: the keys + and - allow to increase or decrease by step the pace of the game.

- COLOR: the color of the court can be changed by pressing the function key followed by 1, 2 or 3. This simulates various kind of tennis courts (grass, clay or decoturf).

- REDO: the key sequence FUNCTION REDO allows to restart a match from beginning.

- BACK: the key sequence FUNCTION BACK allows to go back to the options selection screen.

This file was supplied by GAMES for TEXPAC BBS.

END OF ARTICLE

To The Editor
TIsHUG

From Serge Rebikov

I felt that it is about time that I joined with so many others in praising our little TI99/4A computer. Over the years (someting like 10) on the same machine I have been able to solve quite a few problems,having started with the basic console purchased from Big W for $179 - the system is more sophisticated now. Often the problems were for my work as an engineer, that would have been unique to me.

The TI also started my two daughters to become computer literate when going through their professional courses. They have now moved to greater things, whereas I am now retired and still find the TI useful. For example I have found the calculation of the period between two dates necessary (the "pre" and "post" period for rollovers). Not having a readymade program it was not difficult to write one and to test it. I am submitting it for publi- cation since it could be useful to somebody with a similar problem.

Not having written a program for some time it was good to have a challenge and feel that the rust was coming off my brain! There is always a sense of achievement when the program finally does all the things that you have set out to get.

Regards

```
100 REM "DATES" PROGRAM.THIS PROGRAM CALCULATES THE
PERIOD IN DAYS BETWEEN TWO CONSEC.DATES.MAKES ALLOWANCE
FOR LEAP YEARS.
110 REM PROGRAM BY SERGE REBIKOV,MEMBER OF TISHUG.
120 REM DATE: 14.3.1994
130 REM NO OF BYTES USED 1257
140 CALL CLEAR
150 DIM A1(12),B1(12):: B1(0)=365
160 FOR X=1 TO 12
170 READ A1(X),B1(X)
180 NEXT X
190   DATA    31,334,28,306,31,275,30,    245,31,214,30,
184,31,153,31,122, 30,92,31,61,30,31,31,0
200 DIM A2(12),B2(12):: B2(0)=366
210 FOR X=1 TO 12
220 A2(X)=A1(X):: B2(X)=B1(X)
230 NEXT X
240 A2(2)=29 :: B2(1)=335
250 PRINT :: PRINT
260 PRINT "DATA ENTRY : DAY,MONTH,YEAR"
270 PRINT
280 INPUT "EARLIER DATE : ":DD1,MM1,YY1
290 INPUT "LATER DATE : ":DD2,MM2,YY2
300 IF YY1/4=INT(YY1/4)THEN 310 ELSE 330
310 DAYS1=A2(MM1)-DD1+B2(MM1)
320 GOTO 340
330 DAYS1=A1(MM1)-DD1+B1(MM1)
340 IF YY2/4=INT(YY2/4)THEN 350 ELSE 380
350 DAYS2=DD2+B2(0)-B2(MM2-1)
360 DAYS0=B2(0)-DAYS2
370 GOTO 400
380 DAYS2=DD2+B1(0)-B1(MM2-1)
390 DAYS0=B1(0)-DAYS2
400 P=YY2-YY1
410 IF P=0 THEN 420 ELSE 440
420 DAYS=DAYS1-DAYS0
430 GOTO 510
440 DAYS=DAYS1+DAYS2+(P-1)5
450 IF P=1 THEN GOTO 510
460 FOR Y=1 TO P-1
470 YR=YY1+Y
480 IF YR/4=INT(YR/4)THEN 490 ELSE 500
490 DAYS=DAYS+1
500 NEXT Y
510 PRINT :: PRINT
520  PRINT  "INTERVAL  IN  DAYS  BETWEEN  GIVEN DATES IS
";DAYS
530 END
```

END OF ARTICLE

Myarc Hard and Floppy Disk Controller Card Bug

by Geoff Trott

I identified a bug in the HFDCC DSR in its handling of fractured files on floppy disks some time ago. I tried to find the problem by disassembling the DSR ROMs and examining the code but this was always going to be a long job. In the meantime, Rolf and I sent some money to Beery Miller to help with his buy out of the software from Myarc and asked him if there were the sources for the HFDCC. In due time he sent three disks, one of which contained the desired sources. After a bit of work, I managed to assemble a version which agreed with the contents of my EPROM (version 12) and then started to look for likely spots for the bug, with the benefit of names and comments.

I found the problem rather quickly in the first page of the DSR. The way Myarc did the HFDCC, was to have addresses >4000 to >4FBF for the DSR (just less than 4 Kbytes), >4FC0 to >4FFF for memory mapped I/O, and >5000 to >5FFF for paged RAM in the DSR address space. The DSR code is 16 Kbytes large and so is made up of four 4 Kbyte pages and resides in a 27128 EPROM. In the first page, there are routines for deleting files. As part of these, the routines must read the file control block (FCB) and identify the sectors that are in use by the file. If the file is fragmented, they must pick out the correct sectors to be freed in the bitmap in sector 0 (the volume information block). When a file is re-written, as in the editor, it must do this by first deleting the file and then writing the file to the sectors which are freed up in the bitmap by the delete process. If the wrong sectors are freed in the delete process, real problems arise with fragmented files, as data is written into sectors which are still in use by other files (the reason files are fragmented in the first place is because there are other files in the way). This is where the problem was. In fact there were two problems and its behaviour was far worse than I had originally thought. The fix was quite easy and I will show you with the actual code extract below. (This is located in a file called WIND3/S on the disk I received, near the end of the file.)

```
ATTOP  MOV  @FNLOCF(R3),@STRTSR  save ptr to first FCB
ATTOP1
* This routine deletes or frees all sectors in the VIB
       MOV  R9,R6              get ptr in R6
       ABS  @WDSFIL
       JEQ  ATTOP2

* now just delete all the clusters in the fdr
       BLWP @GETVIB
       JEQ  FRSER0             error
       AI   R6,CLUSTR        R6 contains address of byte
       CLR  R10                R10 is the size to date
       LI   R5,76              76 clusters maximum
FNTPR1 MOVB *R6+,R1            get first two nibbles
       SWPB R1
       MOVB *R6+,R1            put third nibble in front
       MOVB R1,R2             save first nibble of number
       ANDI R1,>0FFF           get AU number
       JEQ  DONCLS             if zero, finished
       SWPB R2                get ready for next nibbles
       MOVB *R6+,R2            get other two nibbles
       SRL  R2,4               get number of AUs
FNTPR2 S    R10,R2            get number in this cluster
*      MOV  R2,R10             first error, change to add
****** changed by Geoff Trott to solve problem with
****** fractured files November 1993. Previous line
****** commented out and the next two added in its place
       A    R2,R10             keep running count
       INC  R10                adjust for base 0
****** This adds 2 bytes to length of program.
       BLWP @FRBITM
       DEC  R5
       JNE  FNTPR1
```

The first bit of code sets everything up by reading the FCB into memory and setting R6 to be the address to the first byte of the cluster information, R5 as a counter for the maximum number of clusters, and R10 to sero. Then the loop is entered whose top is labelled FNTPR1. At the end of the code segment is the decrement of the counter and jump back around the loop if non-zero. The other exit from the loop occurs if the first sector in the cluster is zero. The first piece of code in the loop extracts the two 12 bit numbers from the three 8 bit bytes which contain the cluster information. R1 contains the first sector number of this fragment and R2 contains the size number, which is the size of the file to date, base 0 (i.e. one less than the number of sectors in the file in all the fragments looked at so far). For example, the following sets of numbers could be for a file which has three fragments, each of 3 sectors long.

>022, >002 (first sector >22 and 3 sectors long)
>028, >005 (first sector >28 and 6 sectors in all)
>02E, >008 (first sector >2E and 9 sectors in all)

There was never a problem with getting the numbers from the FCB, nor with the first sector number. The problems arose with the number of sectors to be deleted from the size numbers. The bit map routine (FRBITM) expects to get the first sector number in R1 and the number of sectors to be deleted in R2 in base 0 form. For the first fragment of the example, we need to pass the two numbers shown. For the next fragment, we need to pass the first sector and then the difference between the new size and the first size, minus 1 (>028 and >002). And for the third fragment the number of sectors is once again found from subtracting the last two size numbers, minus 1. As the routine was, the first fragment was fine, the second fragment would be one more than it should be, and the third and subsequent fragments would be much too large as the size of the last fragment was subtracted from the size of the total file.

The fix was to add the new size to R10 (A R2,R10 not just MOV R2,R10) to keep the contents of R10 equal to the current size of the file, and to add one to R10 to make it the actual size of the file or the number of sectors deleted to date. With these two changes, all these problems were resolved. This caused the code to increase by two bytes at this point, shuffling the code from there to the end of the code down by two bytes. This had to be done in an EPROM, which meant that a new EPROM must be burned. However, three pages from the EPROM were not to be changed, only page zero needed to be done.

To do this I wrote a program in C, modelling it on the one done for the EPROM RAMdisk. This program took the output of the assembler in an uncompressed display fixed 80 file and sent it out to the EPROM programmer in Motorola S1 format. The idea is to first read in the contents of an EPROM which needs to be changed into the programmer's memory, then load into the programmer's memory the page (4 Kbyte) which is to be changed and then contents of all the memory of the programmer is programmed into a blank EPROM. Along the way I also changed by hand the default interlace for floppy disks from 4 to 7 so that if an interlace is not given, 7 is used. This means that other disk manager programs like Disk Utilities, DiskReview, DM1000 and other formatting programs which do not allow the interlace to be entered, will format floppy disks with an interlace of 7. This makes much more sense for double density disks, in particular when doing a backup with DM5. It is a bit slow for single density disks, but they are smaller in size anyway. This was changed by changing a byte in the programmer's memory at >2088 from >04 to >07. It is interesting that page 1 uses addresses >2000 to >2FFF in the EPROM while page 2 uses addresses >1000 to >1FFF. A mixup between software and hardware, I guess.

When programmed and put in the card, the new DSR worked as well as I wanted, handling fractured files correctly and formatting with the desired interlace. I realise that not many of our members have Myarc Hard and Floppy Disk Controller Cards but I hope they have found something of interest in the explanation. For those that do have HFDCCs and would like a copy of the new EPROM (version 13), if they can get a blank EPROM (27C128 or 27C256) to me at the next meeting or by mail I will be able to program it for them. I will be sending this to Beery Miller for him to distribute for others and the same conditions will hold for people overseas, except I will expect something to cover the cost of the postage.

END OF ARTICLE

Bits and Bites

By Larry Saunders
March 1994

Subject: CLIPIX

Introduction:

As the number of sources of available clipart increase, so do the size of the pictures. While much of the clipart available for Page Pro doesn't exceed it's size limitations (60 columns x 66 rows), much of what is coming from outside sources (PCX and MAC file extensions) regularly exceeds these limits. If the clipart should contain multiple images within the file, only those within the allowable limits of Page Pro (60 cols x 66 rows) can be clipped without reducing the original image. My only problem with doing this is that many of the images tend to lose definition when they get reduced. My alternative to you is to provide you with a program which will allow you to clip any of these oversized Page Pro pictures with the use of the arrow keys and the Q key.

Loading Procedures:

This program is loaded via E/A option #5. It will not load via Extended BASIC. To load it do the following:

* Place the E/A cartridge in the cartridge port and turn on the computer and peripherals
* Place the program disk in drive one and select Editor/ Assembler from the Master menu.
* Select Option #5 - Load and Run * Type DSK1.CLIPIX and press <ENTER>
* The program will load and run Please note that due to an unusual error, the program will not display a title screen if loaded via an HFDC.

Additionally, compatibility with Horizon RAM-disks is not assured.

Program Operation:

Once the program loads, you'll see a split screen (reminscent of Page Pro) and will be asked to enter an 'Infile:'.

The program is looking for you to specify a Page Pro picture file (device and filename). If the file is found, the complete size (in rows and columns) of the picture will appear to the right of the filename and the upper left portion of the file will appear in the viewing window.

If the file isn't found, the message "File Error" will appear on what you would consider to be line 24.

After the image (or as much as possible) is displayed, the cursor (light green/dark green) will appear in the upper left corner of the screen. To aide you in moving around the screen, the cursor position (row x col) is posted in the scoreboard area. Pressing any of the cursor keys (E/X/S/D) by themselves will move the cursor in that direction by one space. Since this could get rather tiring moving from one side to the other, I've provided alternate cursor keys (E/X/S/D plus the either the function key or the control key), to move you in increments of four (4) or eight (8) spaces. If you decide that you would just like to home the cursor to the upper left portion of the picture (row 1 column 1), just press the H key; the screen will be repainted if necessary. Eventually, all this cursor movement will take the cursor beyond the viewing screen. When this happens, the screen will be moved and ONLY the new screen data will be read and displayed. Since it's not feasible to keep all of the picture within memory, this is the next quickest way to refresh the screen without repainting the entire screen.

To "clip" the picture, I require you to frame the picture area by identifying the upper left and lower right portions of the screen. To start the clip function, put the cursor on the upper left area and ress the Q key. To end the clip, put the cursor on the lower right position and again press the Q key. If pressing the Q key doesn't seem to do anything the second time, you most likely chose a row or col that is less than the row or col of the start clip. Once a valid range is provided, you'll be prompted for an output filename. If you supply a valid name, a new PP picture file will be created. If you decide that you really don't want to create an output file, just press ENTER.

After the file output is complete, you'll be asked if you wish to continue clipping the same file. This program's primary objective is to allow you to clip multiple images from one file, so why waste time having you select the same file over and over again. If you respond "Y" to the continue message, you'll be returned to the current picture file at the exact spot where the cursor was when the clip function was completed. Continue moving the cursor and using q key to clip new files. If you respond "N" to the continue message, the screen will be wiped clean and you'll be asked for a new input file.

This program comes equipped with an ALTERNATE CLIPIX MENU which can be reached in two different ways. The first way is by entering a null string at the Infile: prompt. The second way is by pressing the Fctn Q key while moving the cursor around a Page Pro picture. When the alternate menu is selected, the mode of the screen changes from bitmap to text, and the current infile (if opened) gets closed.

The ALTERNATE CLIPIX MENU gives you four (4) options. Option 1, permits you to load another program image file. The loader for CLIPIX has been placed between >B33A and >B662, to allow you to load most related Page Pro programs (PIXPRO, PAGE PRO, GOFER, FW, etc.).

The second option will quit to the Title Screen of the cartridge. The third option is provided for those times you say "oops". The last option will let you catalog any disk between 1 and 9.

Note: During the use of this program, at times the lower right portion of the screen will be littered with what appears to be garbage. This is a product of the TI disk controller and bitmap mode having VDP memory in common. I've managed to minimize it, but not totally eliminate it. For those using a MYARC disk controller, this problem is non-existant.

END OF ARTICLE

EDITORS COMMENTS

By Loren West

I hope that everyone has had an enjoyable and safe Easter, a few extra kilometres to jog to shake off those Easter eggs.

The last computer club meeting was very well attended, Peter was building an IBM clone computer for Club use, whilst he had one of his computers up and running. While we were looking over his shoulder he showed us some of the things it could do, I could picture myself typing these articles out and listening to my favourite C.D. being played by the same machine at the same time, very impressive.

Dick was busily showing other users some of the hidden secrets of XTREE GOLD, and the "in's and out's" of the DOS system.

Percy was kept busy selling software, hardware, magazines, and collecting our Annual Dues, so if you haven't paid up yet .....don't delay, pay today.

Ross had three TI. systems hooked up to each other and operating independently (I believe this system could operate four separate stations), this system also had a printer attached and any station could have their information printed. Ross had developed his own software for this purpose and as he points out, the cost to have a similar system running in IBM would be far greater than his system.

I would like to ask readers for some articles for this magazine, also any programs that you have developed over the years that you might think others could find interesting. What a chance to share some of your enthusiasm about our TI. with other fellow members.

TREASURER'S REPORT

by Cyril Bohlsen

Income for previous month ..... $ 1955.00

Expenditure for previous month .. $ 1455.89

Profit for previous month ..... $ 499.11

Membership accounted for $1580.00 of Income.

Shop sales ..... $ 375.00 of Income.

The expenditure was made up of the following

Printing & Postage of TND ..... $ 299.44

Purchase of IBM Computer Parts ... $ 810.00

Purchase of BJ Re-inking Kits .... $ 107.00

Aust. Securities Commission ..... $ 111.00

Shop Expenses ..... $ 58.65

BBS Expenses ..... $ 69.80

As you can see we have had an expensive month with the purchase of parts for our IBM Compatable Computer, the only remaining part to purchase is the Hard Disk Drive.

We welcome back into the fold,
John Vandermey of Carlton

SECURITY ON RAMdisks

Retyped by Loren West

This comes from William Berendts, president of the Ozark 99ers User Group, Springfield, Missouri. He writes:

RAMID is a simple 5 line program written to prevent unauthorized entry to your Horizon RAMdisk. Children, grandchildren and others are oftern drawn to the mystique of a computer. Ordinaily, they cannot do much harm, unless you have your system autoload your RAMdisk on power-up. Although written for use with the Horizon RAMdisk, you may be able to adapt it for use with other RAMdisks as well. Not having access to other RAMdisks prevents me from testing it on others.

The programm is named RAMID. Entering the proper ID will result in your RAMdisk, or any disk protected by RAMID, being made avaliable for use.

If your system autoloads and untrained hands - sometimes even trained hands and minds - hit the wrong keys, havoc breaks loose and you can lose everything stored in the RAMdisk's memory.

RAMID might prevent that by requiring anyone firing up the system to enter an identification code before the RAMdisk menu is loaded. An incorrect identification will result in the following message displayed on the screen: "You are not authorized entry to this computer." It is displayed as an endless loop.

For safety's sake, keep an unprotected copy of RAMID on a floppy, stored in your archive file. load the program into memory and enter your personal identification. This can be anything - a name, pet name, social security number, address, and so on - into line 1, replacing the statement "your code." If you wish to give another user access to your system under his own personal identification, enter that user's personal code into line 1, replacing the statement "second code." If only one code is used, then delete the second half of line 1 (AC2$="SECOND CODE") and in line 2, delete OR Q$=AC2$. If you you intend to use RAMID to protect a floppy disk, enter in line 2 RUN "DISK*.LOAD - FILENAME" in place of DELETE "MENU". If you are protecting your RAMdisk, do not replace DELETE "MENU".

After you have saved the personalized program back to your archive floppy, test it by typing RUN and pressing Enter. Enter an incorrect ID. Your RAM menu (or disk) should not load and the access denied message should flash on the screen, along with a sound to alert the system operator. If you attempt to stop the program by pressing FCTN 4 the loop will continue. The only to stop the program is to turn off the console, or pressing FCTN +.

So far so good. Now load the RAMID program into memory again. This time enter your correct ID. This time your RAMdisk menu should come to the screen, or the disk drive you wish to access will start, depending on the entry made in line 2.

If you wish to RAMID to prevent access to your Horizon RAMdisk and every-thing has worked properly, load your RAMdisk Operating System into memory and reconfigure your RAMdisk by editing the ROS in the following manner: First, move the file ID "4 MENU" to the next empty "U" position. Then enter "4 RAMID" into the position formerly occupied by "4 MENU". Save your altered ROS to another disk or under another filename to prevent changing your original ROS.

After changing your ROS, load your personalized RAMID program into memory and save it to your RAMdisk as a protected file by typing the following: SAVE DSK*.RAMID, PROTECTED

The program can be loaded and run, however, it cannot be listed. This prevents an unauthorised user from discovering the personal ID in line 1.

When all is loaded, run your system as you normally would. If your system loads on power-up, the line requesting your ID should be displayed and you need only to follow through as discribed above.

If you decide to change your ID, simply load the unprotected RAMID program from your archive disk, change the ID in line 1 and then save it to your RAMdisk as a protected file, using the filename RAMID.

If you wish to protect a specific disk, save RAMID to the disk under the name "LOAD". Obviously, any other load program on the disk would have to be renamed to avoid overwriting. I have found that just changing the name of the original load program by adding a number to it is easy - LOAD2, for example. Make sure line 2 of RAMID reflects the changed name, i.e. "RUN DSK*.LOAD2".

There is one drawback to the use of RAMID - if any program being used is written in other than Extended BASIC, the request for an ID will be displayed when you exit the program. If the programs are written in XBASIC, it might be necessary to make the appropriate changes to display the main menu without starting over. Generally, using DELETE "MENU" in place of END will get you back to the RAMdisk menu without going through RAMID .If using a disk replace END with "RUN DSK*.LOAD2", or whatever filename you have given the original load program. Programs written in other languages will be more problem-atical, depending on the users expertise in those languages.

```
    100    AC1$="123"    ::
AC2$="456"
    110 DISPLAY AT(8,4)ERASE
ALL:"ENTER YOUR ACCESS CODE:"
:: ON BREAK NEXT :: ACCEP T
AT(12,5):Q$  :: ON BREAK NEXT
:: IF Q$=AC1$ OR Q$=AC2$ THEN
RUN "DSK1.LOAD1"
    120 ON BREAK NEXT ::
CALL
SOUND(500,110,0,130,0,196,0)::
ON BREAK NEXT :: CALL C LEAR
    130 ON BREAK NEXT :: FOR
X=1 TO 8 :: ON BREAK NEXT ::
DISPLAY AT(10,4):"YOU ARE NOT
AUTHORISED":      :TAB(3);"TO
ACCESS THIS COMPUTER!" ::  ON
BREAK NEXT :: NEXT X
    140  CALL SCREEN(2):: ON
BREAK NEXT :: GOTO 140
```

EDITORS NOTE.

I have tried this program on a floppy disk and works quite well.

Next month, if room permits on these pages I will print a program that I wrote for my children, it is also security coded, not quit as suficated but more interisting for the children when they put in the wrong code.

END OF ARTICLE

CHRISTMAS LABELS

Retyped by Loren West

This comes from Jim Lesher, of the Dallas TI User Group. It is onother in a series of items that use the Missing Link.

This programm, lets you make your own borders on address labels. We design our characters by redefining a character set used by our printer. However, we are limited to a 7x9 matrix, which is the maximum size of each character. In this case, the character is a tiny Christmas tree. Actually, the border consists of many tiny trees, encircling a 3 1/2 x 15/16 - inch label.

You will need to adjust the labels in your printer to make them look right. So, put the labels in with the back side toward you. you can see the labels through the backing. This way you can set up the spacing without running a lot of labels. Then when you get it right, turn the labels back over and output as many as you want.

The labels look best in red, blue or green. This programm is set to produce 40 labels, but you can change the number by modifying line 120. Change 40 to whatever number you want.

Here are some numbers to make other characters, just type them in to replace the numbers in line 30.

```
BELL
30 DATA 0,56, 68, 132, 255, 132, 68, 56, 0, 0

CANE1
30 DATA 4, 2, 1, 0, 1, 0, 1, 2, 124, 0

CANE2
30 DATA 0, 12, 2, 0, 1, 0, 1, 2, 124, 0

CROSS
30 DATA 0, 4, 0, 4, 127, 4, 0, 4, 0, 0

TREE2
30 DATA 16, 8, 4, 2, 113, 2, 4, 8, 16, 0

TREE3
30 DATA 16, 8, 20, 2, 253, 2, 20, 8, 16, 0
```

```
10 REM TREE
20 CALL CLEAR
30 DATA 16,8,20,2,113,2,20,8,16,0
40 OPEN #1:"PIO"
50 OPEN #1:CHR$(27);"";
60 PRINT #1:CHR$(27);"E";
70 PRINT #1:CHR$(27);"*";CHR$(1);CHR$(126);
80 FOR I=1 TO 09
90 READ MI
100 PRINT #1:CHR$(MI);
110 NEXT I
120 FOR N=1 TO 1
130 PRINT N
140 PRINT #1:CHR$(27);"S";CHR$(1);
150 PRINT #1:CHR$(27);CHR$(51);CHR$(12);
160 C$=RPT$(CHR$(126),35)
170 PRINT #1:C$
180 B$=RPT$(CHR$(126),01)
190 FOR X=1 TO 09
200 PRINT #1:B$;
210 PRINT #1:TAB(35);B$
220 NEXT X
230 PRINT #1:C$
240 PRINT #1:CHR$(27);CHR$(51);CHR$(12)
250 NEXT N
```

END OF ARTICLE

LEARN TO KNOW YOUR TI
LESSON 15

with Percy Harrison

In this lesson we will introduce you to shortcuts using LET, INPUT and LIST together with a brief description of the differences between commands, statements and functions.

Keep in mind that INPUT, when used without a message in fonts, prints a "?" for a prompt. When used with a message there is no prompt, and there is no space after the message unless you put one inside the quotes.

We will show you how to place characters in rows and columns on the screen. The HCHAR command puts single characters or a horizontal line of characters on the screen. Likewise, VCHAR puts a single character or a vertical line of characters on the screen. The character is described by its ASCII number. The numbers for a few punctuation characters useful for graphics are given in this lesson but ASCII numbers will be described more fully in a later lesson.

LESSON 15 SHORTCUTS AND GRAPHICS

A LET SHORTCUT

These two lines do the same thing:

```
10 LET A=41     AND    10 A=41
```

also these two:

```
10 LET B$="HI"  AND    20 B$="HI"
```

You can leave out the word LET from the LET statement! The computer knows that you mean LET whenever the line starts with a variable name followed by an "=" sign.

AN INPUT SHORTCUT

Instead of:

```
10 PRINT "ENTER YOUR NAME"
11 INPUT N$
```

You can do:

```
10 INPUT "ENTER YOUR NAME ":N$
```

Put a colon between the message "ENTER YOUR NAME" and the variables.
Examples:

```
10 INPUT "AGAIN? <Y OR N> ":A$
20 INPUT "LOCATION ": X,Y
30 INPUT "MONTH, DAY, YEAR ":M$,D,Y
```

A LIST SHORTCUT

There are five ways to use the LIST command:

LIST	lists whole program
LIST 48	lists line 48
LIST 50-75	lists all lines from 50 to 75
LIST -27	lists all lines from beginning to 27
LIST 90-	lists all lines from 90 to the end

KEY WORDS IN TI BASIC

Some key words can only be used as commands. Commands tell the computer to do something. We have learned:

```
LIST   NEW   OLD   RUN   SAVE

Right:      LIST
Wrong:      10 LIST
```

Some key words can only be used in statements which are in program lines. We have learned:

```
STEP   GOTO   IF   THEN   INPUT   FOR   NEXT

Right:      10 INPUT A
Wrong:      INPUT A
```

If you try to use these commands without line numbers the computer prints:

```
                * INCORRECT STATEMENT
or              * CAN'T DO THAT
```

Some key words are functions. We have learned:

```
INT()   RND   TAB()
```

The rest of the key words can be used either as a command or in program lines as statements. We have learned:

```
CALL XXX   LET   PRINT   REM

Right:      10 PRINT
Right:      PRINT
```

LO-RES GRAPHICS

"Lo-Res graphics" means "low resolution pictures".

It means drawing pictures using dots and lines of dots.

DRAWING DOTS

We will use some punctuation characters to draw pictures.

Each character has a number.

Here are some good ones for pictures:

	32	blank (good for erasing)
*	42	star
+	43	plus
-	45	minus
0	48	zero
O	79	letter O

Try adding more HCHAR statements to draw more dots and other punctuation marks on the screen.

Look: 30 CALL HCHAR (row,column,character number)

You can put any number from 1 to 24 in the row place.

You can put any number from 1 to 32 in the column place.

You can use any character number from 32 to 126.

Try this:

```
10 REM ONE STAR
20 CALL CLEAR
30 CALL HCHAR(3,7,42)
99 GOTO 99
```

Press FCTN CLEAR to exit the program.

Line 30 does this:

goes down 3 rows from the top
counts across 7 spaces from the left
puts character 42 there
(character 42 is the "star")

Try this:

```
10 REM CHARACTERS
20 CALL CLEAR
25 FOR C=33 TO 126
30 CALL SOUND(100,900,10)
32 X=INT(32*RND)+1
34 Y=INT(24*RND)+1
40 CALL HCHAR(Y,X,C)
50 FOR T=1 TO 200
51 NEXT T
60 NEXT C
```

Run this program and then write down what each line of the program does. By doing this you will get a better understanding of programming.

DRAWING HORIZONTAL LINES

Use CALL HCHAR to draw horizontal lines:

Try this:

```
10 REM HORIZONTAL LINES
15 CALL CLEAR
20 CALL HCHAR(5,3,43,20)
```

It is just the same as drawing a dot, but you have one more number inside the bracket.

```
2 CALL HCHAR(row,column,character,how many)
```

So:

```
20 CALL HCHAR(3,7,88,8)
```

Means:

count 3 rows down from the screen top
then go 7 across
put character 88 (the letter "X") there
then put 8 more characters across to the right.

The "H" in HCHAR means "horizontal" line.

DRAWING VERTICAL LINES

Add:

```
70 CALL VCHAR(8,4,33,7)
```

to the first program under "Drawing horizontal lines" and run the program again.

This new line means:

count 8 rows down
go 4 characters across
put the character 33 there
then put 6 more down from there.

Assignment 15:

1. Write a program which uses each of the shortcuts at least once.

2. Write a "vacation" program. It should ask how much you want to spend. Then it should tell you where you should go or what you should do.

3. Write a "crazy" program which asks your name. The program is to have three funny ways of saying that you are crazy and should randomly choose one of these and print it after your name.

4. Put a dot on your screen 11 rows from the top and 7 columns across.

5. Put a horizontal line of stars on line 3 running from column 4 to column 12.

6. Put a vertical line of dots starting at row 4, column 6 and 9 characters long.

Because of the nature of Lesson 14 there are no published answers to that lesson. If you had any problems understanding it please get in touch with me for help.

Next month we will look at the IF statement with numbers and the END statement.

Bye for now.

END OF ARTICLE

ERROR CODE LISTING.

Reprinted from the BBS

EX-BASIC ERROR CODES.

```
10 NUMERIC OVERFLOW
14 SYNTAX ERROR
16 ILLEGAL AFTER SBRTN.
19 NAME TOO LONG
20 UNRECOGNIZED CHAR
24 $/# MISMATCH
28 IMPROPERLY USED NAME
36 IMAGE ERROR
39 MEMORY FULL
40 STACK OVERFLOW
43 NEXT WITHOUT FOR
44 FOR-NEXT NESTING
47 MUST BE IN SBRTN.
48 RECURSIVE SBRTN. CALL
49 MISSING SUBEND
51 RETURN WITHOUT GOSUB
54 STRING TRUNCATED
56 SPEECH $ TOO LONG
57 BAD SUBSCRIPT
60 LINE NOT FOUND
61 BAD LINE #
62 LINE TOO LONG
67 CAN'T CONTINUE
69 COMMAND ILLEGAL IN PRGM.
70 ONLY LEGEL IN PRGM.
74 BAD ARGUMENT
78 NO PROGRAM PRESENT
79 BAD VALUE
80 NIL
81 INCORRECT ARGUMENT LIST
82 NIL
83 INPUT ERROR
84 DATA ERROR
97 PROTECTION VIOLATION
109 FILE ERROR
130 I/O ERROR
135 SBRTN NOT FOUND
```

****

EDITOR / ASSEMBLER ERROR CODES.

XB ERROR EQUATES

```
ERRNO >0200 2 NUMERIC OVERFLOW
ERRSYN >0300 3 SYNTAX ERROR
ERRIBS >0400 4 ILLEGAL AFTER SUBPROGRAM
ERRNQS >0500 5 UNMATCHED QUOTES
ERRNTL >0600 6 NAME TOO LONG
```

```
ERRSNM >0700 7 STRING NO. MISSMATCH
ERROBE >0800 8 OPTION BASE ERROR
ERRMUV >0900 9 IMPROPERLY USED NAME
ERRIM >0A00 10 IMAGE ERROR
ERRMEM >0B00 11 MEMORY FULL
ERRSO >0C00 12 STACK OVERFLOW
ERRNWF >0D00 13 NEXT WITHOUT FOR
ERRFNN >0E00 14 FOR-NEXT NESTING
ERRSNS >0F00 15 MUST BE IN SUBPROGRAM
ERRRSC >1000 16 RECURSIVE SUBPROGRAM
ERRMS >1100 17 MISSING SUBEND
ERRRWG >1200 18 RETURN WITHOUT GOSUB
ERRST >1300 19 STRING TRUNCATED
ERRRBS >1400 20 BAD SUBSCRIPT
ERRSSL >1500 21 SPEECH STRING TOO LONG
ERRLNF >1600 22 LINE NOT FOUND
ERRBLN >1700 23 BAD LINE NUMBER
ERRLTL >1800 24 LINE TOO LONG
ERRCC >1900 25 CAN'T CONTINUE
ERRCIP >1A00 26 ILLEGAL IN PROGRAM
ERROLP >1B00 27 ONLY LEGAL IN PROGRAM
ERRBA >1C00 28 BAD ARGUMENT
ERRNPP >1D00 29 NO PROGRAM PRESENT
ERRBV >1E00 30 BAD VALUE
ERRIAL >1F00 31 INCORRECT ARGUMENT LIST
ERRINP >2000 32 INPUT ERROR
ERRDAT >2100 33 DATA ERROR
ERRFE >2200 34 FILE ERROR
ERROR >2400 36 I/O ERROR
ERRSNF >2500 37 SUBPROGRAM NOT FOUND
ERRPV >2700 39 PROTECTION VIOLATION
ERRINV >2844$$44$$9= >2900 41 NUMERIC OVERFLOW
WRNST >2A00 42 STRING TRUNCATED
WRNNPP >2B00 43 NO PROGRAM PRESENT
WRNINP >2C00 44 INPUT ERROR
WRNIO >2D00 45 I/O ERROR

******************************************************

EXECUTION ERRORS

0-7 STANDARD I/O
08 MEMORY FULL
09 INCORRECT STATEMENT
0A ILLEGAL TAG
0B CHECKSUM ERROR
0C DUP. DEFINITION
0D UNRESOLVED REF.
0E INCORRECT STATEMENT
0F PROGRAM NOT FOUND
10 INCORRECT STATEMENT
11 BAD NAME
12 CAN'T CONTINUE
13 BAD VALUE
14 NUMBER TOO BIG
15 STRING/NUMBER
16 BAD ARGUMENT
17 BAD SUBSCRIPT
18 NAME CONFLICT
19 CAN'T DO THAT
1A BAD LINE NUMBER
1B FOR NEXT ERROR
1C I/O ERROR
1D FILE ERROR
1E INPUT ERROR
1F DATA ERROR
20 LINE TOO LONG
21 MEMORY FULL
22 UNKNOWN ERROR CODE

******************************************************

LOADER ERROR CODES

0-7 STANDARD I/O
8 MEMORY OVERFLOW
9 NOT USED
10 ILLEGAL TAG
11 CHECKSUM ERROR
12 UNRESOLVED REF.

******************************************************

TI BASIC ERROR CODES PERTAINING TO DISK SYSTEM

# FIRST # SECOND #
0: OPEN CAN'T FIND SPECIFIED DISK DRIVE
1: CLOSE DISK OR PROGRAM IS WRITE PROTECTED
2: INPUT BAD OPEN ATTRIBUTE
3: PRINT ILLEGAL OPERATION
4: RESTORE DISK FULL OR TOO MANY FILES OPENED
5: OLD ATTEMPT TO READ PAST EOF
6: SAVE DEVICE ERROR
7: DELETE FILE ERROR
9: EOF

******************************************************

TI WRITER ERROR CODES

0 INDICATES DISK CONTROLLER NOT ON
          OR
    DISKETTE NOT INITIALIZED
6 NO DISK IN DRIVE No.1
          OR
    DISK UPSIDE DOWN
          OR
    DRIVE IS NOT TURNED ON
7 NO DISK IN DRIVE No.1

00 ILLEGAL USE OF LoadF, PrintF, SaveF
02 NO FILE ON DISK WITH FILENAME USED
04 DISK IS FULL
06 PrintF COMMAND IN PROGRESS WAS INTERRUPTED
          OR
    DISK DOOR WAS OPENED WHEN LIGHT WAS ON
07 INVALID FILENAME (NAME TOO LONG )
          OR
       (INVALID CHARACTERS )
15 INVALID DISK DRIVE No. OR DEVICE

******************************************************

DISK MANAGER ERROR CODES

#: FIRST # SECOND #
1: OTHER RECORD NOT FOUND
2:   SEEK/STEP   CYCLIC REDUNDANCY
CODE
3: INPUT LOST DATA
4: PRINT WRITE PROTECTED
5: NIL WRITE FAULT
6: NIL NO DISK or
                NO DRIVE or
                DRIVE NOT READY
7: NIL INVALID INPUT
8: NIL NIL
: SPECIAL ERROR CODE$FOV

******************************************************

I/O ERRORS
# FIRST # SECOND #
1: OPEN DEVICE NOT FOUND
2: CLOSE WRITE PROTECTED
3: PRINT INVALID I/O COMMAND
4: RESTORE OUT OF SPACE
5: OLD EOF
6: SAVE DEVICE ERROR
7: DELETE FILE MISMATCH
                OR
DATA MISMATCH

Retyped for TEXPAC BBS by Alistair Leslie
of TIsHUG.
```

END OF ARTICLE

EMUSIC PREPROCESSOR INSTRUCTIONS

By Norm Sellers
Version 1.2
Copyright 1986

Part 1 of 2
July 3,1986

The 'FLIPPY' disk you received contains a music utility program written in Editor/Assembler as a subroutine to be called from an Extended Basic program. This subroutine allows you to simply and quickly code up music as remarks in the calling Extended Basic program, using the Extended Basic Editor capibilities (i.e. REDO). You may code, run, change, and rerun without compiling or reading a disk file. Simply save your program when all is well. Another fringe benefit of this program is you can play the same song countless different ways (change keys, chord inversions, minors, 3 parts or 4 chopped parts etc.).

The subroutine entry name is EMUSIC. It has been loaded with SYSTEX (written by Barry Moore) with his permission. I have included his files 'SYSTEX' and 'SYSTEXDOC' in case you also have a slow loading object file.

Subroutine EMUSIC attempts to give the Extended Basic programmer assembler capabilities in music automatically. All you need to do is to code the music as it appears on the music sheet.

The calling statement and arguments are described as follows:

```
CALL LINK("EMUSIC",STRT$,STP$,OPT$)
```

where:

STRT$=character string to search for to turn on music processing (max length of 6).

STP$=character string to search for to stop music processing and return to E.B. (max length of 6).

OPT$= "RUN" to scan and play music from the beginning of the program.

"CON" to continue a previous scan from where it was stopped.

Subroutine EMUSIC is designed to allow you to graphically put a picture on the screen, call EMUSIC for part of a song, return to Extended Basic to put another picture on the screen, and return to EMUSIC to CONtinue. Sprites may also be used as birds, snow etc. while the music is playing.

EMUSIC is designed to enable you to enter coded music in 1 to 4 part harmony. At all times all three sound generators are being used. If one note is requested, as in a solo, that note is resonated with the 2nd and 3rd generators by setting them to frequencies as close as possible to the original note without equalling it. This creates a beat note much like vibrato and gives notes a full, ringing beautiful sound.

When two different notes are requested, as in a duet, the 3rd generator is set to resonate with the note on the first generator. This again improves the quality of the sound produced.

When three different notes are requested, as in a trio, the three notes are played on the three generators. If the notes on the 1st and 2nd generators happen to be the same, in a duet or trio (or the 2nd and 3rd generators hit the same note in a trio), the generator with the smallest number in the equal pair is changed to be a resonating generator automatically.

The notes are communicated to the subroutine numerically by numbering notes, starting at LOW 'A' and counting upward all black and white notes. For example, on Page 197, Appendix D in the TI Extended Basic Manual, 110='A' is represented as '1', 117='A#','Bb' is represented as '2', etc. counting upward. This table may also be found in the Editor/Assembler Manual, pages 318 through 320. The last note on Page 320 is 'F', 4 octives above middle 'C' which is the 69th note above the LOW 'A' starting the table. Subroutine EMUSIC accepts notes 1 through 69 as music notes and accepts note 70 as silent as in a musical rest (being set to about 40,000 hertz which is above human hearing).

Many musical needs go below LOW 'A' so subroutine EMUSIC uses the periodic noise generator and the third sound generator to generate notes down to LOW LOW 'A'. These notes are refered to as 'LOW NOTES' and have a slightly different sound quality but there is no way I know of to avoid this difference. LOW NOTES start with 'G#','Ab' immediately under LOW 'A' and count down into the negative numbers. LOW 'G#','Ab' is note '0', 'G' under this is '-1' etc. for one octive, to LOW LOW 'A' being note '-11'.

LOW NOTES of course cannot be resonated as mentioned before since there is only one noise generator. Therefore, if a single LOW NOTE is to be played as in a solo, the LOW NOTE is double resonated one octive higher with diminished volume automatically with the other two generators.

Another limitation to LOW NOTES is only one LOW NOTE may be sounded at a time. If two LOW NOTES are attempted to be played at once, an incorrect high pitched note will be heard instead. To avoid this happening, simply turn off one of these low notes using the volume 'V' statement to be discussed later.

The Volume 'V' statement adds versatility to the music you hear by allowing you at any time to change the volume on any musical part, and even turn off any musical part or parts. For example, a song coded in 4 part harmony may be analyzed by playing one part only, or by playing one part with any other part one at a time, or by playing one part louder than all other parts.

EMUSIC Language Description

Musical information is placed in Extended Basic Remarks from which subroutine EMUSIC reads. The format of the statements is as follows:

1. A statement consists of a single letter surrounded by at least one blank before and after. This letter identifies the type of statement, ie 'V' volume, 'N' note, 'T' time etc.

2. Arguments for the statement follow the 'Type of Statement' letter.

3. Each of the 1 to 4 musical parts is identified to subroutine EMUSIC by the 'V' volume, 'N' note, 'K' key and 'F' fade statements. Each of these statements have 1 to 4 numeric arguments. The numbers are separated by at least one blank. The first part on these statements should be the highest part and go down the chord so the lowest note is last on the 'V' and 'N' statements, ie.

```
! N 35 32 15 11
```

4. Multiple statements may be used on one remark, ie.

```
! V 3 3 7 3 N 35 32 15 11
```

In this example the four parts are: tenor, lead, baretone and base with the baretone volume turned off (Note: Baretone also could have been turned off if 15 had been 70).

A complete list of statement type descriptions follows:

```
! Start
```

"Start" is any character string to start sound processing (max length of 6 characters).

```
! B b1 b2 (default is 'B 30 100')
```

b1 is the time in 60ths of a second of the smallest note in the music which is refered to here as 'a Beat'. Note EMUSIC cannot handle fractional beats.

b2 is the number of times per beat that a wait loop is executed to ALMOST finish processing the current sound.

```
! K k1 k2 k3 k4 (default is 'K 0 0 0 0')
```

k1 thru k4 are the key changes (+ or -) of each of the four parts independently (can be used for chord inversions, i.e. raise the base only by 2 octives). Note: write minus numbers like '-2'. The argument adds to the current key of each part. If an argument is zero, the corresponding part returns to the key that the original music was written in.

```
! M m
```

m is the number of the note of the key signature when not zero. This as a non zero changes the music to the corresponding minor key. If m is zero, music is not converted to minor (or minor conversions are turned off).

! D d or ! D ON or ! D OFF
d may be "ON", "OFF" or a 2 character alpha-numeric string to display at the upper left corner of the screen to indicate which line of music is processing or which measure. The "ON" and "OFF" turn on and turn off this display.
! V v1 v2 v3 v4 (default 'V 7 7 7 7')
v1,v2,v3,v4 are the volumes of the corresponding 1 to 4 musical parts from 0 thru 7, 0 being loudest and 7 being silent (a part is silent). If less than 4 parts are being used throught out the song, only the first n volumes of the first n parts need be entered. Note: a "V" statement must be processed before any sound can be heard since the default of all unentered volumes is 7, silent, however, once entered, the volumes remain in affect until the next "V" statement is processed.
! T t
t is the integer number of beats to play the following notes until the next "T" statement is processed. Here a beat is considered to be the time that the smallest note in the music will play (since fractional beats are not allowed).
! N n1 n2 n3 n4 (default 'N 70 70 70 70')
n1 n2 n3 n4 are the note numbers of the 1 to 4 notes to play in a chord at the same time starting with the highest and going down the chord to the lowest note. These note numbers must range between -11 and 70 (70 being a silent rest). When less than 4 parts are used in a song, the unused parts need not be entered since the default of the unentered notes is 70 silent (unless they are place keepers).
! S s
s is 1 to 9 line number assigned by you to save a point in the music that you can goto with a "G" statement (used for first and second endings etc.)
! G g1 g2
g1 is 1 to 9, the line number assigned by you in an 'S' statement to 'G' Goto g2 times. When the "G" statement count has been satisfied, (ie. if you say G 1 2 and you have gone to 1 twice, the third time the statement is scanned, it is ignored, however the next time it is executed, the Goto loop is again initiated as before. Note: Forward scans are permitted, ie. if you say 'G 5 1' and have not previously saved line 5, EMUSIC scans forward for line 5, and goes to it if found. If not found in a forward scan, the music ends returning to E.B. Note: Please refer to file ENDTAG which illustrates the full power of the 'S' and 'G' statements.
! C c1 c2 c3
c1 is the number of chops per beat. When 4 different notes are requested to be played, you may choose two parts (with c2 and c3 being values 1 to 4 each with c2<c3) to alternate or 'CHOP' similar to the way a xylophone is often played.
! F f1 f2 f3 f4 (default is 'F 2 2 2 2')
f1,f2,f3,f4 are the fades for each part indepentently. When a chord is playing, it starts with the volumes on the "V" statement multiplied by 2 and is faded 3 times by adding the fade numbers to the volumes as the chord is playing. This creates the "piano" sound. Note these numbers may also be negative for special strange effects if the chord notes start quietly enough to get louder as they play. Note if less than 4 numbers are entered, the last entered is copied for the unentered parts on the fade statement. When the fade becomes too much (either positive or negative), the note is turned off).
! any music statment(s) / comment
This "/" statement may be used before any statement to temporarily comment out the remainder of the line, or for information not to be processed, ie. putting the key number on the "M" statement. It simply tells the preprocessor to scan the next line immediately to look for music instructions.
! Stop
"Stop" is any character string to stop sound processing and return to E.B.(max lengh of 6 characters)

Also active keys when sound is being processed:
H : Hold the currently playing note until released.
T : Hold notes until TI99 sound processor is finished (should be used to determine b2 on the 'B' statement).
FCTN 4 (CLEAR): Immediately return to E.B. (hold key until next statement is scanned).
FCTN + (QUIT): Return to TI Banner Screen (hitting quit does not necessarily mean you must reload EMUSIC for more music but CALL INIT does).

Note: If you lose the music routine (ie. CALL INIT) just run DSK1.LOAD, then select DSK1 and MUSICPREXN from the directory.

Recommended Procedure for Coding a Song.

Start your own DSK2 songs disk (any label). Copy the DSK2.LOAD to it (optional unless you have only one disk drive). Code and SAVE your songs as follows:

1. Look at the key signature (how many sharps or flats are at the beginning of each line of music) and turn to the SCALES page matching it.

2. Look through the music for the smallest note (ie. the note with the most flags). Code the smallest note on the "T" statement as 1. For example if the smallest note is a 16th note (with 2 flages), then the quarter note is coded 'T 4'. This is necessary since EMUSIC cannot handle fractional beats.

3. Decide how fast the song should be played on the 'B' statement. The first number on the 'B' statement is the number of 60ths of a second that the smallest note will play, as in step 2 above. For example, if the song contains 16th notes, and the first number on the 'B' statement is 20, when the 16th notes are played with the 'T 1' statement, they will play for 20/60=1/3 second. The second number on the 'B' statement is about 100. It is the number of times a wait loop is executed waiting for the playing notes to ALMOST finish. This is done to prevent the click that is heard if the note is completed before the next is started. To determine the second number on the 'B' statement, first try about 100. Play the song. Press the letter 'T' (upper case) while the song is playing. The clicks will be heard between notes because this makes one note finish before the next is started. If the second number on the 'B' statement is:

Too small: Pressing 'T' makes the tempo much slower.

Too large: Clicks are heard without pressing 'T'.

Adjust the second number on the 'B' statement so the clicks are never heard but the song is as close to the correct tempo as possible. 4. Enter the command OLD DSK1.MUSICMODL. Correct the 'B','M', 'K' and 'C' statements to match your new song. The comment after the 'M 0' statement should contain the key number found on the SCALES page. Then if you wish to switch to minor, you don't have to find the music to see what key it was written in.

Note: If the music itself changes key, I recommend putting in the music code another 'M' statement with the new key number where the key change occurs. When this music is to be played in minor, both 'M' statements need to be activated for their respective keys.

5. Enter the 'V' statement BEFORE the first 'N' statement(s) that it pertains to (either on the same or previous line). Keep in mind that the TI-99 sound processor receives the note only when both the 'V' and 'N' statements have been processed and another 'V' or 'N' statement is about to be processed. Also 'N' statements save notes WITH their volumes. The Fade 'F' statement is also needed before the notes with their volumes are sent to the TI-99 sound processor if you wish to change the default fade of 'F 2 2 2 2' statement.

6. If the music you are coding is written for a vocal part, always code that part as the first part of the 'V','N','K' and 'F' statements. When the solo is resting, the first part note is 70. Also the volume of the vocal part could be louder than the other parts or accompanyment, and the fade for the vocal could be 0 for a first try.

7. Code the entire music through without reguard to the first endings, second endings, CODAs etc. When finished, add 'S' and 'G' statements to have the EMUSIC subroutine play the entire song correctly, keeping in mind:

1: The last executed or scanned 'S' statement is in effect for each of the 1 to 9 line numbers.

2: The 'G' statement transfers to the last saved 'S' corresponding to it. For example:

```
! S 1
! G 1 2
```

indicates goto statement 1 twice and fall through the 'G' statement the 3rd time. This means that portion would be heard 3 times. After the 'G' statement is executed once as a 'fall through', it will then be active again as a Goto for another 2 repeats if executed again.

If a 'G' statement requests an unsaved line number, 'G' will foreward scan for the first 'S' statement with the requested line number, then goto it.

3: Adjust the fade for each part with the 'F' statment. For best sounding results, when a note spans several statements, turn its fade off until its last occurance (see song MTNHIGH).

4: Adjust the chop and determine which parts will be chopped if applicable. To perminently disable chop in a song, just set one of the volumes to 7 if all four parts are playing. Otherwise chop is off anyway.

5: At anytime, local fixes may be made to improve a sound with ANY of the EMUSIC Language Statements. Most songs I coded do not have local fixes since this makes it more difficult to play the same song many different ways.

System Requirements:

Extended Basic, Memory Expansion or an equivalent and atleast one disk drive are required. Also a sound monitor or TV is required.

Memory Usage:

EMUSIC is not relocatable from >24F4 to >34CC which leaves >B1C bytes for other assembly routines (including their 8 byte REF/DEF table entries) in low memory expansion.

Notes on Included Music Files:

These files do not represent the best that EMUSIC can perform. I have quickly thrown them all together as examples of what can be done with EMUSIC. Some special things to look for in these files are:

1. CHIMES: Shows how a note's fading may be continued after the next note begins.

2. DEMO: Shows how graphics, sprites and alternating between BASIC and EMUSIC may be done. Also this gives examples of key changes, converting to and from MINOR.

3. ENDTAG: Special study of this program shows the full power and ability of the 'S' and 'G' music statements by executing the same tag in three passes with each given a different set of feature instructions, ie. key, chop etc.

4. DANCER and UNEEDEDME are two I bought sheet music and coded up. I hope you enjoy them as much as I do.

5. I have put the LOAD program on the DSK2 SONGSEZ diskette for anybody with only one disk drive. The initial load must always be from the DSK1 EASYMUSIC Diskette. If you have two or more disk drives, you should copy one side of my system to a separate diskette and put them in DSK1 and DSK2 as I have labeled them.

Some Final Notes:

I hope you find this program enjoyable, entertaining and educational. As far as I know, this is the only program of its kind for the TI-99/4A. I hope soon we will be trading music disketts with each other (like 45 records) except we can keep a copy too. If you produce a particularly nice sound and video presentation you wish to sell, please provide your customers with my two sided diskette as you received it (if they don't already have it).

I have started buying music just to code up to listen to. I enjoy playing songs differently each time (ie. to sound like a music box, like a banjo, or like a calliope etc.). The songs may either give you the choice of playing again or going to the song menu. Another option is one song could start the next like an automatic record changer. I am in a barbershop quartet (singing bass) so I at times use this program to learn new songs.

I have spent many hours on this program and would like to share it with you. I would appreciate it if you would help by distributing a copy to your TI-99 friends who like computer music. Also, if you like this program, it would be really appreciated if you could send $10 to help me convince my family that all the sacrifices they made for me to write this program were worth while. Anybody, including computer clubs, may distribute copies under the following conditions:

1. That the program be made available to others at NO CHARGE other than normal reproduction fees.

2. The complete package (2 documentation files, 2 load programs, 13 music files, and two SYSTEX files--although not needed for music) must be included with each copy

3. My identification must not be removed.

4. While this program is free to be copied for personal use only, I ask that you get written permission from me for any other than personal use of this program, as outlined above. Lets keep the whole world of TI-99's singing. You may contact me through the Delaware Valley Users Group whose library contains a copy of this disk, or at:

Norm Sellers
15 Dorset Drive
Broomall, PA 19008
Phone: (215) 353-0475.

EDITORS NOTE.

This program was purchased from the TIsHUG shop and may be of interest to others.

END OF ARTICLE

FUTURE MEETINGS

In future, meeting topics will be set down well ahead of time. They will follow a set pattern each week. The general timetable is as follows.

12.00-1.00 Directors meeting
12.30-4.00 Shop
01.00-1.30 Group Meeting
01.30-4.00 Groups for TI and IBM

GROUP MEETINGS

GROUP ONE will meet with Ross Mudie, and will concentrate on using the basic TI system. It will aim at helping members to use their TI effectively. Ross will take the group through a hands on, question and answer, demonstration approach.It will develop according to the questions and interests of the group.

GROUP TWO will meet with Larry Saunders each week, and test out new programs or shop software. Programs will be demonstrated, and help given with any requested software.

Those interested in the IBM will have the choice of two groups.

IBM GROUP ONE will initially concentrate on building up a club system from scratch with Peter Schubert. Hopefully we will gradually build up a really useful system, which will help all our members to both understand and use their systems.

IBM GROUP TWO will concentrate on using the basic software tools efficiently. We will bring in the necessary people to help us at each level.

The program is as follows;

APRIL	Using a DOS system. An overview of DOS.
MAY	Using XTREE to manage files.
JUNE	Using XTREE GOLD
JULY	Using WINDOWS
AUGUST	WINDOWS cont.
SEPTEMBER	WORD PERFECT
OCTOBER	WORD PERFECT cont.
NOVEMBER	SPREADSHEETS. Using AS EASY AS 123.
DECEMBER	ANNUAL GENERAL MEETING.

Special interest groups will be developed as needed.

Be in it mate,

OR I'LL RIP YOUR

BLOODY ARMS OFF.

END OF ARTICLE

Pheta Doc Ver 1.5-051187

By Larry Saunders

S P A C E S T A T I O N P H E T A

Documentation for Mini-men E/A version 1.0
By Jeff Bunting 1985. Converted variable 80 format and edited by Larry Saunders 1994

NOTICE: This program is being distributed as shareware, freeware, fairware, or whatever you like to call it. Feel free to copy it and give it to your family, friends, and neighbors. If you like the program and would like to reimburse me for my time effort please mail me for my current home address on Compuserve 71550,2316 or Genie J.D.Bunting or write in care of Roanoke Valley 99'ERS P.O. BOX 12522 Roanoke, VA 24026.

BACKGROUND: One day, while traveling through an unexplored sector of space, your ship's scaners picked up a distress signal. A few parsecs away. Upon following the signal you found that it emanated from a battle-scarred space station.

When your ship neared the station, however, its automatic defences came to life severely damaging your ship. Your only hope is to dock with the station and try to find some way to replenish your ship's energy reserves.

You manage to scrounge up some old A-37 type spacesuits from the cargo hold. They're not very good, but they're all you got. Your pulse quickens as you step through the airlock and into the old, abandoned spacestation, not knowing what perils await you.

The object of SPACESTATION PHETA is to explore the many different levels of the huge structure in an attempt to collect as many energy packs as you can.

Each level of the station has its own dangers to be wary of as well as the constant threat of exhausting your oxygen supply.

INSTRUCTIONS

Here are a few things you will be seeing in the space station

KEY	10 Points	X Level
OXYGEN	100 Pts.	+ Oxygen
ENERGY	200 Pts.	1st Pack
	300 Pts.	2nd
	400 Pts.	3rd
???	5000 Pts.	+ ???

YOUR BRAVE SPACEMAN

More things you'll be seeing, Ladder, Electrified floor, Doors,

Next is a brief discussion of each object.

Keys are used to unlock doors - you can have only one key in your possession at a time and each key can be used only once. Thus, after going through a closed door your key will be expended.
Your score for picking up a key is determined by multiplying the level you are on by ten. Thus all keys on the first level are worth ten points each, keys on level two are worth twenty pionts each, and so on.

Picking up an oxygen tank awards you with one hundred points and some additional oxygen.

Your oxygen level is indicated by a black bar at the bottom of the screen. If you run out of oxygen then you will asphyxiate. (A TERRIBLE WAY TO GO...).
Keep in mind that you receive a bonus for each unit of oxygen you have remaining after completing each level.

The acquisition of energy packs is your primary objective. Scoring for energy packs is progressive. that means the score value increases with each one you pick up. The first energy pack you pick up on a level is worth two hundred points. The value for the next one is three hundred, the next one is worth four hundred and so on. The scoring begins at two hundred at the start of each new level and after you lose a spaceman. If you are able to collect every E.P. on a level then you are awarded a bonus which is a progressive.

This bonus starts at one thousand pionts and increases by one thousand points each time this feat is accomplished. It is possible to get every energy pack on every level. It is not always easy to do this, but it is always possible.

EXTRA SPACEMEN:

Extra spacemen are earned by pepeitively obtaining all energy packs on a level. The first extra spaceman is awarded after three levels have been completed in this manner. The number of levels which you need to clear increases as the game progresses.

LADDERS:

Your spaceman can climb UP or DOWN Ladders. He cannot, however, JUMP onto or OFF of a ladder.

FLOORING:

This part is quite simple. Girders are safe to walk on. Electrified floors are not safe. There are a few other types of floors, but you'll have to experiment with them yourself.

DOORS:

In order to go through a door you most have a key. Once you go through a door you are free to pass through it without a key. Remember that each key can only be used once.

Moving your spaceman you have the option of using either the keyboard or a joystick for input. Both methods consist of five commands Left, Right, Up, Down, and Jump.

MOVING YOUR SPACEMAN:

Your spaceman can safely fall down one level.

A fall of more than one level can be fatal!

JUMPING:

Your spaceman can jump gaps of space. When the jump button is pressed your spaceman will jump either Left or Right, depending upon which was the last direction in which he moved. Remember that you are unable to jump onto or off ladders. If there is an obstruction in the path of a jump them the jump will be cut short. A few other notes about jumping. Your spaceman - can jump down two levels (but fall only 1 level) - can jump onto objects - can only jump small gaps - cannot jump if there is something directly above his head.

STARTING THE GAME

After following the loading procedures (given later in this program) and running the program the title screen should arrear. Press S to start the game or O to view the options.

OPTIONS:

Pressing O will bring up the options screen. There are 7 options to choose from. Lettered A through G.

Each letter is as follows:

A Starts game.

B-D Selects method of input. Asterisk indicates the current mode. When using Joysticks be sure to place Alpha lock in the UP position.

E Toggles sound ON / OFF. You may need to adjust the speed (OPTION F) if you turn the sound OFF.

F Sets movement speed of the spaceman. The speed can be set from 00-99. 00 is the fasteet speed and 99 is the slowest.

NOTE: That 2 digits must be entered. If you wish a number below 10 then a zero must be entered as the first digit.

G Lets you select the keys you wish to use if playing on the keyboard. Be careful not to select the same key for two different directions or you will not be able to move properly. If you make a mistake in this option finish each category and then select option G once again.

LOADING INSTRUCTIONS:

Program loading from disk:
1. Select TI Extended Basic.
2. Select "Space Station Pheta" from the ROOT menu the program will load and run.

LOADING E/A VERSION

1) Select EDITOR/ASSEMBLER.
2) Select OPTION 5 RUN PROGRAM FILE
3) TYPE DSK1.PHETA1
4) PRESS ENTER

If you have any questions or comments please write me. Am also instrested in receiving original screen designs, the best of which I will compile and distribute with credit to the designers.

END OF ARTICLE

HUNTER VALLEY

Phone numbers

There has been a mix up with the phone numbers for the Hunter Valley Regional Group, the editorial staff wish to correct this now and hopes that no inconvenience to its members and others wishing to use there facilities, the phone numbers are as follows: BBS (049)428176 If you wish to leave a message please use (049)428617.

GAMES INFORMATION
IBM Edition
#3

By Robert Brown

Welcome to the third article of Games Information for IBM Computers. So far I have given you the solutions to Ultima I II. This month we finish off the series pf Ultima Games with Ultima III. I hope you enjoy it!

ULTIMA III Strategy

Lord British's "Ultima III: Exodus" is his latest, and most ambitious, offering in an outstanding fantasy role-playing series. Every aspect of this game, from the documentation to the computer graphics, is a logical, yet enhanced, extension of the earlier two chapters of the "Ultima" saga. In "Exodus," however, there is a fundamental new twist: No longer must a solitary character strive to overcome the evil abroad in the land; a party of up to four players may join the quest. The strategy tips in this article should aid your party Move out by thoroughly studying the four booklets that come with the game: The Book of Play, the Ancient Liturgy of Truth, the Book of Amber Runes, and the Player Reference Card. You must then form a party by designating race and profession and by allocating 50 points among four attributes. Each profession has different valuable characteristics, and part of this game's flexibility is that you may try various combinations.

Perhaps the best initial party will be composed of an elf thief, a dwarf fighter, a bobbit cleric, and a fuzzy wizard. A dwarf or bobbit paladin, or a human ranger, may easily be substituted for the dwarf fighter. It is certainly desirable to include both a cleric and a wizard in your group because only they have the capacity to develop advanced spell-casting powers.

In combat situations, each party member takes a turn in order. Therefore, it is important to position the characters in such a way as to take advantage of their particular abilities. Generally, characters who have more weapon/armor options should be in first and second place; the spell casters should be in the next rank. I had very good luck with "Snatch" (elf thief) and "Devo" (dwarf fighter) in the front, with "Wanda" (fuzzy wizard) and "Clarissa" (bobbit cleric) in the rear. Since combat may seem too fast and confusing at first,

it may help you to give your characters mnemonic names.

II. JOURNEYING ONWARD

Your little band begins its adventure near the castle and city of Lord British on the grassy plains of the continent of Sosaria. Immediately equip your characters with daggers and cloth armor. Then, save the game. Frequent "saves" will help you to avoid getting stuck with a group of dead bodies on your hands; if disaster seems imminent, you can turn off your machine and restore your previous game position.

Enter the town, where it is safe. Examining the player roster, you will see that each party member possesses 150 gold pieces and 150 items of food. You should transfer extra gold to one character, preferably a fighter type, so that he may purchase a bow. Long-range weapons such as bows and slings are vital to surviving combat. Of course, daggers may be thrown, but they are then lost; it veed *transacts with abundant supply. Remember to re- merchants or you might find a character with a bow and leather armor in her inventory standing naked and empty- handed against eight titans. It goes without saying that your group must have adequate supplies of food.

Any player may attempt to steal from the gold-laden chests that are located in weapons and armor shops; however, thieves have the most success. Lower level characters haven't a prayer against guards, so larceny is a risky endeavor. It is not possible to steal food in this game; however, chests occasionally contain a valuable weapon or item of armor. If you wish to steal with impunity, you can bribe guards. Greasing guards' palms causes them to instantly disappear.

Gold, money, crass lucre: You will need copious amounts to advance in "Ultima III." Sometimes you'll think it's money that makes Sosaria go 'round! Although many of the clues vital to winning the game may be freely obtained by transacting with people in towns and castles, gold is needed to purchase not only important information from barkeeps and oracles, but is also necessary in order to raise your players' attributes, equip them with better weapons and armor, and bribe guards. You may wish to create several "sacrificial" characters whose sole purpose is to transfer all their gold, weapons, food, and armor to one of your "real" party members. Using this admittedly opportunistic technique, your "real" party can become well-equipped even before its first foray.

A note about "roll over": Lord British has almost British has almost completely corrected the problem

(from "Ultima II") of players' "numbers" rolling over to zero after 99 or 9999. Now, if you inadvertently purchase more than 99 daggers, only 99 will show up (not 0) on the status report. The same holds true for other equipment and players' attributes. However, a bug remains in the food counter. Be careful not to exceed 9999 items of food or very strange things will happen to your players' names and armor. [Note: Rollover still occurs in some versions.]

III. THE INHABITANTS OF SOSARIA

Within the towns or Lord British's castle, you will encounter guards, merchants, barkeeps, jesters, oracles, or healers as well as any of the regular player types such as fighters, thieves, clerics and wizards. You must transact with almost everyone you meet in order to learn the clues which are vital to winning this game. Carefully explore every inch of every town; you can locate informative individuals behind shops, within wooded areas, or on the other side of locked doors. [See the "Layout" and "Clues" sections for more specific details.]

IV. COMBAT

Much of your time, especially in the beginning stages of the game, will be devoted to slaying monsters. On land, in the sea, and deep within the dungeons of Sosaria your party will engage in combat with more than two dozen species of monsters as well as renegade player types. Examine your Book of Play for the descriptions of these creatures.

Some monsters (orcs, goblins, and trolls) may be dispelled by your wizard's "Repond." The undead creatures (skeletons, ghouls, zombies) are turned by your cleric's "Pontori." When your cleric or mage reaches his or her full potential, he or she (or "O"!) can cast helpful (transport, light, heal, cure, map, resurrect) or devastating (wound, negate time, destroy) spells. However, since spell casting temporarily drains magic points, most combat involves the use of weapons. Be aware that a cleric's use of the "P" (Resurrect from Ashes) spell drains wisdom points.

The weapons available range from daggers to +4 bows (6550 g.p.) to the mysterious "exotics." Naturally, as your players gain strength, one arrow may be all that is needed to dispatch the dread balron. Long-range weapons should, therefore, be obtained as soon as possible. The first shot in a combat situation, slings or bows can mean the difference between life and death. Hand-to-hand weapons are relatively ineffective because your players have to waste turns moving toward the enemy and taking hits instead of fighting. Since daggers may be thrown, they are more versatile than maces. The same is true regarding the use of bows as opposed to swords. The ranger I created was more successful and efficient with his simple bow than a +2 sword.

Armor is, of course, vital in protecting your players from injury. Depending on their professions, characters may wear anything from cloth to +2 plate (8250 g.p.). Try to obtain leather armor, at least, for those players who may wear it; they will survive more easily and will rapidly advance in experience levels.

Players' hit points are directly related to their experience. Each character begins with 150 hit points at level one. As the game proceeds and your players slay numerous monsters, their experience points will increase. Remember, however, that only the character striking the death blow wins experience points! As additional levels are reached, the party should visit Lord British who will see that his wizard raises the characters' hit points. However, after level five, the Lord will refuse to raise the hit maximum. Instead, he says: "Seek ye the Mark of Kings." Once your group obtains this mark (one of four available -- each with a different purpose), Lord British will raise your characters' hit points, according to his or her experience level, to the absolute maximum of 2550.

If you want to build up backward players' hit points, you must position them so that they may inflict the fatal blows. This may even involve passing over (by hitting the space bar) a couple of players in favor of your "slower" ones.

After your players reach level five or so, and have purchased decent weapons, armor and, hopefully, a few powders (to negate time), they may use the following procedure to rapidly build up more experience points: Enter a town that has only two sets of guards stationed at the entrance; attack and kill the guards; immediately leave the town and save the game. Guards are worth 15 experience points each, and, although they are extremely powerful, they don't throw firebell.

While your party is still relatively inexperienced, try to avoid sea serpents, men-o-war, and pirates. You can usually see these opponents approaching, and can detour around them. Do not descend below level four or so in any dungeon until your party is quite strong; otherwise, the manes, devils, and wyverns, among others, will surely decimate your group. Several monsters not only wound your players, but poison them. However, when your party is healthy and well-equipped, it can win thousands of experience points and gold pieces by slaying these creatures. In the case of pirates, your group can also commandeer a frigate; no experience points or treasure is gained if the ship's cannons are used in battle.

Before your group obtains long-range weapons, try to avoid thieves, cutpurses, and brigands. These characters love to steal the weapons or armor that are not currently in use by players.

Within dungeons, the invisible gremlins like to steal food. Be sure to carry plenty of rations or your band might starve to death!

Here is a list of various opponents and the experience points received by players when they defeat these creatures:

Merchants = 1; Jesters and Grasss = 2; Goblins, Trolls, Orcs, and Floors = 3; Skeletons, Zombies, and Ghouls = 4; Cutpurses, Brigands, and Thieves = 5; Golems, Giants, Titans, and Horses = 6; Fighters, Manes, Gargoyles, and Daemons = 8; Wizards, Pincers, and Bradles = 10; Serpents, Dragons, Griffons, Wyverns, and Guards = 15; Men-O-War, Orcuss, Devils, and Balrons = 20.

The tougher the enemy, the more experience points gained by your players.

Note: Do not try to engage Lord British in battle; he is invincible. He will allow you to take the chests behind the force field, but will attack if you initiate violence within his domain.

V. HEALING

The natural result of all this fighting is injured, or even deceased, party members. Be sure that you carefully map Sosaria, pinpointing the locations of healing kiosks. Your group will have to visit the healers frequently, especially if a character is poisoned. As always, gold works wonders.

Of course, clerics as well as advanced wizards may cast healing spells. Use their powers with discrimination since every spell drains magic points .

Healing fountains are located inside dungeons. A drink from one will restore your players' lost hit points. You cannot tell whether a fountain is healthy or poisoned without testing it. There are four types of fountains: good, innocuous, harmful, and poisonous. Save your game before descending into a dungeon and be careful!

VI. TRANSPORTATION

There are three modes of transportation in this game: foot, horse, and frigate. Most of the time, your players will be on foot and their food supplies will diminish rapidly as they explore Sosaria. Obtaining a horse not only slows the decrease in food, but allows your party to out run monsters. There are only two places where horses may be purchased or stolen: in the City of Dawn or the City of Devil Guard. [Check the "Layout" section for the locations of these cities.] Horses may be ridden through moon gates.

Frigates may be commandeered by defeating a pirate crew. Once you have a frigate, your party may then traverse the seas, visiting important islands and learning the secret of the whirlpool.

The whirlpool is both beneficial and harmful. It may destroy ships that are docked and empty. Yet, by sailing into the whirlpool, your party will be transported to the Lost Continent of Ambrosia where the players will be able to raise their attributes and obtain the four cards needed to destroy Exodus.

VII. ITEMS

There are four valuable items which your players must purchase in order to win the game. These are: torches, keys, gems, and powders. Only in thieves' guilds can your party buy these special items [check the "Layout" section for locations].

Torches, of course, are needed for light while the party explores dungeons. Because of the prevalence of "strange wind" throughout all of the dungeons, numerous torches should be carried.

Keys are used to unlock doors. They are particularly useful in Lord British's Castle, the City of Montor West, and Ambrosia.

Gems give you a bird's eye view of your current location. Buy as many gems as possible; you can use them to map dungeons, explore the nooks and crannies of cities, and proceed effortlessly through mazes.

Powders are, perhaps, the most valuable objects of all: They stop time. Use powders frequently during fierce battles, especially in the Castle of Death [Note: This may not work in some versions]. The incessant fireballs, though, are not affected by powders.

VIII. DUNGEONS

Unlike in "Ultima II," your group must explore all the dungeons in order to win this game. Four "marks" must be obtained by each party member. Marks are usually located in a wall on the 8th level of a dungeon. The Mark of Kings is needed before Lord British will raise the characters' hit maximum above 550; the Mark of Force is used to pass uninjured through force fields; the Mark of Fire allows the party to walk through fire; the Mark of the Snake is needed to bypass the Silver Snake which guards the Castle of Death.

The best dungeon strategy is to use plenty of gems to locate ladders and/or cast ladder up and ladder down spells. It is not wise to linger on the upper levels if your group is simply looking for marks. Enter the dungeon and descend. Then, obtain the mark and exit the dungeon. Of course, it is helpful to explore all levels of all dungeons, but this can be fatal if your party is weak.

IX. MOON GATES

Ah, these mysterious and ephemeral portals! What is their secret? By stepping into a moon gate, your group will be transported to various locations, some of which are inaccessible by any other means. There are eight moon gates, and the most important is, perhaps, the one that delivers your players to the dungeon of the Time Lord. Experiment with different gates and note their locations, the timing of their appearances, and their destinations.

An understanding of the cycles of the two moons of Sosaria, displayed on the cloth map and constantly counted on your screen, is vital in your search for the legendary City of Dawn. [See the "Layout" section for more details.]

X. DEFEATING EXODUS

Who or what is Exodus? The product of an unholy alliance between Mondain and Minax, Exodus is more machine than monster. Could he or it be a computer? Perhaps so: Exodus' powers are legion and he is "turned off" by inserting four cards, in proper order, into slots in his "body." The Time Lord, if he will, can tell your party the proper order of the cards.

Do not attempt to destroy Exodus until your party is extremely strong. Located within the aptly named Castle of Death, Exodus is protected by an army of fiends as well as a continuous stream of fireballs. Further, the usual weapons are ineffectual against monsters in this castle; your group must be equipped with "exotic arms" in order to slay these creatures in hand-to-hand combat.

After your little band battles its way to Exodus, it must conquer the strange and invisible floors. The floors repeatedly attack until they have been completely destroyed. The best strategy against the floors is to have your cleric cast his "O" spell and/or your wizard cast her "P" spell. Then, move forward step-by-step, attacking in all directions. The floors are difficult, but certainly not impossible to overcome.

A final note: If you try to leave the Castle of Death without destroying Exodus, or if you linger near the entrance, your players will have to battle the grasss. The same strategy should be used against grasss as is used in combat

=-=-=-=-=-

I. THE CLOTH MAP

The map that is supplied with the game is very useful, especially if you take out a sturdy black pen and mark locations on it. I used triangles to designate castles, squares to represent towns, circles for dungeons, and squiggly lines for moon gates. The map shows the continent of Sosaria as explored by the late, great Hawkwind. Your party will exceed even his accomplishments before the game is over.

Also displayed on the map are a series of drawings of various locations as well as representations of the cycles of the two Sosarian moons. The runes are not difficult to decipher. For example, on the top left is a picture of the Castle of Lord British. On the top right is Death Gulch. The moon of Trammel has a 48 day cycle, and the moon of Felucca has a 16 day cycle.

II. THE CITIES

Here is a list of the cities of Sosaria, their locations, and their important features:

1. CITY OF LORD BRITISH: Located to the northeast of Sosaria, this city overlooks a harbor. Within the town you will find a pub, a grocery, a weapons shop, and an armory. As always, be sure to transact with everyone you meet.

2. MONTOR WEST: South of Lord British are the twin cities of Montor West and Montor East. In the western town you will find a grocery, two pubs, a weapons shop, an armory, a prison, and four guard stations. It is important to speak to two thieves who are inside prison cells. Either bribe the many guards or be prepared for a series of long battles if you enter the prison.

3. MONTOR EAST: Again, you will visit a pub, an armory, and a weapons shop. Guard stations abound, but you need not do battle in this town. Simply transact with all the player types you meet; valuable clues are to be had here.

4. CITY OF YEW: Within this holy city are dozens of clerics as well as the Circle of Light, Rogation Worship, Sanescere healing kiosk, and the Aliment food shop. Take care in this heavily forested town else you might stumble into a wall of fire. The City of Yew is located in mountains to the west of Lord British.

5. CITY OF MOON: Further west from Yew lies Moon. Moon may also be reached by travelling southeast from Montor East. This small town stand. The forested areas should be careful ly explored for hidden clues; avoid those balrons, though!

6. CITY OF GREY: Due south of Moon, on a southwestern tip of Sosaria, you will find the City of Grey. This town has a pub, a grocery, a weapons shop, an armory, a thieves' guild, and a chamber containing chests. Vital clues are to be had from a thief, a fighter, and a cleric. Be sure to visit Grey often in order to stock up on gems, powders, torches, and keys from the guild.

7. DEATH GULCH: Off the coast, on a large island to the southeast of British, is Death Gulch. This town is heavily guarded, but violence can be avoided. Here you will find a pub, grocery, weapons, and armor. The daemon and guard at the front entrance may be cautiously bypassed in order to enter the town. Bring keys. You can also sneak into Death Gulch through the forested areas to the north and south of the entrance. This town has several mazes, force fields, and a river of fire. None is of much importance. Lots of chests here.

8. CITY OF FAWN: On a small island off the coast of Sosaria, to the northwest of British, is the City of Fawn. Here you will find a pub, a food shop, a healing kiosk, and a thieves' guild. Four clerics will impart significant clues.

9. DEVIL GUARD: This town is locked within mountains to the southwest of British. To reach Devil Guard, your party must take one of three moon gates [discussed below]. A pirates' frigate, firing from the shore of a small lake with no outlets, attacks your group when you visit this town. Devil Guard is a dangerous place, but a very important one. Here, there are six fighters and one thief who can give you winning clues. Devil Guard contains a pub and grocery, a healing stand, another thieves' guild, and a stables. You may purchase four horses here for 800 g.p.; they cannot be stolen.

10. CITY OF DAWN: Dawn, the city of myths and magic! It is located, for a brief moment only, in the forest southwest of British. Dawn possesses the usual line-up of shops: food, pub, weapons, and armor. However, you will also find another thieves' guild here as well as an oracle and stables. Both the weapons and armor shops sell advanced items. Bring thousands of gold pieces and your party will leave Dawn very well-equipped. Horses here may be stolen, but expect a series of battles with Dawn's many guards. There are three clerics, hiding in the southeast corner of the town, who can tell you much about "exotic arms." However, two of the clerics will not let your party pass in order to transact with the third; you will have to kill them. Prepare your group by bribing as many guards as possible before dealing with the clerics.

III. THE CASTLES

There are two castles in this game: Lord British and Death. Lord British's Castle is next to his city. You will visit the castle many times in order to transact with the Lord and to purchase cures or healings. Most of the areas in this castle are unimportant; however, by using keys to unlock doors, you can locate several individuals who have important knowledge about your quest. In particular, explore the prison, the winding corridors, and the exterior of the castle.

The Castle of Death is the domain of Exodus. It is located on an island off the southwest coast of Sosaria. Reach it by sailing past the Silver Snake (be sure to bring horses with you!). If you survive the dangers, you will find Exodus in a chamber against the northern wall.

IV. THE DUNGEON: Dungeons are hazardous places. Not only do they house an abundance of vicious creatures, but there are traps, gremlins, strange winds, poisoned fountains, mazes, and dead ends. Here is a list of the dungeons of Sosaria, their locations, and their important features:

1. PERINIAN DEPTHS: This dungeon is located to the northeast of the City of Lord British. There are two Marks of Kings on the 1st level, and another on the 8th level. A Mark of Fire may be found on the 8th level as well. There is a healing fountain in the southwestern corner of the 8th level.

2. DARDIN'S PIT: To the northwest of British is Dardin's Pit. The Mark of Kings may be found in two locations on level 8.

3. FIRES OF HELL: Surrounded by fire, this dungeon is located in mountains to the southeast of British. The Mark of Force and the Mark of Fire are on the 8th level. There are also four fountains on level 8, one of each type.

4. DOOM: Southwest of British, in a heavily forested area, is the Dungeon of Doom. In the center of level 8, you will find the Mark of Force. All four types of fountains are on level 8 as well.

5. MINES OF MORINIA: North of the City of Grey, on the west coast of Sosaria, are the Mines. The Mark of Kings is located on level 3 and on level 8. Also on level 8 is the Mark of Fire. Your party will use transport and ladder spells often in an effort to fully explore this dungeon.

6. SNAKE: On an island off the southeast coast of Sosaria is the Dungeon of the Snake. Important clues are to be had on levels 2 and 3. There are two healing fountains on level 6 and one healing fountain on level 7. Both the Mark of Kings and the Mark of the Snake are located on level 8.

7. TIME: This dungeon is completely surrounded by impenetrable mountains and may only be reached via moon gates [discussed below]. There are two healing fountains on level 1, another on level 4, and another on level 8. The Mark of Kings may be found on level 2 and on level 8. Most importantly, the Time Lord himself resides on level 8. Listen carfully to what he says.

V. THE FORGOTTEN LAND: AMBROSIA

In order to raise your players' attributes and to obtain the cards needed to defeat Exodus, your group must pass through the whirlpool to a submerged located on Ambrosia: Dexterity (southeast), Wisdom (northeast), Intelligence (northwest), and Strength (southwest).

Use gems to map this continent; the many mazes can be quite confusing. Also, keys are needed to unlock several different doors. Be sure to bring lots of gold!

Commandeer frigates from the pirates you encounter; you will need to cross water to reach two of the shrines. The game cannot be "saved" while you are visiting Ambrosia, but you can sail back into the whirlpool to return to Sosaria.

VI. MOON GATES

There are eight of these mysterious portals scattered throughout Sosaria. Depending upon the phases of the twin moons, the shimmering gates will teleport your party to various different locations. Here is a list of the locations of the moon gates, the phases of the moons when the gates appear, and their destinations:

1. NORTH OF THE CITY OF MOON: (00) -- static; (01) -- to the dungeon island SE of British; (02) -- to the north gate at the mountainous dead end.

2. DUNGEON ISLAND SE OF BRITISH: (13) -- to the south coast, southwest of the Montors; (14) -- to the south gate at the mountainous dead end; (15) -- to the Castle of Death.

3. NORTH GATE AT MOUNTAINOUS DEAD END: (26) -- to Devil Guard; (27) -- to the Time Lord's Dungeon; (20) -- north of the City of Moon.

4. SOUTHWEST OF THE MONTORS: (31) -- to the dungeon island SE of British; (32) -- to the north gate at the mountainous dead end; (33) -- static.

5. SOUTH GATE AT MOUNTAINOUS DEAD END: (44) -- static; (45) -- to the Castle of Death; (46) -- to Devil Guard.

6. CASTLE OF DEATH: (57) -- to the Time Lord's Dungeon; (50) -- north of the City of Moon; (51) -- to the dungeon island SE of British.

7. EAST OF DEVIL GUARD: (62) -- to the north gate at the mountainous dead end; (63) -- southwest of the Montors; (64) -- to the south gate at the mountainous dead end.

8. TIME LORD'S DUNGEON: (75) -- to the Castle of Death; (76) -- to Devil Guard; (77) -- static.

ULTIMA III Clues for the Desperate

I. BARTENDERS' CLUES

In most cities, your group will visit pubs. Transacting with a bartender, and payment of copious sums of gold, causes this little tale to unfold: "Ambrosia. Ever hear of it in conjunction of the moons finds a link! Nasty creatures, nasty dark, sure thee ready, fore thee embark. None return so I'm told, from the pool, dark and cold. Shrines of knowledge, shrines of strength, all are lost into the brink. Fountains fair and fountains foul, all are found in dungeons bowel. Exodus: Ultima III, which is next? Now could it be? Seek ye out the Lord of Time, and the one way is a sure find!"

Doggerel, to be sure, but very informative.

II. ORACLES' CLUES

Oracles are found in only two places. Once again, transacting and paying reveals:

"And so the sage said unto thee: If thou can solve my rhyme, you'll learn of marks and playing cards and hidden holy shrines. Of marks I say there are but 4, of Fire, Force, Snake, and King. Learn their use in Devil Guard or death you'll surely bring. Shrines there are again but 4, to which you go and pray. Their uses are innumerable and clues throughout I say. The cards their suits do number 4, called Sol, Moon, Death, and Love. Unto the Montors thou must go for guidance from above. To aid thee in thy cryptic search, to dungeons thou must fare. There seek out the Lord of Time to help you if he cares."

III. OTHER CLUES

By transacting with the various individuals whom your party encounters in cities, the group will learn a series of extremely important clues:

"Marks" are useful!

"Exotics" are useful!

"Cards" are useful!

4 cards, 4 panels! Hot metal leaves a mark!

4 marks, 4 uses!

The King favors a mark!

"Marks" gained in dungeons!

A "mark" helps invoke the Snake!

Mark thee well!

Seek ye the dungeon of fire!

Exodus lies beyond the Silver Snake!

Only with exotic arms can you win!

Dawn comes each new pair! Exotic clues found at Dawn!

Only exotics will protect you from great evil!

Search and ye shall find!

<Pray> in the Circle of Light!
Invoke the Silver Snake!

<Pray> for the Invocation!
To pass you need a mark!

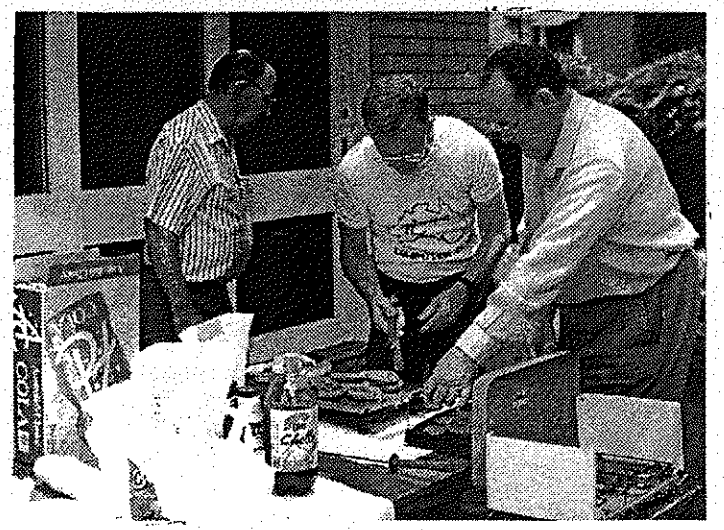
Seek the jester in castle fire!

Seek ye the shrines of truth!

<Search> for cards! <Search> the shrines!

<Bribe> guards! They will leave!
<Dig> up exotics!
<Dig> carefully!
<Dig> on the isles!
<Insert> cards into panels!
West-8, South-35. And Dawn awaits!

TI XMAS 1993



REGIONAL GROUP REPORTS

Meeting Summary For MAY

Central Coast	14/5/94 Saratoga
Glebe	12/5/94 Glebe
Hunter Valley	14/5 21/5/94
Illawarra	10/5/94 Keiraville
Liverpool	13/5/94 Yagoona West
Sutherland	20/5/94 Jannali

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043) 69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The Meetings are usually held on the second or third Saturday of each month at members homes starting at 3pm. Check the location with Geoff Phillips by leaving a message on (049) 428 617. Please note that the previous phone number (049) 428 176 is now used exclusively by the ZZAP BBS which also has TI support. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the second Tuesday of each month after the TIsHUG Sydney meeting at 7.30pm, at the home of Geoff & Heather Trott, 20 Robsons Road, Keiraville. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. Contact Geoff Trott on (042) 29 6629 for more information.

---

* LIVERPOOL Regional Group *

****

Regular meeting date is the Friday folling the TIshug Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 644-7377 (home). Mum will let you know were I am or when I will be home.

NOTE: I will be doing relief managering at several stores, and will not be easy to reach during the day. Some of the stores I will be managering trade to 10pm/11pm/12pm and I am working up to 14 hours a day, 5 days a week.

*** ALL WELCOME ***

13th May 1994
My Place : 34 Colechin St. Yagoona West

10th June 1994
My Place : 34 Colechin St. Yagoona West

for now Larry.
Liverpool Regional Co-Ordinator

SUTHERLAND Regional Group

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Peter Young.

TIsHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the MEADOWBANK PRIMARY SCHOOL, on the corner of Thistle Street and Belmore Street, Meadowbank. Cars can enter from Gale Street and park in the school grounds. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

MAY MEETING - 7th MAY

JUNE MEETING - 4th JUNE

****

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

June - 11th June
July - 16th July

These dates are all Saturdays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00 pm, at the latest. Longer articles should be to hand well before the above dates to ensure there is time to edit them.

****

This months list of words is based around the subject of Four Wheel Drive vehicles

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

Find these hidden words

HOW TO PLAY

All the words listed below appear in the puzzle horizontally, vertically, diagonally even backwards.

ABS	DIESEL	DRAG
ENGINE	EXHAUST	FOURXFOUR
GUTSnGRUNT	HORSEPOWER	JEEP
LANDCRUISER	PATROL	RATIO
RECOVERY	ROUGH	RPM
STABILIZERS	SUSPENSION	TOW-STRAPS
TURBO	WINCHES	

This puzzle was compiled using ASHLEY LYNN'S programme "Word Puzzle" which is avaliable from the TIsHUG shop.

JUST A ONE LINER (ED)

Q. What do cats strive for.
A. Purrfection.