8704 (036)
PUNN

WORDPLAY APRIL 1987

WHAT'S INSIDE

CLUB OFFICERS

President	Keith Fast 777-1531
Vice-President	Dale Kirkwood 646-4354
Secretary	Don Barker 223-1749
Treasurer	Mike King 357-4413

PUNN Staff

Librarians	Ron Mayer 232-7363
	Walt Morey 239-5105
	Jim Thomas 284-2425
Hardcopy	Mike Calkins 636-1839
Program Chairman	Ted Peterson 244-1587
Workshop Chairman	(open)
Membership Chairman	Terry Priest 649-9583
Newsletter Editor	Charles Ball 639-0466
Ass't Editor	Dan Hawes 620-9725

BBS Committee

Chairman:	Al Kinney 640-5860
	Ron Mayer 232-7363
	Mike King 357-4413

BBS Phone Number 233-6804

THE PREZ SAYS . . . .

In past years, the club leaders have released surveys to get a feel for what types of things the members are doing with their computers, and what kind of equipment they are doing it with.

Well, it's been quite a while now since the last survey. We have had a lot of new people join, and we are starting to feel a little out of touch with the masses. Our membership chairman, Terry Preist, has put together a rather inclusive survey which will be distributed at the next club meeting.

I would like to encourage each member to take the time to thoughtfully complete one. These surveys provide us with the information we need to put together more beneficial and informative meetings.

The surveys are voluntary and you can fill out whatever information you feel will help us in planning workshops and programs. This is your chance to tell us what direction you would like to see the club go.

I would also like to thank the entire membership for your envolvement in meetings. Last month, although the hour got late, nearly everybody was attending and participating in workshops. That's what makes our club stand out from some other organizations (toot-toot).

See ya on the 7th.

--Keith Fast

CLUB NEWS & VIEWS

The board is making further studies relating to a TI Faire to be held later this year. Many factors are involved such as hotel accommadations, speakers and vendors. More information on this later....We still need a Workshop Chairman. This job needs to be filled. It provides a much needed service to the membership....The introduction o the new computer is now set for the end of April....Triton will also be starting delivery of their TURBOXT IBM compatible during April so TI'ers will have many interesting decisions to make before long....Be sure to type in the Diagnostic program presented in this months Wordplay-you'll find out how good your consul is working....April workshops will include: Fasterm Part II by Al Kinney and more on PRBase (printouts) by Keith Fast....Don't forget your library when you come to the meetings. There are many fine programs available--and everytime you purchase a disk or cassette you are contributing to the sucess of the clubs treasury as well as obtaining fine software....Your newsletter editor is always looking for articles and programs of interest. Why don't you send him something for the May issue of Wordplay.

****
*
* Murphy's Rule:
*
* Work hard, save your money
* and when you are old you will
* be able to buy what only the young
* can enjoy.
*
****

GENIAL TRAVELER

Genial Traveler is a fairly recent new product on the market. It is a "Diskazine or a magazine on a disk. The Editor is Barry Travers of Genial Computerware, 835 Green Valley Drive, Philadelphia, PA 19128. You will note the spelling of Traveler in the name. It is not a misspelling. Drop the "el" and see what you get.

Those who have subscribed say that Barry has a good grasp on the users who read his product. It is not overly simple or difficult. He includes almost everything a hardcopy magazine does (minus the advertising of course). There are articles, reviews, comments, an editorial, programs, tutorials, etc. In short it is very complete.

One program on his TRAVelER 17 (Vol 1, No 4, side 1) is an updated version of his Archiver program. What is an Archiver you ask. It will "pack" a series of programs/files into one file. Or it will "unpack" a file that was packed using the Archiver packer option. This is a very useful program as related files can be kept together without fear of losing one along the way. Or for uploading and downloading from Bulletin Boards.

Some of the games are reported to be good. This will depend on the user who will determine what good is. Other programs in previous issues included a Multiplan template for 1986 taxes and a program that makes your DV/80 files readable on your monitor using only E/Basic.

Barry draws from all corners of the TI community. He includes articles and/or programs from such people as Peter Hoddie (Pre-Scan It), Tom Freeman, (DISKAS-SEMBLER), Paul Carlton (Fast-Term), Walt Howe, etc.

The document files can be read on your terminal or printed on your printer. This can be a useful feature for those who do not have a printer. Also for those who dislike typing in the source code just to see if a program is as interesting as it sounds, Barry has already done the hard work for you. There is something for everyone.

The cost for for a six issue subscription (one every other month) for a year is $30.00.

ONE LINER

Here's a one liner that will read DIS/VAR 80 files and display them on the screen

```
1 IF F THEN IF EOF(1)THEN EN
D ELSE LINPUT #1:X$ :: PRINT
 X$ :: CALL KEY(0,K,S):: IF
K=13 THEN ACCEPT VALIDATE(CH
R$(13)):D$ :: GOTO 1 ELSE 1
ELSE CALL CLEAR :: INPUT "DS
Kx.FILE? ":F$ :: IF POS(F$,"
DSK",1)THEN OPEN #1:F$,INPUT
 :: F=1 :: GOTO 1 ELSE 1
```

THE PRINTERS APPRENTICE

THE Printers Apprentice (PA) is an exciting new software package from McCann Software. PA is a print shop type program which allows you to custon design pages to be printed on your Epson or Gemini printer. TI-Artist, P files, and TI-Writer text files may be intermixed in any position on the page. Text and pictures may be placed on the page with a resolution of one dot position on your printer. RLE pictures may be converted to TI-Artist format using MAX-RLE and then used by PA.

Printers Apprentice is written in FORTH, and consists of 4 programs that are accessed through a menu. The first program is the Character Editor. This program allows you to design your own fonts, or modify the fonts provided. Six such fonts are provided with PA. A disk with additional fonts is also available. Fonts from other programs such as TI-Artists are not compatible with PA. You can, however, create text using TI-Artist, save it as a picture, and use the picture file with PA.

The second program is the Picture Editor. The Picture Editor gives you limited ability to draw pictures, but it is primarily meant to be used to edit TI-Artist pictures, or clip parts of those pictures to size.

The third program is the Formatter. This program takes TI-Writer text and converts it to any of the available fonts. It also lets you select the density of the print, character spacing, micro-justification, etc.

The last program is the Scheduler. The Scheduler puts all the pieces together and prints the page. It also allows you to place text and pictures anywhere on the page. The Scheduler allows you to place up to 100 different items on a page. The Picture Editor and Formatter programs must be used to create files compatible with the Scheduler. Raw TI-Writer files, and TI-Artist pictures are NOT compatible with the Scheduler.

The manual supplied with PA is probably the worst part of this package. While it contains all the details about the program, it does not contain many examples or descriptions of how to use the various programs. There too many "trees" to see the "forest". With a great deal of study and practice you CAN figure out how to use the program.

Printer's Apprentice is available from McCann Software, P.O. Box 34160, Omaha, Nebraska 68134. The price for Printer's Apprentice is $22.50, and the additional font disk is $11.50. Updates are sent free to registered owners.

Enter Extended Basic and type in the code as you see it. When the computer honks and will not accept any more code, Enter and Press Fctn 8 (REDO) and continue to Enter the rest of the code. While the File is scrolling accross the screen you can stop it by pressing ENTER. Pressing ENTER again will restart the scrolling.

The Game Room

by The TInman

```
100 REM FLIP
110 REM TI-99/4A EXTENDED BA
SIC
120 REM WESLEY R RICHARDSON
130 REM BLUEGRASS 99 COMPUTE
R SOCIETY
140 REM VARIABLES B,C(),C$,D
$,I,K,M,N,P,S
150 DIM C(9)
160 CALL CLEAR
170 DISPLAY AT(10,11):"F L I
 P"
180 DISPLAY AT(12,2):"BY WES
LEY R. RICHARDSON"
190 CALL CHAR(92,"FFFFFEFCF8
FEFEFEFEFEFEFEF8F8FFFFFFFF3F
3F3F3F3F3F3F3F3F0F0FFFFF")
!ONE
200 CALL CHAR(96,"FFFFFCF8F1
F3FFFFFFFFFCF8F0F0FFFFFFFF3F
1F8FCFCFCF8F1F3FFF0F0FFFFF")
! TWO
210 CALL CHAR(100,"FFFFFCF0F
3FFFFFFFF3F3F3F0F8FFFFFFFF3
F1F8FCFCF1F1FCFCF8F1F3FFFFF"
)! THREE
220 CALL CHAR(104,"FFFFF3F3F
3F3F3F3F0F3FFFFFFFFFFFFFFFF
FFFFFFF3F3F0F0F3F3F3F3FFFFF"
)! FOUR
230 CALL CHAR(108,"FFFFF0F0F
3F3F3F3F0F3FFFFFF0F0FFFFFFFF0
F0FFFFFFF3F1F8FCF8F1F3FFFFF"
)! FIVE
240 CALL CHAR(112,"FFFFFCF8F
3F3F3F0F0F3F3F3F8FCFFFFFFFF3
F1FCFFFFF3F1FCFCFCF1F3FFFFF"
)! SIX
250 CALL CHAR(116,"FFFFF8F8F
FFFFFFFFFF1F1F3FCFCFFFFFFFF0
F0FCFCF9F9F3F3F7F7FFFFFFFFF"
)!SEVEN
260 CALL CHAR(120,"FFFFFCF8F
3F3F3F3F8FCFCF1F3FCFFFFFFFF3
F1FCFCFCF1F1FCFCFCF1F3FFFFF"
)! EIGHT
270 CALL CHAR(124,"FFFFFCF8F
3F3F3F8FCFFFFF3F8FCFFFFFFFF3
F1FCFCFCF0F0FCFCFCF1F3FFFFF"
)! NINE
280 CALL CHAR(128,"3F1F1C1C1
C1C1F1F1C1C1C1C1C1C1C3EF8F80
8000020E0E020000000000000000"
)! F
290 CALL CHAR(132,"3E1C1C1C1
C1C1C1C1C1C1C1C1C1C1C1F3F00000
0000000000000000000808F8F8"
)! L
300 CALL CHAR(136,"070101010
10101010101010101010107F0C0C
0C0C0C0C0C0C0C0C0C0C0C0C0F0"
)! I
310 CALL CHAR(140,"3F1F1C1C1
C1C1C1F1F1C1C1C1C1C1C3EE0F01
808080818F0E000000000000000"
)! P
320 REM INITIALIZE
330 B=99999
340 CALL CLEAR
350 CALL SCREEN(6)
360 CALL MAGNIFY(4)
370 RANDOMIZE
380 CALL CHARPAT(58,C$)
390 CALL CHAR(37,C$)
400 FOR I=1 TO 3
410 CALL CHARPAT(87+I,C$)
420 CALL CHAR(39+I,C$)
430 NEXT I
440 C$="FFFFFFFFFFFFFFFF"
450 C$="0000000000000000"
460 CALL CHAR(36,C$)
470 CALL CHAR(91,C$)
480 CALL CHAR(90,D$)
490 CALL COLOR(3,2,15)
500 CALL COLOR(4,2,15)
510 CALL COLOR(8,9,15)
520 REM RESTART POINT
530 N=-1
540 FOR I=1 TO 4
550 CALL SPRITE(#(I+9),124+4
*I,12,16,32+32*I)
560 NEXT I
570 FOR I=1 TO 13
580 CALL HCHAR(I+7,3,36,13)
590 NEXT I
600 DISPLAY AT(8,15):"GOAL %
 ZZZ"
610 DISPLAY AT(9,22):"ZZZ"
620 DISPLAY AT(10,22):"ZZZ"
630 DISPLAY AT(12,15):"PRESS
% [[["
640 DISPLAY AT(13,22):"[[["
650 DISPLAY AT(14,22):"[[["
660 DISPLAY AT(16,15):"R%RES
ET Q%QUIT"
670 DISPLAY AT(18,15):"MOVES
%    0"
680 DISPLAY AT(20,15):"BEST
%"
690 GOSUB 1670
700 FOR I=1 TO 9
710 C(I)=9
720 CALL SPRITE(#I,88+4*I,9,
59+34*INT((I-1)/3),34*(I-3*I
NT((I-1)/3))-15)
730 NEXT I
740 P=1+INT(9*RND)
750 M=7
760 IF (P=1)+(P=3)+(P=7)+(P=
9)THEN M=10
770 IF (P=5)THEN M=11
780 GOSUB 1340
790 REM MAIN LOOP
800 GOSUB 1440
810 CALL KEY(0,K,S)
820 ON K GOTO 830,890,940,10
00,1050,1120,1170,1230,1280
830 REM K=1
840 P=1 :: GOSUB 1340
850 P=2 :: GOSUB 1340
860 P=4 :: GOSUB 1340
870 P=5 :: GOSUB 1340
880 GOTO 800
890 REM K=2
900 P=1 :: GOSUB 1340
910 P=2 :: GOSUB 1340
920 P=3 :: GOSUB 1340
930 GOTO 800
940 REM K=3
950 P=2 :: GOSUB 1340
960 P=3 :: GOSUB 1340
970 P=5 :: GOSUB 1340
980 P=6 :: GOSUB 1340
990 GOTO 800
1000 REM K=4
1010 P=1 :: GOSUB 1340
1020 P=4 :: GOSUB 1340
1030 P=7 :: GOSUB 1340
1040 GOTO 800
1050 REM K=5
1060 P=2 :: GOSUB 1340
1070 P=4 :: GOSUB 1340
1080 P=5 :: GOSUB 1340
1090 P=6 :: GOSUB 1340
1100 P=8 :: GOSUB 1340
1110 GOTO 800
1120 REM K=6
1130 P=3 :: GOSUB 1340
1140 P=6 :: GOSUB 1340
1150 P=9 :: GOSUB 1340
1160 GOTO 800
1170 REM K=7
1180 P=4 :: GOSUB 1340
1190 P=5 :: GOSUB 1340
1200 P=7 :: GOSUB 1340
1210 P=8 :: GOSUB 1340
1220 GOTO 800
1230 REM K=8
1240 P=7 :: GOSUB 1340
1250 P=8 :: GOSUB 1340
1260 P=9 :: GOSUB 1340
1270 GOTO 800
1280 REM K=9
1290 P=5 :: GOSUB 1340
1300 P=6 :: GOSUB 1340
1310 P=8 :: GOSUB 1340
1320 P=9 :: GOSUB 1340
1330 GOTO 800
1340 REM SET COLOR
1350 IF C(P)<>9 THEN 1400
1360 C(P)=15
1370 CALL COLOR(#P,15)
1380 CALL HCHAR(12+INT((P-1)
/3),23+P-3*INT((P-1)/3),48+P
)
1390 RETURN
1400 C(P)=9
1410 CALL COLOR(#P,9)
1420 CALL HCHAR(12+INT((P-1)
/3),23+P-3*INT((P-1)/3),91)
1430 RETURN
1440 REM CHECK FOR SOLUTION
1450 N=N+1
1460 DISPLAY AT(18,26-LEN(ST
R$(N))):N
1470 IF C(5)<>9 THEN 1660
1480 FOR I=1 TO 4
1490 IF C(I)<>15 THEN 1660
1500 NEXT I
1510 FOR I=6 TO 9
1520 IF C(I)<>15 THEN 1660
1530 NEXT I
1540 B=MIN(B,N)
1550 GOSUB 1670
1560 DISPLAY AT(22,3):"SOLUT
ION !!"
1570 DISPLAY AT(23,3):"ONLY"
;N-M;"MOVE(S)TRA MOVES"
1580 CALL SOUND(500,440,0)
1590 CALL KEY(0,K,S)
1600 IF S=0 THEN 1590
1610 IF K=81 THEN 1790
1620 IF K<>82 THEN 1590
1630 DISPLAY AT(22,1):""
1640 DISPLAY AT(23,1):""
1650 GOTO 520
1660 RETURN
1670 REM BEST SCORE
1680 DISPLAY AT(20,22):SEG$(
"     ",1,5-LEN(STR$(B)));ST
R$(B)
1690 RETURN
1700 REM WAIT FOR KEY
1710 CALL KEY(0,K,S)
1720 IF S=0 THEN 1710
1730 IF K=81 THEN 1790
1740 IF K=82 THEN 520
1750 IF (K<49)+(K>57)THEN 17
10
1760 K=K-48
1770 IF C(K)=9 THEN 1710
1780 RETURN
1790 REM QUIT
1800 CALL CHARSET
1810 END
```

FLIP FLOP

This little game of "Flip-Flop" will test your patience. When you run the program you will be presented with a square divided into nine small connected squares, one or more of which will be white and the rest red. Your job is to change the colors so that finally you have the center square red and all the rest white. You do this by pressing the number of any white square and the ajacent colors will flip from white to red or vice-versa. There is a scoring area that keeps track of the number of times you hit the keys and another area that records the shortest time for a player. I did it once in 17 tries but have not come anywhere near that on subsequent games.

COMPUTER DIAGNOSIS

```
10 RANDOMIZE
20 CALL CLEAR
40 PRINT "THIS IS A"
50 PRINT "COMPUTER DIAGNOSIS
"
60 PRINT
70 PRINT "PRESS (ENTER) AFTE
R"
80 PRINT "EACH REPLY."
90 PRINT
100 PRINT "WHAT IS YOUR CODE
 NAME?"
110 INPUT N$
120 IF N$=" " THEN 100
130 CALL CLEAR
140 PRINT "ALRIGHT, " ;N$;",
"
150 PRINT "YOUR CODE NAME IS
 ACCEPTED."
160 PRINT "NOW, TYPE A FOUR
DIGIT CODE"
170 PRINT "NUMBER, THEN PRES
S (ENTER)."
180 PRINT
190 INPUT "WHAT IS YOUR NUMB
ER?":N
200 IF N>999 THEN 210 ELSE 2
20
210 IF N<10000 THEN 260
220 PRINT
230 PRINT "INVALID NUMBER! T
RY AGAIN."
240 PRINT
250 GOTO 190
260 CALL CLEAR
270 PRINT "YOUR NAME IS: ";N
$
280 PRINT
290 PRINT "YOUR NUMBER IS: "
;N
300 PRINT
310 PRINT "IS THAT RIGHT? (Y
/N)"
320 CALL KEY(0,A,ST)
330 IF ST=0 THEN 320
340 IF A<>89 THEN 20
350 CALL CLEAR
360 PRINT "O.K., ";N$;", PRE
SS (C) TO"
370 PRINT "BEGIN A QUICK"
380 PRINT "HARDWARE CHECK."
390 CALL KEY(0,A,ST)
400 IF A<>67 THEN 390
410 CALL CLEAR
420 FOR T=1 TO INT(8*RND)+5
430 PRINT "LOCATION#";T;"= A
CCEPTABLE"
440 FOR D=1 TO 300
450 NEXT D
460 NEXT T
470 PRINT "LOCATION#";T;"= E
RROR DETECTED!"
480 PRINT
490 PRINT "PRESS (ENTER) FOR
 HELP"
500 CALL KEY(0,KEY,STATUS)
510 IF KEY<>13 THEN 500
520 CALL CLEAR
530 PRINT N$;", DIAGNOSTICS
SHOW"
540 PRINT "THAT YOUR COMPUTE
R HAS A "
550 FOR X=1 TO 27
560 READ P
570 PRINT CHR$(P);
580 FOR D=1 TO 200
590 NEXT D
600 NEXT X
610 PRINT
620 PRINT "PLEASE WAIT 15 SE
CONDS FOR A"
630 PRINT "COMPLETE DIAGNOSI
S."
640 FOR D=1 TO 700
650 NEXT D
660 CALL CLEAR
670 FOR T=15 TO 1 STEP -1
680 PRINT "TIME:";T
690 FOR D=1 TO 200
700 NEXT D
710 CALL CLEAR
720 NEXT T
730 CALL CLEAR
740 CALL SCREEN(10)
750 REM **BE SURE YOUR ALPHA
 LOCK KEY IS DEPRESSED!**
760 A$="FFFFFFFFFFFFFFFF"
770 CALL CHAR(128,A$)
780 READ A,B,R
790 IF A=-1 THEN 820
800 CALL HCHAR(A,B,128,R)
810 GOTO 780
820 READ A,B,R
830 IF A=-1 THEN 860
840 CALL VCHAR(A,B,128,R)
850 GOTO 820
860 GOTO 860
870 DATA 83,69,86,69,82,69,3
2,73,78,84,69,82,78,65,76
880 DATA 32,77,65,76,70,85,7
8,67,84,73,79,78,2,3,5,2
890 DATA 9,5,2,15,5,2,21,5,6
,3,5,6,9,5,6,15,5,8,16,1
900 DATA 9,17,1,10,18,1,11,1
9,1,11,21,5,11,27,4,14,3
910 DATA 5,14,9,5,14,15,5,14
,27,4,18,3,4,18,27,4,23,9
920 DATA 5,23,15,5,23,21,5,2
3,27,4,-1,-1,-1,2,3,10,2
930 DATA 7,10,2,9,10,2,13,5,
2,15,10,2,19,5,2,23,10,2
940 DATA 27,10,14,3,10,14,9,
10,14,13,10,14,15,10,14
950 DATA 19,10,14,21,10,14,2
7,5,18,30,5,-1,-1,-1
```

This month we present two interesting proograms that will help you in your programming. The first is "COMPUTER DIAGNOSIS". You will definitly want to type this in right away because it will give you some interesting data on your computer and how well it is functioning. The second program, "CREATE A SCREEN" fills a definte need and is a time saver for those that write programs. This program takes all the guess work out of where to display all the instructions and data that you want to have appear on the screen. When you use the program you will be presented with a blank screen with a cursor and you simply arrange where you want particular information to go. When you have finished your printer will go into action and print out instructions on where to DISPLAY AT(0,0). You can then use these instructions in your program and will have everything displayed just where you want it without a lot of trial and error.

CREATE A SCREEN

```
1 ! ********************
2 ! *  CREATE A SCREEN  *
3 ! ********************
4 !
5 ! ********************
6 ! * screen editor for *
7 ! * the 99/4A computer*
8 ! *  extended basic   *
9 ! ********************
10 !
11 ! ********************
12 ! * 21 suns  software *
13 ! * by: j. d. canning *
14 ! *   may 16, 1986    *
15 ! ********************
16 !
17 ! may be merged into
18 ! an extended  basic
19 ! program  for  use
20 ! while a program is
21 ! resident in memory
22 ! to  use  this way,
23 ! type  "RUN".   be
24 ! sure  none of your
25 ! program lines  are
26 ! 1 to 92  prior  to
27 ! merging  the file.
28 ! dim variables  may
29 ! not  permit  this
30 ! code to be merged.
31 ! check  for this if
32 ! problems   occur.
33 ! be sure cursor  is
34 ! not over  any text
35 ! when   you   press
36 ! proc'd.  you  will
37 ! lose that piece of
38 ! information.
39 !
40 ! setup run
41 !
42 OPTION BASE 1 :: DIM T(24),S$(24)
43 CALL CHAR(134,"0000000010000000"):: CALL COLOR(13,15,1):: L$=RPT$("-",28):: L1$="12345678901234567890123456789"
44 !
45 ! title
46 !
47 DISPLAY AT(7,8)ERASE ALL:"CREATE A SCREEN": :L$ :: DISPLAY AT(12,7):"21 suns software": :TAB(11);"may 1986";: :L$
48 DISPLAY AT(21,1):L$:" be sure your printer is on":L$ :: OPEN #1:"PIO",OUTPUT :: FOR I=1 TO 750 :: NEXT I
49 !
50 ! instructions
51 !
52 DISPLAY AT(2,9)ERASE ALL:"INSTRUCTIONS":L$ :: DISPLAY AT(5,5):"cursor control keys": :TAB(5);"fctn E ----- up"
53 DISPLAY AT(8,5):"fctn X ----- down":TAB(5);"fctn S ----- left":TAB(5);"fctn D ----- right"
54 DISPLAY AT(12,5):"fctn 6 ----- process":TAB(18);"screen" :: DISPLAY AT(15,5):"fctn 8 ----- start over": :L$
55 DISPLAY AT(18,3):"use  the above keys to":" move the cursor or pass":" program  control. type":" data as you desire it to"
56 DISPLAY AT(22,3):"appear on the screen.":L$:TAB(7);"*PRESS ANY KEY*"
57 CALL KEY(0,K,S):: IF S=0 THEN 57
58 !
59 ! screen
60 !
61 CALL CLEAR :: CALL SCREEN(7):: FOR R=1 TO 24 :: DISPLAY AT(R,1):RPT$(CHR$(134),28):: NEXT R
62 A$="redo to start" :: Y=7 :: Z=2 :: GOSUB 92 :: A$="procd to process" :: Y=6 :: Z=31 :: GOSUB 92 :: R,C=1
63 DISPLAY AT(R,C)SIZE(1):CHR$(30):: CALL KEY(0,K,S):: IF S=0 THEN 63
64 IF K<6 OR K>13 THEN 73
65 DISPLAY AT(R,C)SIZE(1):CHR$(134)
66 ON K-5 GOTO 67,63,68,69,70,71,75,72
67 CALL SOUND(200,1400,0):: GOTO 61
68 C=C-1 :: DISPLAY AT(R,C)SIZE(1):CHR$(30):: GOTO 63
69 C=C+1 :: DISPLAY AT(R,C)SIZE(1):CHR$(30):: GOTO 63
70 R=R+1 :: DISPLAY AT(R,C)SIZE(1):CHR$(30):: GOTO 63
71 R=R-1 :: DISPLAY AT(R,C)SIZE(1):CHR$(30):: GOTO 63
72 DISPLAY AT(R,C)SIZE(1):CHR$(134):: R=R+1 :: C=1 :: GOTO 63
73 DISPLAY AT(R,C)SIZE(1):CHR$(K):: C=C+1 :: IF C=29 THEN R=R+1 :: C=1
74 GOTO 63
75 A$="one moment please" :: Y=5 :: Z=2 :: GOSUB 92 :: A$=" reading screen  " :: Y=5 :: Z=31 :: GOSUB 92 :: CALL SOUND(150,-1,0)
76 FOR C=1 TO 24 :: S$(C)="" :: NEXT C
77 FOR R=1 TO 24 :: C=1
78 CALL GCHAR(R,C+2,K):: IF K=30 OR K=134 AND C<28 THEN C=C+1 :: GOTO 78
79 IF K=30 OR K=134 AND C=28 THEN T(R)=0 :: GOTO 81
80 T(R)=C :: FOR I=C TO 28 :: CALL GCHAR(R,I+2,K):: S$(R)=S$(R)&CHR$(K):: NEXT I
81 NEXT R
82 !
83 ! output to printer
84 !
85 CALL CLEAR :: CALL SCREEN(9)
86 PRINT #1:TAB(19);"ROW  COLUMN";TAB(43);"PHRASE":TAB(33);L1$:""
87 FOR R=1 TO 24 :: PRINT #1:TAB(19);R;TAB(26);T(R);TAB(33);S$(R):: NEXT R
88 PRINT #1:TAB(33);L1$ :: PRINT #1:"":"" :: GOTO 61
89 !
90 ! dsv subroutine
91 !
92 FOR X=1 TO LEN(A$):: CALL VCHAR(Y-1+X,Z,ASC(SEG$(A$,X,1))):: NEXT X :: RETURN
```

NEW SECTION!

Hey all you gamers! Now's your time to show off those high scores you've hidden for all these years! A new high score section is being added to WORDPLAY. Simply call in your score for one of the ten games listed below, and if it is the HIGHEST high score for the month, you'll see it (with your name or alias) in the GAME ROOM. Here are the games available for entry in the Game Room:

1. JUMPY
2. PARSEC
3. MUNCHMAN
4. TI-RUNNER
5. TI INVADERS
6. BARRAGE
7. LASSO
8. SLYMOIDS
9. CENTIPEDE
10. CAR WARS

More games may be added later as desired. Call Dan Hawes at 620-9725 for more information.

TI-WRITER PROBLEMS

PUNCTUATION

Typists will always use two spaces after a punctuation mark ending a sentence. TI-Writer, for some strange reason, does things a little different.

For example:

"." The period(.) - TI-Writer will always put 2 spaces after every period that has been *followed by* a single space. This is fine if the period is at the end of the sentence, but what if you are using an abbreviation within a sentence? The formatter will put 2 spaces here also, but you properly only want one. What you need to do in this case is use the required space symbol (^) after the period of an abbreviation. This will give you the desired one space when using the formatter. (A period followed by no space will appear as just that.)

The exclamation and question marks (!) (?) - In these cases the formatter will not automatically give two spaces as it properly should. To make your document look correct you will need to add one space and one required symbol (^).

THE PERIOD AND DECIMALS

The formatter thinks that any line which begins with a period is a formatter command and will delete the whole line. If by chance your document contains a value such as .10 and the wraparound caused by Fill and Adjust of the formatter puts it at the beginning of the line, the whole line will disappear. To correct this you could put a zero in front of your decimals (0.10).

ASTERISK AND NUMBERS

If you are printing out of the formatter and your document contains an asterisk followed by two or more numeric digits, the asterisk and the two digits will disappear. For instance, A*256 becomes A6. What's happening here is that the TI-Writer program misinterprets the asterisk and two digits as an instruction to input data from a "value file", as in mail merge. This is described on page 111 of the TI Instruction book. To correct this problem you will need to type two asterisks followed by two dummy numbers, then the actual digits. For example, type A**25256 instead of A*256.

REQUIRED SPACE

If you tie words together for the purpose of underlining (&) or overstriking (@) with the required space (^), the Fill and Adjust of the formatter will leave gaping blanks in your lines. If you tie too many together, the line will extend beyond the right margin. It would be better to put a seperate & or @ in front of each word. Be sure to include the spaces between the words. If you want a (^) to appear in your text, you will need to transliterate it. (@) and (&) are typed twice in succession to get them to print.

OTHER PROBLEMS

Other problems have been noted in TI-Writer that cause erratic and destructive commands, but they are not fully documented.

MULTIPLAN DISK DRIVE DEFAULT

I have been using Multiplan almost as long as I have had my TI 99-4/A and the more I use it the more features I find about it.

I think that even if the amount of storage is not as great as Multiplan provides on other computers with more memory that it still is a great spreadsheet.

One feature that was needed I thought was a simpler way to designate the data output to a different disk drive. Until I got my additional drives this was not a problem. One drive simply required disk swapping.

When I finally installed drives #2 and #3 I found that I needed to call up the (T)ransfer (O)ptions to have the program automatically save all data to a drive other than DSK.1. This needed to be done each time I entered Multiplan to view or review a particular spreadsheet.

It was called to my attention that the machine language that Multiplan uses to designate the DSK1. is located on the second sector of the MPINTR file on the Multiplan disk.

You can change the default drive from DSK1. to DSK2. (or whatever) by changing the hex digits that appear on this sector.

I used Advanced Diagnostics, but any disk sector editor should do the same thing. Using Advanced Diagnostics I first used FF (Find File) to locate where MPINTR resided on the disk. It started on sector 271 so I then used ES 272 (Edit Sector) to change the code and finally WS (Write Sector) to write the changed code to the Multiplan Disk.

The code to change is 4B31. Change it to 4B32 or 3 or 4 depending on what disk drive you want the default. Using Advanced Diagnostics you will find this code begins on row 7, column 9.

I suggest you do all this on a copy disk in case something goes wrong. It worked for me the first time and it really is a time saver when you first call up Multiplan.

Charles Ball - Editor

CLUB VISITOR

We have had a letter from Mike Shayne who was a visitor from Pittsburgh at our March meeting.

He sends us a disk full of "*goodies*" that will be added to the club library.

It is encouraging to see visitors from other clubs attend our meetings. This visit from Mike was a direct result of our ad in the Sunday paper that precedes our Tuesday meetings.

FILE PROCESSING

File processing on the TI is not as difficult as you might believe. The hardest part for me was figuring out the "examples" that were in the owner's manual. They all went something like this:

```
100 OPEN #2:"CS1",INTERNAL,INPUT,FIXED
-
-
- program lines
-
-
290 CLOSE #2
300 END
```

This, in my opinion falls under the heading of "poor documentation". What was left out was the most important part! I tried and tried to get my computer to process files. I failed because I didn't know what to tell the computer to do with the files once it was open. I couldn't get past the mental block that told me "file processing is different from programming". In fact, programming is just a form of file processing.

The TI 99/4A handles ALL input and output through files. Most of the time, we are completely unaware that we are dealing with a "file" while programming. Page II-119 of the User's Reference Guide states "ALL TI BASIC statements which refer to files do so by means of a file number between 0 and 255 inclusive," "...file number 0 refers to the keyboard and screen of your computer and is always accessible...". Since file 0 is always accessible, statements such as PRINT, INPUT, RESTORE, etc. which refer to the keyboard or screen do require a file number with them. You can however, write a statement such as:

```
100 PRINT #0:"print this to screen"
```

and have it do exactly the same thing as:

```
100 PRINT "print this to screen"
```

You can also INPUT from file #0, but since file 0 is always open, statements like OPEN #0 or CLOSE #0 will generate an error message.

All other open files must be referred to by their number. Remember that this number is only used by the program to remember which file is which and is not a part of the file at all. As a matter of fact, you could open a file with one number, process it somehow, close it, and then reopen the same file with a different number...all this within the same program!

Now that I've got you thoroughly confused, I'll give you a short sample file processing program to try to clarify what I've been saying. Most of us think of a file as being a disk or cassette. While these are indeed files to the computer, they are by no means the only ones we have available. This short program opens a file to the Speech Synthesizer, sets up a FOR-NEXT loop to print a couple of sentences to both the screen and the Synthesizer, and then closes the file. You will need a TE-2 module to run the program. If you don't have a TE-2, just change the file name in line 110 from SPEECH to PIO or whatever your printer requires. This will give output to the screen and the printer instead.

```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
120 FOR Y=1 TO 7
130 READ X$
140 FOR X=0 TO 1
150 PRINT #X:X$
160 NEXT X
170 NEXT Y
180 CLOSE #1
190 DATA THIS IS A TEST OF
THE SCREEN AND SPEECH FILES ON THE
200 DATA TEXAS INSTRUMENTS
99/4A HOME COMPUTER.  IT
SHOULD HELP
210 DATA TO DEMONSTRATE HOW
ALL INPUT AND OUTPUT IS
TREATED AS A FILE BY THE
COMPUTER
```

In this program, line 110 OPENs a file to the speech synthesizer (or printer). Lines 120 to 140 set up some loops to read from the DATA statements and switch between files (0 and 1). Line 150 PRINTs the output to both outputs (0 and 1). Lines 160 and 170 increment the loops. Line 180 CLOSEs the computer's association with file #1, and lines 190 to 210 are the DATA read by line 130.

The point is that the lines between 110 and 180 are the ones that do all the work. Whether you are working with a file or just printing to the screen, the programming is the same. All you have to do is tell the computer where you want the data to go to or to come from.

Try modifying line 110 from OPEN #1:"SPEECH",OUTPUT to OPEN #1:"DSK1.TESTFILE",OUTPUT. This will cause the second output (remember that #0 is going to the screen) to go to a disk in drive #1 under the filename of "TESTFILE". Try some other experiments in line 110 like using "CS1", "PIO", or "RS232" instead of "SPEECH". These will cause the output to go to the cassette recorder, printer, or modem respectively in addition to the screen.

Once you have mastered OUTPUTing to peripheral devices, the next logical step is to learn how to get INPUT from them. Some devices, such as the printer or speech synthesizer, by their very nature are one-way devices. Trying to get input from them would surely lead to hours of frustration. Keeping that in mind, we will concentrate on the devices that have two-way communication with the computer. The disk drive and cassette recorder are the primary devices we use for file storage. My experience with cassette based files has left me somewhat dissatisfied. While there are provisions for storing SEQUENTIAL files on cassette, it is a cumbersome operation as best. There also seems to be a bug in the I/O routines for input from cassette. If you do any file storage and retrieval from cassette, keep in mind that the delay between the prompt:

```
*PRESS CASSETTE PLAY CS1
 THEN PRESS ENTER
```

and the actual reading of data is longer in most cases than the tone leading to the data. I have found that if I press ENTER first, then wait for the screen to scroll up 1 line before pressing cassette play that I have no problems. If you don't do this the computer may miss the beginning of the file and give an error.

Since getting input from cassette and disk is very similar, I won't spend any more time on cassettes.

Getting input from a disk file is almost the same as sending output to it. First, you have to OPEN the file to the disk. This is done exactly the same as before, except instead of "OUTPUT" following the file name, we use "INPUT". The words INPUT and OUTPUT are two of the 4 modes that can be used to open a file. The third, UPDATE, is the default and means you can either read from it or write to it. If you don't specify one of the 4 modes, UPDATE will be assumed by the computer. The last mode is called APPEND and will only allow OUTPUT to the end of a file. Let's look at our program again. If you haven't already done so, change line 110 to OPEN #1:"DSK1.TESTFILE",OUTPUT and run the program. Type in the new program below (or modify the old one to match).

```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
115 OPEN #2:"DSK1.TESTFILE",INPUT
120 FOR Y=1 TO 7
130 INPUT #2:X$
140 FOR X=0 TO 1
150 PRINT #X:X$
160 NEXT X
170 NEXT Y
180 CLOSE #1
190 CLOSE #2
```

The main differences between this program and the first one are that we have added a second file number and name to the program (line 115), changed the "READ X$ to "INPUT #2:X$, and deleted the data statements at the end of the program. We are now getting the data from the disk file that we just saved under the name of "TESTFILE", and #0 means the keyboard and screen. File #0 is an "UPDATE" file, #1 is an "OUTPUT" type file and #2 is an "INPUT" type file.

This has been very basic stuff so far, but in order to learn "FILE PROCESSING", you must understand the basics of how your TI-99/4A computer communicates with it's peripherals. Once you figure out that the computer treats EVERYTHING as a file, you will be on your way to writing your own file processing software.

(SNUGLETter - December 1986)

P.O. BOX 15037
EAST PORTLAND STATION
PORTLAND, OR 97215

ALL GENERAL MEETINGS ARE HELD ON THE FIRST TUESDAY OF EACH MONTH, AT THE PGE BUILDING 3700 SE 17TH. PORTLAND, OR

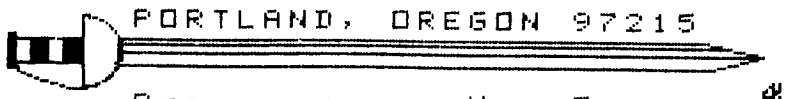
!! NEXT MEETING DATE !!

APRIL 7, 1987

THE **PUNN** NEWSLETTER

Wordplay

P.O.BOX 15037
PORTLAND, OREGON 97215



POINTING THE WAY FOR USERS OF TI's 99/4 COMPUTERS

APRIL 1987-VOLUMN VI-ISSUE 4