Ø26

8808

WORDPLAY

The PUNN Newsletter - Portland, Oreson

August 1988

What's Inside

Club Officers

President	Al Kinney	640-5860
Vice-President	Dale Kirkwood	645-4117
Secretary	Don Barker	223-1749
Treasurer	Chuck Neal	642-7292

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	Jim Smith	233-0273
Membership Chairman	Terry Priest	649-9583
Newsletter Editor	Charles Ball	639-0466

16576 SW Matador Lane-King City, OR 97224

BBS Committee

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

BBS Phone Number 503/233-6804

From the President

As summer gets into full swing, it gets harder to sit down in front of the computer, until the sun sets and things start to cool off. By then my eyelids get heavy and the good plans I had earlier seem to fade into TOMORROW...

But, just around the corner, in August is our Second Annual PUNN Picnic. PUNN Members and guests still have time to get their tickets (you can still get'em at the door!), and we'll have plenty of food and good times. See elsewhere in this issue for details of when, where and how much. Don't miss out on this, it was fun last year and should be even better this year.

There are some outstanding new software packages in recent release. One of them, TELCO, was featured in our last meeting, and seems to be rapidly establishing itself as the premier comm program for the TI. At only $20, it is a bargain in anyone's book. Another, I've only heard about, is a new database program that mimics dBASE II of the IBM world. While not a fairware program, it only sells for $20. There have been several new updates for DM1000, but unfortunately we have been told any version later than 3.5 has a possible SERIOUS error in it, so you might want to use it with caution.

Speaking of communications, if you have been on vacation or missed the last couple of meeting, you'll want to RUN, not walk to this one. Mike Caulkins, our Hard Copy Librarian, rounded up an outstanding bargain on some Prometheus 2400 bps modems, and a number of us took advantage of our group purchasing power to get one for only $151.50, delivered. The Users Group also purchased three additional modems to be sold on a first-come, first-served basis to members. See our hard-working Treasurer, Chuck Neal to get yours. If there are other group purchases we can do, let us know.

Al Kinney

News and Views

There will be no regular meeting on August 2nd as this date has been reserved for the ANNUAL PICNIC!- If you have not already done so, you can order your tickets from Don Barker-call him at 223-1749-Tickets will be available at the Milwaukie Elks Club Picnic Grounds however it will be helpful in planning the food requirements if you purchase your tickets in advance- - -A very interesting program is planned for the September 6th meeting-Paul Coleman, one of our members, and a talenteo programmer will be here to demonstrate some of his latest programs-you'll not want to miss this- - -Jim Thomas reminds that he has the Music Library from Amion-20 disks full of great music programming-order them at the Library- - -Ron Mayer reports that Wes Richardson from the Blue Grass Users Group, Lexington, KY was a guest of his earlier in July-He brought with him a number of programs that are being cataloged and will be available to members- - -Chuck Neal reports a balance of $1795.53 in the club treasury- - -Sales of the GENIAL TRAVelOR disks have contributed some $70.00 to our funds-these disks

(continued on page 3)

Maze Challenge

This little maze game should be fun for kids and adults alike and could provide an evenings entertainment. After you type it in, it'll take a couple of minutes to display on the screen. (I wonder if prescan would improve this?)

The display will show the heading across the top indicating what you score is and the correct answer can be had by hitting the appropriate key.

You need to move through the maze using the FCTN arrow keys and of course a little study is needed to select the correct route. A plus sign is the starting point and an "F" at the bottom is the finish. Each move is counted and if you have to backtrack those moves count too. The computer keeps a running count of your moves and you can compare this against the number at the top, which is the shortest route from the start to the finish.

When you have finished the computer will signal.

You then have three choices, Replay (try to find a better route(, play a different maze or ask for the computer path (the correct route). You can load this game into either BASIC OR XBASIC. It's rather difficult to beat the computer.

I'm not quite sure where this came from. Found it among some game disks I had accumulated over the years. It doesn't seem to have a print out, but perhaps a screen dump would do that job.

If anyone out there could improve this program with a printout, I would like to hear from you.

(Charles Ball, Editor)

Error Handling

TI Extended Basic gives you the option of including your own error handling routines in your programs. A common problem you may encounter is while doing multiple RETURNS. When your basic program branches to a subroutine from a GOSUB, it places the return address in a buffer called the Stack. The number of return addresses that can be placed in the Stack is limited. If you place too many return addresses in the Stack, you will get a 'Memory Full' error. To avoid this error, you must have the program return from any subroutine with a GOSUB.

A problem arises when your program branches to any error routine because of the ON ERROR statement. Returns will remain pending in the Stack. This could happen at almost any point in a program. To remedy this you can simply exit an error routine with a RETURN. This method returns you to the line that caused the error.

An alternative would be to use the RETURN NEXT statement to return the program to the line immediately following the one that caused the error. This method works in many cases but if an error is likely to repeat indefinitely, another solution is needed.

A third solution is to use RETURN (line number). When you specify a line number after the RETURN statement, the program execution branches to that line number. This gives absolute control over error routine exiting, however returning in this manner pops only one return address off the Stack. If the line on which the error occured is embedded deeper than the line you return to, then return addresses remain on the Stack. This may result in a Memory Full message or the program may return to an unpredictable location.

There is one way to clear the return stack, but it should be used with caution. If not done properly you may end up with a RETURN WITHOUT GOSUB error message. After clearing the return stack, you must return to a location in the program without a pending RETURN. If you should branch back to a subroutine, its RETURN will cause an error message since the return Stack has already been cleared.

By having the RETURN (line number) statement return to itself, you create a loop that repeats until the stack is empty and a RETURN WITHOUT GOSUB error is generated. By trapping this new error, you can branch to a non-subroutine location in your program with a cleared Stack.

The program listed here demonstrates this procedure. Line 100 ON ERROR sets the initial error trapping for the main body of the program. Lines 140, 150, and 160 place 3 return addresses on the GOSUB stack. An intentional misspelling on line 170 now causes a syntax error and sends the program to line 190 (the error routine). Line 190 prints a message letting you know that it has entered the error routine. Line 200 sets up a new line to branch to in case of an error. Line 210 loops back to itself continously, taking return addresses off the Stack and then ON ERRORs to line 220. Line 200 simply contains a time delay and then proceeds back to the program start. Lines 200 to 220 can be lifted and adapted to most any error routine.

```
100 ON ERROR 190
110 CALL CLEAR
120 A=A+1
130 PRINT A
140 GOSUB 150
150 GOSUB 160
160 GOSUB 170
170 FRIT"PRINT IS MISSPELLED
."
180 PRINT "IT WON'T MAKE IT
HERE. "
190 PRINT "THIS IS THE ERROR
 TRAPPING ROUTINE.  IT WILL
POP THE":" PENDING RETURNS O
FF THE":" STACK AND GO TO TH
E START OF THE PROGRAM."
200 ON ERROR 220 ____
210 RETURN 210
220 FOR DELAY=1 TO 500 :: NE
XT DELAY :: GOTO 100
```

```
***********************************  ********************************
*                                 *  *                              *
*    There is plenty of money around.  *  *  MURPHY'S RULE:              *
*       The trouble is that everyone   *  *      All Warranties Expire     x
*      owes it to somebody else.      *  *     Upon Payment of Invoice    *
*                                 *  *                              *
***********************************  ********************************
```

Listing for Super Maze

```
100 REM **************
110 REM * SUPER MAZE *
120 REM **************
130 REM
140 GOSUB 340
150 GOSUB 1300
160 GOSUB 600
170 GOSUB 1670
180 MSG$="0131"&STR$(ANSW)
190 GOSUB 2750
200 GOTO 230
210 CALL KEY(3,KY,ST)
220 IF ST=0 THEN 210
230 IF KY=65 THEN 270
240 IF KY=78 THEN 140
250 IF KY=82 THEN 310
260 GOTO 210
270 CALL CHAR(144,"00003C3C0
000FFFF")
280 CALL CHAR(145,"00003C3C"
)
290 CALL COLOR(15,16,11)
300 GOTO 210
310 GOTO 1150
320 GOTO 170
330 REM SET INITIAL VAR
340 CALL CLEAR
350 DIM MZ1(22,30)
360 FOR I=1 TO 15
370 CALL COLOR(I,4,4)
380 NEXT I
390 DATA 128,808080808080FFF
F,129,000000000000FFFF
400 DATA 130,800080808080808
0,131,0
410 DATA 132,0010107C1010FFF
F,133,009090FC9090FFFF
420 DATA 134,0010107C101,135
,809[illegible]909(
430 DATA 136,FFFFFFFFFFFFFF
F,144,000000000000FFFF
440 DATA 145,0
450 RESTORE 390
460 FOR I=1 TO 11
470 READ A,B$
480 CALL CHAR(A,B$)
490 NEXT I
500 KMOV=-1
510 ANSW=1
520 KMOV=0
530 FOR I=1 TO 22
540 FOR J=1 TO 30
550 MZ1(I,J)=0
560 NEXT J
570 NEXT I
580 RETURN
590 REM CREATES MAZE
600 RANDOMIZE
610 FOR J=1 TO 22
620 FOR I=1 TO 29 STEP 7
630 FOR K=130 TO 131
640 N=INT(((I+7)-I+1)*RND)+I
650 IF N>30 THEN 700
660 X=MZ1(J,N)
670 IF X=130 THEN 640
680 IF (X=144)+(X=145)THEN 7
00
690 MZ1(J,N)=K
700 NEXT K
710 NEXT I
720 NEXT J
730 FOR I=1 TO 22
740 FOR J=1 TO 30
750 X=MZ1(I,J)
760 IF (X=144)+(X=145)THEN 8
80
770 IF X=0 THEN 860
780 IF (X=128)+(X=130)THEN 8
90
790 IF I<13 THEN 890
800 IF (X=131)+(X1=1)THEN 85
0
810 X1=1
820 IF J-1<1 THEN 850
830 IF MZ1(I,J-1)=145 THEN 8
60
840 GOTO 890
850 X1=0
860 MZ1(I,J)=129
870 GOTO 890
880 [illegible]=[illegible]+1
890 CALL HCHAR(I+1,J+1,MZ1(I
,J))
900 NEXT J
910 NEXT I
920 CALL VCHAR(1,1,136,24)
930 CALL VCHAR(1,32,136,24)
940 CALL HCHAR(1,1,136,32)
950 CALL HCHAR(24,1,136,32)
960 MSG$="[illegible]SCORE: TI-"&ST
R$([illegible]
970 MSG$=MSG$&" KEYS-    AN
S-"
980 GOSUB 2750
990 MSG$="24[illegible] REPLAY    A-
ANSWER    N-NEW"
1000 GOSUB 2750
1010 FOR I=1 TO 12
1020 CALL COLOR(I,16,7)
1030 NEXT I
1040 CALL COLOR(14,7,7)
1050 CALL COLOR(13,16,4)
1060 CALL COLOR(15,16,4)
1070 CALL GCHAR(SR,SC,NR)
1080 IF NR=144 THEN 1110
1090 CALL HCHAR(SR,SC,134)
1100 GOTO 1120
1110 CALL HCHAR(SR,SC,132)
1120 CALL HCHAR(FR,FC,70)
1130 RETURN
1140 REM REPLAY
1150 FOR I=1 TO 15
1160 CALL COLOR(I,4,4)
1170 NEXT I
1180 CALL CHAR(144,"00000000
0000FFFF")
1190 CALL CHAR(145,"0")
1200 FOR I=1 TO 22
1210 FOR J=1 TO 30
1220 CALL HCHAR(I+1,J+1,MZ1(
I,J))
1230 NEXT J
1240 NEXT I
1250 ANSW=1
1260 KMOV=0
1270 GOSUB 920
1280 RETURN
1290 REM CREATES SOLUTION
1300 CALL CLEAR
1310 RANDOMIZE
1320 J=1
1330 K=INT(11*RND)+10
1340 SR=2
1350 SC=K+1
1360 L=INT(2*RND)+1
1370 IF J+1=23 THEN 1630
1380 IF J+1+L>22 THEN 1390 E
LSE 1400
1390 L=22-J
1400 FOR I=J TO J+L-1
1410 MZ1(I,K)=145
1420 NEXT I
1430 MZ1(I,K)=144
1440 J=J+L
1450 L=INT(10*RND)+1
1460 D=INT(2*RND)+1
1470 IF D=1 THEN 1550
1480 IF K+L>30 THEN 1550
1490 FOR G=K TO K+L-1
1500 MZ1(J,G)=144
1510 NEXT G
1520 MZ1(J,G)=145
1530 K=K+L
1540 GOTO 1360
1550 IF K-L<2 THEN 1480
1560 FOR G=K TO K-L+1 STEP -
1
1570 MZ1(J,G)=144
1580 NEXT G
1590 MZ1(J,G)=145
1600 K=K-L
1610 GOTO 1360
1620 MZ1(22,K+1)=144
1630 FR=23
1640 FC=K+1
1650 RETURN
1660 REM CALL KEY MOVEMENT
1670 R=SR
1680 C=SC
1690 CALL KEY(3,KY,ST)
1700 IF ST=0 THEN 1690
1710 KMOV=KMOV+1
1720 MSG$=STR$(KMOV)
1730 IF KMOV<10 THEN 1770
1740 IF KMOV<100 THEN 1760
1750 CALL HCHAR(1,23,ASC(SEG
$(MSG$,3,1)))
1760 CALL HCHAR(1,22,ASC(SEG
$(MSG$,2,1)))
1770 CALL HCHAR(1,21,ASC(SEG
$(MSG$,1,1)))
1780 CALL SOUND(10,500,0)
1790 REM CHECK UP
1800 IF KY<>69 THEN 1990
1810 CALL GCHAR(R-1,C,NR)
1820 IF NR=70 THEN 32767
1830 IF (NR=130)+(NR=131)THE
N 1860
1840 IF NR=145 THEN 1860
1850 IF (NR=134)+(NR=135)THE
N 1890 ELSE 1690
1860 R=R-1
1870 IF NR=130 THEN 1930
1880 IF (NR=131)+(NR=145 THE
N 1950
1890 CALL HCHAR(R,C,MZ1(R-1,
C-1))
1900 R=R-1
1910 ANSW=ANSW-1
1920 GOTO 1690
1930 CALL HCHAR(R,C,135)
1940 GOTO 1960
1950 CALL HCHAR(R,C,134)
1960 ANSW=ANSW+1
1970 GOTO 1690
1980 REM DOWN
1990 IF KY<>88 THEN 2240
2000 CALL GCHAR(R,C,NR)
2010 IF (NR=132)+(NR=133)THE
N 1690
2020 CALL GCHAR(R+1,C,NR1)
2030 IF (NR1=136)+(NR1<128)T
HEN 1690
2040 R=R+1
2050 CALL GCHAR(R,C,NR)
2060 IF NR=70 THEN 2690
2070 IF (NR=144)+(NR=129)THE
N 2140
2080 IF (NR=145)+(NR=131)THE
N 2180
2090 IF NR=128 THEN 2160
2100 IF NR=130 THEN 2200
2110 CALL HCHAR(R-1,C,MZ1(R-
2,C-1))
2120 ANSW=ANSW-1
2130 GOTO 1690
2140 CALL HCHAR(R,C,132)
2150 GOTO 2210
2160 CALL HCHAR(R,C,133)
2170 GOTO 2210
2180 CALL HCHAR(R,C,134)
2190 GOTO 2210
2200 CALL HCHAR(R,C,135)
2210 ANSW=ANSW+1
2220 GOTO 1690
2230 REM RIGHT
2240 IF KY<>68 THEN 2420
2250 CALL GCHAR(R,C+1,NR)
2260 IF (NR=128)+(NR=130)THE
N 1690
2270 IF (NR=133)+(NR=135)THE
N 1690
2280 IF NR=136 THEN 1690
2290 C=C+1
2300 IF NR=70 THEN 2690
2310 IF (NR=144)+(NR=129)THE
N 2360
2320 IF (NR=145)+(NR=131)THE
N 2380
2330 CALL HCHAR(R,C-1,MZ1(R-
1,C-2))
2340 ANSW=ANSW-1
2350 GOTO 1690
2360 CALL HCHAR(R,C,132)
2370 GOTO [illegible]
2380 CALL HCHAR(R,C,134)
2390 ANSW=ANSW+1
2400 GOTO 1690
2410 REM LEFT
2420 IF KY<>83 THEN 2670
2430 CALL GCHAR(R,C,NR)
2440 IF (NR=133)+(NR=135)THE
N 1690
2450 IF (NR=128)+(NR=130)THE
N 1690
2460 CALL GCHAR(R,C-1,NR1)
2470 IF NR1=136 THEN 1690
2480 C=C-1
2490 CALL GCHAR(R,C,NR)
2500 IF NR=70 THEN 2690
2510 IF (NR=144)+(NR=129)THE
N 2580
2520 IF (NR=145)+(NR=131)THE
N 2620
2530 IF NR=128 THEN 2600
2540 IF NR=130 THEN 2640
2550 CALL HCHAR(R,C+1,MZ1(R-
1,C))
2560 ANSW=ANSW-1
2570 GOTO 1690
2580 CALL HCHAR(R,C,132)
2590 GOTO 2650
2600 CALL HCHAR(R,C,133)
2610 GOTO 2650
2620 CALL HCHAR(R,C,134)
2630 GOTO 2650
2640 CALL HCHAR(R,C,135)
2650 ANSW=ANSW+1
2660 GOTO 1690
2670 IF (KY=65)+(KY=78)THEN
2730
2680 IF KY=82 THEN 2730 ELSE
 1690
2690 FOR I=1 TO 10
2700 CALL SOUND(80,900,0)
2710 CALL SOUND(80,600,0)
2720 NEXT I
2730 RETURN
2740 REM MSG PRINT
2750 FOR I=1 TO LEN(MSG$)-4
2760 LTR=ASC(SEG$(MSG$,I+4,1
))
2770 ROW=VAL(SEG$(MSG$,1,2))
2780 COL=VAL(SEG$(MSG$,3,2))
2790 CALL HCHAR(ROW,COL+I-2,
LTR)
2800 NEXT I
2810 RETURN
```

(continued from page 1)

are still available for $6.00 per issue (each issue consists of two single sided-single density disks)-your purchase of these and other items from the library help to keep our finances in good order- - -Ashley Reed, Don Steffan and John Usher volunteered to de-bug some programs for the Wordplay last month and these programs will appear in a future edition of our newsletter- - -At future meetings interesting programs from other User Groups will be available for de-bugging or typing in-you may volunteer (or assigned!) one of these tasks- - -When you see the editor approaching you with a sheet in hand don't be too surprised-this will be your chance for recognition- - -Don Barker reminds us that the Elks Swim- ming Pool is available for all ($1.25) when you come to the Picnic- - -This is your newsletter-let us know what you want to appear in it-we have a wealth of material to select from but we need to know what YOU want- - -

- -the editor.

Stand Up and Say It

(This month we're going to put to use some of the good things we've learned about "Grammar and All That" - how to use it in front of an audience.)

There is no getting around the fact that public speaking is a reward or a burden that sooner or later most all of us will have to deal with. If a businessman is not invited to speak or at least to chair a meeting, he has not arrived and probably never will.

There are many organizations today that can help you on the way to becoming a successful speaker-The Dale Carnegie Course; Toastmasters; and other local groups that show you how to get up in front of a group and tell your story.

A lot of people seem to have a natural talent to speak extemporaneously and do it well, although they must constantly fight the temptation to go on and on, beyond the endurance of their listeners-a temptation, we regret to say, that is often too strong to resist. Most of us, however, prefer to depend upon a text or comprehensive notes, in case we forget what we're up in front of the audience for in the first place. Memorizing a speech takes too much time and reading a speech from a complete text can be deadly to the audience.

The best rule to follow is to be an "expert". In other words never get up to speak about a subject unless you know what you talking about. We're all an expert about one or more things and this is the only thing you should ever get up and talk about. If you do this, and prepare, you'll be on your way to a good speaker.

Jokes should be avoided or very short and slanted to your subject. You run the risk of a joke falling flat, a disaster from which you can never recover, no matter how brilliant you subsequent oratorical efforts. A little humor is OK but remember an audience always relishes a small anecdote that shows the speaker at a disadvantage.

One of the secrets of good speech-making is simple: make your talk shorter than your audience expects. A short talk, no matter how dull, is never a disaster but an oratorical masterpiece that exceeds a reasonable length of time is unforgivable.

And a final word to the Chairman or Master of Ceremonies. Introduce your speakers briefly and shut up. Nobody came to hear you. At best you are a necessary evil, and your success will be in reverse proportion to the amount of time your mouth is open.

(Charles Ball, Editor)

Partial Files-TI Writer

Saving and merging PARTIAL FILES is a useful feature in TI WRITER. Portions of previously written letters and documents can be incorportated into new files without retyping the text over again. If you have gone to the trouble to prepare an involved letterhead, you can easily merge it into the beginning of subsequent letters. If you want to take out a paragraph from a letter, but possible want to use it later, you can save it as a seperate file that can be expanded or merged later. If you write to a variety of family and friends you can write a long letter and then combine the paragraphs selectively to your various individuals. Thus each letter will be unique and combine the apporpriate paragraphs dedicated to that particular person.

If you are going to do a lot of work with partial files it's convenient to have a hard copy printed with line numbers. You can do this printing from the Editor. Normally when a printing from the Editor you would command PF <ENTER> PIO <ENTER>. To get the line numbers you would insert an L in front of the PIO (PF <ENTER> L PIO <ENTER>. Of course you need to remember that if your printer only prints 80 columns that the line numbers and spaces take up 6 spaces so your text should not exceed 74 characters if you want to print the line numbers. If you print out in elite or condensed this number can be increased accordingly.

Once you have a printout with line numbers, it's easy to save or merge a partial file. If you wanted to save lines 11 through 23, you would key SF <ENTER> 11 <SPACE> 23 <SPACE> DSK1.(filename) <ENTER>. But what if you don't want an entire line? You can do this by moving the cursor in front of the part of the line you want to save and press INSERT CHARACTER (FCTN 2). This will break your sentence in two, causing the portion of the line split to be placed on the next line. This would shift the document down 1 line so in the above example you would then save lines 12 through 24 instead of lines 11 through 23. If you don't want everything on line 24, split it in the same manner. If you plan to split a lot of lines you should do so before you make your printout with line numbers.

You can merge entire files into your documents or just part of a file can be added. If you want to merge an entire file, first determine where you want it to go. For instance if you want to print a file after line 14 you would key LF <ENTER> 14 <SPACE> DSK1.(filename) <ENTER>. This will insert the document called (filename) between lines 14 and 15 of your present document. If you only want to merge lines 7-15 of (filename) between lines 14 and 15 you would key LF <ENTER> 14 <SPACE> 7 <SPACE> 15 <SPACE> DSK1.(filename).

After you have merged your file, you can use REFORMAT (CTRL 2) to eliminate extra spaces and merge all of the text into a single paragraph.

This feature of TI Writer makes it a value tool that ranks with the more expensive word processing programs.

PUNN Picnic -- August 2nd!

Hi-Res Graphics Part V

(Fifth in series by Ann Dhein. This series reviews the various drawing programs that allow the user to create graphics by turning on (or off) the smallest addressable unit of the screen-a single pixel.)

Four of the programs have the ability to magnify a small part of the picture you are working on so that it temporarily fills the screen. This allows you to work on small details with a high degree of accuracy. Joy Paint call this mode "fatpixel", Paint 'N Print calls it "magnify". TI Artist and Graphx call it "zoom". Joy Paint also displays a normal sized version of the graphics in the upper right corner of the screen so that you can see what your changes are going to look like as you make them. TI Artist lets you use other drawing functions while in the zoom mode - even to making a zoomed hard copy or saving to disk. While in the zoom mode Graphx provides a marker to show where color boundaries begin and end. If you happen to be using the grey checker board pattern used for marking color boundaries in Graphx, it will still be present in your zoomed in copy.

Paint 'N Print has a high degree of magnification. Each pixel is shown as a square eight times its original size. Each square is outlined in a fine black line and each block of 64 are outlined in a bold black line. The bold line marks the color boundary for each character block.

When the zoom function is chosen Joy Print, TI Artist and Graphx all let the user choose which part of the screen will be zoomed by showing a box with which to enclose the desired area. In the Paint 'N Print environment, the place where the cursor is sitting when zoom is chosen becomes the central point of the screen, with the screen then acting as your "window" to a small but highly magnified part of the drawing. All of the drawing is accessible by scrolling it by this point, but the cursor never moves.

Most drawing packages have provisions for setting aside part of a picture and later adding it to another picture. This "scratchpad" memory can be handled in two ways: by saving a permanent version of the clipped picture onto a disk which you can reload as needed; or by storing the picture-part in intermediate memory where you can recall it when you need it, even though you have loaded new picture files in and out of the program since the picture part was saved.

When this type of picture-part is saved to disk it should not be confused with a regular picture file. When a picture file is loaded into your program, whatever you had on the screen before is erased and gone, and the new picture takes its place. Picture-parts however, are loaded IN ADDITION to whatever else is already there. These small pictures have become very popular with the drawing community so that they have their own special term - clipart.

Each program is unique in its handling of this additional storage. Joy Paint uses internal storage for a Cut and Paste method much like the paint programs that other popular computers use. All screens are saved in the same format. When something is wanted from another picture, save the current picture first, then load in the picture to be borrowed from. "Cut" out the piece you wish to use. Reload the original picture and "Paste" the new part anywhere on the drawing.

Bitmac uses the "Store" function for internal temporary storage. Current screen graphics can be overlaid with graphics stored on a disk, using what is called "Boolean Input". This allows special graphics effects which are unique to Bitmac.

TI Artist also has unique storage methods. Besides the normal full screen picture files, parts of pictures can be saved as "instances" or "slides". Slides are a collection of up to 24 miniture designs that can be independently designed, rotated and moved around on your drawing. Instances are images that can be added to your drawings or combined together in whatever manner you wish. They can become a permanent, editable part of your drawing. The nice thing about instances is that they are saved in a DISPLAY/VARIABLE 80 format which can easily be transported to Extended Basic programs or TI Writer files as well as being used for clipart.

Graphx has a very powerful "Clipboard" feature. With it you can create and store clipart permanently on a disk and it is also possible to copy a portion of one picture into another, much like Cut & Paste. A portion of a picture, or even serveral pictures, can be stored,then decided on later as to which ones to keep and which ones to erase.

In the high resolution mode each graphic position available to be used on our electrronic drawing board is called a pixel. You may remember being told that the screen is like a grid with 256 pixels accross and 192 pixel rows; and that each individual pixel on the screen can be turned off or on separately while you are drawing - all 49,152 of them! Right? Wrong if you are using color! Color resolution for the 99/4A is not the same as drawing resolution. We still have the same 192 rows of pixels, but instead of 256 pixels across, we have only 32 graphic positions across each row. Each row of pixels is grouped in eights, starting from the left of the screen, and each set must be the same two colors - a foreground and a background.

Now you can see why color resolution is 32 x 192. Any given group of eight horizontal pixels MUST be the same two colors. The groups on either side can carry entirely different colors, but each group is limited to two colors. Knowing this, and arranging your drawings according to the color boundarys is important when working with color.

Most programs make full use of the TI-994/A's 15 brilliant colors, allowing control over the foreground and background colors, and in many programs over the screen color as well. Sometimes the screen color is called the "backdrop".

All programs using color allow the swapping of one color in a drawing for any other. When the exchange takes place, every incidence of that color on the screen is swapped for the new one. Additionally, some programs like TI Artist and Graphx allow selective repainting of a chosen area.

Some of the programs provide special helps for working with color. TI Artist provides a function that lets a special color cursor move on color boundaries. Graphx does the same; also providing a "Grey and White Checkerboard" function which is handy for planning drawings which will use a lot of different colors. This makes it much easier to plan the various colors in your picture so that they don't bump into each other. When you no longer need the grid simply choose the "Remove Grey Boxes" option.

For special color effects, two programs that shine are Draw-A-Bit with its Redraw feature described earlier and Paint 'N Print which includes five extra rainbow colors in varying widths of horizontal and vertical stripes. Draw 'N Plot makes limited use of color. Only two are used at any one time - foreground and background. These colors can be easily switched so you can see how the various combinations of color look together.

Besides the Graphic Package, which doesn't use color either, Joy Paint is the only major paint program not using color. Here emphasis is on the manipulating of picture components, and color is used only as a background, with the pencil line alwys being your choice of either black or white. Painting refers to filling shapes with the many patterns available, or using the air brush to "spray paint" an area with a chosen pattern.

A Slide Show is a method of presenting pictures in a selected order. Bitmac is the only program with this feature built in; Draw A Bit and Draw 'N Plot have disk demos that you can adapt for your own pictures. TI Artist has an excellent companion disk called Display Master that gives you many options in designing your own slide display. Asgard Software put out a slide show program for Graphx files.

The Undo command lets you "take back" the last step of a drawing. If something was moved or erased that shouldn't have been, no harm done, just "undo" it. Joy Paint is the TI-994/A's only program with this feature but it is quite common in paint programs for other computers.

(Part VI will appear next month)

Relational Expressions

Jim Peterson

What are "Relational Expressions", you might say? The "blue book" that came with your computer says nothing about them, and most of the programming tutorial books on the subject are equally silent. If you waded through the computerese and mathematese text of the User's Reference Guide, you found them discussed on page II-14 under Relational Expressions and on page II-51 under IF-THEN-ELSE, but you probably didn't realize their potential. Then, you graduated to Extended Basic and found those easy-to-use, in-the-clear logical expressions AND, OR, NOT and XOR, and you looked no further.

So, what can a relational expression do? Nothing that can't be done without it. But it can often do the job so much more compactly, so much more efficiently, and therefore so much faster. So let's learn to use them. And let's learn in plain English, not computerese. The following may not be technically correct, but it's the way it all works out.

First, every expression has a true/false value, which is entirely different and separate from the value of the variables or numbers or strings it contains. On the TI-99/4A, a false statement has a value of 0, which is easy to remember - A FALSEHOOD IS WORTH NOTHING. Unfortunately, a true statement has a value of -1, which doesn't quite fit in too well. On some other computers you may have learned that a true expression has a value of +1, but on the TI it is -1. So, if in the expression ...F=7 :: IF F=8 THEN..., F=7 has a value of -1 because obviously F does equal 7, and F=8 has a value of 0 because it is not true.

Second, when an IF statement refers to a variable without an "=" sign, it means "<>0". For instance, IF X THEN 1000 means "if X is more or less than 0, if it is not 0, if it is anything other than 0, then go to 1000".

Third, the computer will try to use the expression mathematically before it tries to interpret its true/false value. Remember that everything within parentheses is worked first. For instance...X=1 :: Y=2 :: IF (X=1)+(Y=2) THEN 1000...Since both are true, this works out to IF(-1)+(-1)<>0 THEN 1000, and since -1 plus -1 is not 0, we go to 1000. On the other hand, X=1 :: Y=2 :: IF X=1+Y=2 THEN 1000 will first be calculated as X=1+Y, which comes out as X=3, and then as X=3=2, which has a true/false value of 0 (false) because x=3 has a true/false value of 0 (false), not 2!

Finally, always remember that a variable keeps its previous value until the calculation of an entire equation is completed. X=3 :: X=X+(X+3)*X-X/X^X+(X=0) is worked as X=3+(3+3)*3-3/3^3+(3=0).

Now that you have assimilated this vast knowledge, how can it be used? The most common way is in the expression IF (X=1)+(Y=2) THEN 200. In this case, if it is true that X=1 but Y does not equal 2, then-1+0 is<>0 so you go to 200. If X is not 1 but Y=2, then 0+-1 is still <>0, and if X=1 and Y=2 then -1 plus -1 is still <>0, so you still go to 200, but if X is not 1 and Y is not 2 then 0+0 is not <>0 so you do not. Of course, in Extended Basic, you could simply write IF X=1 OR Y=2 THEN 200.

If you want to go to 200 only if X=1 or if Y=2 but not if both are true, then you can write IF (X=1)+(Y=2)=-1 because either -1 plus 0 or 0 plus -1 will equal -1. In Extended Basic, this is the "exclusive OR", IF X=1 XOR Y=2. And if you want to go to 200 only if both are true, you can write IF (X=1)+(Y=2)=-2, or more commonly IF (X=1)*(Y=2) because if either or both are not true the multiplication by 0 will give 0. In Extended Basic, this is IF X=1 AND Y=2.

You can even write more complicated versions, carefully watching your parentheses, such as IF (X=1)+((Y=2)*(Z=3)) which translates to IF X=1 OR Y=2 AND Z=3. So, if you're programming in Extended Basic, why bother with all those parentheses? Why not just use OR and AND? In the above cases, that is true. But you have not yet begun to see the power of relational expressions!

Since the true/false value is a numeric value, it can be used in calculations, and it does not have to be used with an IF statement. For instance, this is a statement that I have used within a loop to alternate control of the two joysticks between two players ...X=X+1+(X=2)*2 :: CALL JOYSTICK(X,Y,Z). In this, the first time around, X has not been given a value, so the equation is read X=0+1+(0=2)*2 and since 0 does not equal 2, 0+1+(0*2)=1 and joystick #1 is activated. Next time around, X=1 and X=1+1+(1=2)*2 gives a value of 2, since 1=2 has a true/false value of 0. The third time around, X now has a value of 2 and X=2+1+(X=2)*2 which is worked as X=2+1+(-1)*2 and then X=2+1+(-2) which is X=2+1-2 and X=1 again!

If you think that's neat, look at this one from the Airport Area UG newsletter, credited to Robert Cooley, X=X=0 :: CALL JOYST(X+2,Y,Z). Here, the first time around, X does equal 0 so the statement X=0 has a true/false value of -1 so X=-1 and X+2 activates joystick #1. Then X=-1 so X has a true /false value of 0, so X=0 so X+Z activates Joystick #2..and so on! Of course, you could also write IF X=1 THEN X=2 ELSE X=1 if you prefer.

Another example: A=INT(10*RND):: B=INT (10*RND):: FOR J=A TO B. Now, if the random B happens to be smaller than the random A, the loop falls through with nothing happening. You could add a line IF A>B THEN T=1 ELSE T=-1 and FOR J=A TO B STEP T. But why not just FOR A TO B STEP (B<=A)+ABS(A<=B). If B<A then -1+ABS(0) gives a STEP -1 to count backwards, but if A<B then 0+ABS(-1) gives STEP 1 and if A+B then 0+ABS(0) equals STEP 0!

Another example: 100 INPUT "SCREEN COLOR ? ":S :: FOR SET=1 TO 14 :: X=SET+1-(SET)>=S) :: CALL COLOR(SET,X,X):: NEXT SET. That changes the character set to colors 2 to 16 in sequence, skipping over whatever color has been selected for the screen.

Strings can also be manipulated. 100 P$(1)="S" 110 INPUT "HOW MANY? ":N :: PRINT "THE PRICE IS "STR$(n)&" DOLLAR"&P$(ABS(N>1)) :: GOTO 110. Or more efficiently 100 INPUT "HOW MANY? ":N :: PRINT "THE PRICE IS & "STR$(N)SEG$("DOLLARS",1,7-(N>1)) :: GOTO 100

However, it is also possible to overdo it. The following routine will read key input to move the cursor around the screen in all 8 directions, stopping at the borders or

(continued on page 7)

Relational Expressions

(continued from page 6)

travelling along them if struck diagonally. However, it requires so many calculations for each key input that it is not the fastest method for accomplishing this.

```
100 CALL CLEAR :: R=1 :: C=3
110 CALL KEY(3,K,ST):: IF ST=0 THEN 110
120 C=C+((K=82)+(K=68)+(K=67))*(C<32)-
    ((K=87)+(K=83)+(K=90))*(C>2)
130 R=R+((K=90)+(K=88)+(K=67))*(R<24)-
    ((K=87)+(K=69)+(K=82))*(R>1)
140 CALL HCHAR(R,C,42):: GOTO 110
```

So - for compact, efficient programming, learn to use the relational expressions. But also learn when not to use them.

--Jim Peterson, TIGERCUB

Mystery Program

I cannot promise you that this program will show you how to balance your checkbook or aid you in keeping track of you mailing lists.

However try typing it in. It takes only a few minutes and you will see what clever people these 99/4A programmers are. As you can see by the listing, this one uses sprites and you will need XBasic to run it.

```
1 !SAVE DSK2.MYSTERY #2
100 !!!!!!!!!!!!!!!!!!!!!!!!
110 !  MYSTERY PROGRAM #2
120 !    by Chris Schram    !
130 ! Requires EX. BASIC
140 ! and Expansion Memory !
150 !!!!!!!!!!!!!!!!!!!!!!!!
160 CALL CLEAR :: CALL SCREE
N(1)
170 CALL INIT
180 FOR X=1 TO 28
190 RANDOMIZE
200 CALL PEEK(-31808,A,B)
210 CALL SPRITE(#X,46,16,A+1
,B+1,A-128,B-128):: CALL PEE
K(-31877,C):: IF C AND 32 TH
EN CALL SCREEN(10):: CALL SC
REEN(1)
220 CALL LOAD(-31744,A,"",-3
1744,B)
230 NEXT X
240 GOTO 180
```

2nd Annual P.U.N.N. PICNIC

Tuesday August 2, 1988-Milwaukie Elks Club

For Members and Their Families

$2.00/person

For Non-members, Guest & Door Sales

$4.00/person

Menu: Hamburgers, Hot Dogs, Potato Salad, Pork 'n Beans Chips, Watermelon, Pop, Punch, Coffee & all the fixings!

Swimming at the Elks Pool ($1.25).
See the July Wordplay or the
PUNN BBS for details

For tickets see Terry Priest
or Don Barker.

Personal Loader

This program can be made into your own personal loader for any disk that you have. In line 120 you can select the colors of the display to any that you like. As shown here it would be white characters on a black background but just put in anything that you prefer. In line 130 you can change the dimension to the number of files on your disk.

You will need to ENTER the name of your programs as DATA in line 200 (ENTER-EXIT PROGRAM) also.

Starting in line 280 list each of your programs on a seperate line as shown in the sample listing. Now save this program on your program disk as LOAD and you will be able to run any program that appears on that particular disk.

```
100 !SIMPLE LOADER & CALL KE
YPRESS DEMO - MODIFIED 3/88
110 !BY Paul E. Scheidemantl
e
120 CALL CLEAR :: CALL SCREE
N(5):: FOR A=0 TO 14 :: CALL
 COLOR(A,16,1):: NEXT A
130 DIM A$(22)
140 !MENU
150 RESTORE :: READ B :: IF
B>11 THEN C=1 ELSE IF B<7 TH
EN C=3 ELSE C=2
160 FOR A=1 TO B :: READ A$(
A):: DISPLAY AT(A*C,4):CHR$(
A+64)&" - "&A$(A):: B$=B$&CH
R$(A+64):: NEXT A
170 !SAVE # PROGRAMS (+1 for
 EXIT) IN THIS DATA STATEMEN
T!
180 DATA 4
190 !PLACE NAMES OF PROGRAMS
 HERE!
200 DATA EXIT PROGRAM,COLOR,
CONVERSION,HEX
210 DISPLAY AT(24,1):" YOUR
CHOICE:"
220 CALL KEYPRESS(24,23,B$,6
6,RK)
230 !ADD LINE # OF NEXT PROG
RAM TO BE RUN IN LINE 220 SU
CH AS 281,282,283, ETC!
240 IF RK>65 THEN DISPLAY AT
(12,1)ERASE ALL:"Loading....
":"":A$(RK-64)
250 ON RK-64 GOTO 260,280,29
0,300,280
260 DISPLAY AT(24,1):" QUIT
ARE YOU SURE? : (Y/N)" :: CA
LL KEYPRESS(24,29,"YNyn",89,
RK):: IF RK=89 OR RK=121 THE
N CALL CLEAR :: END ELSE 210
270 !PLACE RUN STATEMENTS HE
RE CONSECUTIVELY!
280 RUN "DSK1.COLOR"
290 RUN "DSK1.CONVERSION"
300 RUN "DSK1.HEX"
310 END
320 SUB KEYPRESS(ROW,COL,CHK
$,DC,RK)
330 CALL KEY(3,RK,S):: CALL
HCHAR(ROW,COL,32):: CALL DEL
AY(9):: IF S=0 THEN CALL HCH
AR(ROW,COL,DC):: CALL DELAY(
9):: GOTO 330
340 X=POS(CHK$&CHR$(13),CHR$
(RK),1):: IF X=0 THEN 330 EL
SE IF RK=13 THEN RK=DC
350 IF RK>31 AND RK<128 THEN
 CALL HCHAR(ROW,COL,RK)
360 SUBEND
370 SUB DELAY(D):: FOR A=1 T
O D :: NEXT A :: SUBEND
```

Talking Typewriter

Talking Typewriter is a little program that can help young people become acquainted with the keyboard of a computer or a typewriter.

Type in the program and then run it. Whenever you press a key the speech synthysizer will speak out the character. As programmed it will recognize letters but you could change line 140 so that it would speak out numbers as well.

Have fun with this little program.

```
100 REM TALKING TYPEWRITER
110 CALL CLEAR
120 CALL KEY(0,K,S)
130 IF S=0 THEN 120
140 IF K>90 THEN 120
150 IF K<65 THEN 120
160 CALL CLEAR
170 CALL SPRITE(#1,K,2,85,12
0)
180 CALL MAGNIFY(2)
190 CALL SAY(CHR$(K))
200 GOTO 120
```

PUNN
P.O. Box 15037
Portland, OR 97215

DISCLAIMERS: The PUNN User's Group is not affiliated with or sponsored by TI and has no relationship with them, implied or otherwise.
Mention of a company or product is not an endorsement of that company or product.
We are not a subsidiary or branch of any other User's Group and any relationship we may have with other groups is on the basis of equals.

ALL GENER MEETINGS ARE HELD
ON THE FI AY OF EACH
MONTH ILDING
37
PO
!! NEXT
AUGUST 2N

2nd. Annual
P.U.N.N. PICNIC
Tuesday August 2, 1988-Milwaukie Elks Club

The PUNN Newsletter

WORDPLAY

P.O. Box 15037 Portland, OR 97215

Pointing the Way For
Users of TI's 994/A Computer

August 1988-Volumn- 7-No. 8