THE GUILFORD 99'ER NEWSLETTER

VOL.4 NO.8 AUGUST 1987

Mack Jones, President
Robert Carmany, Vice President
Mike Garrett, Secretary/Treasurer
Robert Dobo, Program Library
BBS: (919)274-5760(OPUS)

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (Herman Geschwind, Editor)

OUR NEXT MEETING

DATE: August 4, 1987. TIME: 7:30 PM PLACE: Glenwood Recreation Center 2010 S. Chapman Street.

We have planned something different for the August meeting. Instead of software demos and presentation we will present a hardware project this time. George Mcormick will show us how to build a Supercart. Just in case you are wondering, a Supercart is a home-built module with battery-backed GROM chips. The Supercart can be used like any other TI Solid State module, except that it can be loaded with the software of YOUR choice. Supercarts have taken the TI community by storm due to their low cost and the added convenience and versatility that they present.

PRES PEEKS

The July meeting showed that a few members are willing to brave the heat and the storm to attend something they enjoy! With foreboding storm clouds hovering overhead, we still had a very good turn-out.

I would like to thank Bob for a very informative lecture on some of the speech capabilities of Forth. I would also like to thank Herman for his talk and demo on the Personal Accounting software program. Also his remarks and demo on Roulette.

Ever see a wooden disk drive case? I never had either until meeting night when Bonnie and Ben Jones pulled in beside me and raised the trunk lid. I thought I was looking at a wooden P/Box!! Really, it is a beautiful job that was made for them by a gentleman in Thomasville for their drive. Ben explained that it has no cooling fan or cards, just a disk drive. I had brought my PB so we didn't use it but I wish I had asked Ben to bring it in to show the members. Perhaps he will next month.

I had been reading articles from several of the news letters on changing the VDP load resistor to remove the shading you get around the letters on your monitor or TV set. I decided to try it before I would recommend it for the members, so after hunting through my scrap box of about 2,000 resistors of every value, I finally found a 330 ohm resistor. I didn't want to try it on my good console, so I used old faithful out in the shop that I use for my experiments , and removed the 560 ohm resistor that TI for some reason has placed instead of the intended 330 ohm. I inserted the 330 ohm and soldered it in. I wanted to make sure it did the job before I went to all the trouble of putting the case back together, so I hooked it up naked. To my surprise, all the white shading (I have a black and white TV I use in the shop) was gone from the letters. I wanted to check it with color, so I brought it into the house and hooked it to my color monitor. The dark shading that I usually have beside white letters were gone also. I took the mama board back out and put it all together, turned it on and...NOTHING!! I knew that I had hooked the keyboard up, I just knew it! I opened it back up and lo and behold, about 10 of the keyboard socket wires had broken off! I guess that I had opened and closed the case so many times that they had just been flexed too much. Talk about a time consuming job, just try resoldering all those little wires back on the keyboard! But it works fine now.

If you intend to replace the resistor, you will find it in the upper left hand corner of your mother-board just to the left of the TMS chip. It will be the bottom of two resistors that are located right under two transistors. The 560 ohm resistor will be color coded Green,Blue,Brown. The 330 ohm will be coded Orange,Orange,Brown. If you would like to have the color/value of resistors , here they are:

BLACK - 0	GREEN - 5
BROWN - 1	BLUE - 6
RED - 2	VIOLET- 7
ORANGE- 3	GRAY - 8
YELLOW- 4	WHITE - 9

The fourth band of color will either be gold or silver. This is the tolerance of the resistor...Gold being 5 percent, and Silver being 10 percent. To read the value of a resistor, read the bands from left to right with the tolerance band on the right. Thus, a resistor that reads "red,green,black" would have a value of 25 ohms, since the black band indicates no zero. The third band is the multiplier, and indicates how many zero's to add to the first two bands. Another example would be "brown,black,green" which would be 1 Meg-ohm. This may be of help to you someday. Also, resistors are supplied in 1/4, 1/2, and 1 Watt sizes, as well as wire-wound types of much larger wattage. The ones most used in computers is 1/4 which is about the diameter of a pencil lead, to the 1 watt which is about the size of a cigarette filter. The one needed for your VDP project will be a 1/4 Watt size.

If there is any doubt in your mind as to whether you should try this modification, don't do it! It takes a very small wattage soldering iron and not a little know-how on de-soldering, and re-soldering to accomplish the job and I would hate for you to mess up a good console. Before you tackle it, let's talk about it.

See you all at the August meeting and hopefully George McCormick will show us how to build a Super Cart. (Submitted by "Mac" Jones)

TI SHOPPER

I'm going to start this month's column off with a status report on FUNNELWEB. The McGoverns have once again been busy and are in the midst of adding a further enhancement to the system. The word from Tony McG. is that option #6 on the loader screen is being modified to allow the loading of A/L object code into low memory. He stated that there was some "tidying up" to do before it is released but it should be finished soon.

Marty Kroll's DISK CATALOGUE program (Herman's brief demo last month) brings some welcome improvements to this fine program. First of all, the sort routines have been speeded up and the ability to merge several data files has been added. There is also mention of a COMPANION disk to be released soon that will allow the user to add program information to the files, etc. The version number is now 1.5.

Of course, the "new" Myarc Geneve is currently being shipped. It is supposed to be 95% compatible with current TI programs. It seems, however, that the "free" software for Geneve has failed to materialize. The eight pieces of software are NOT shipped with the machine as advertised. Instead, purchasers are given a coupon to be mailed in and the software will be shipped "as soon as available". That appears to be typical of Lou Phillips and Myarc of late.

Ryte Data has come out with 2 EPROMs for your TI Disk Controller that will allow you to use a quad-density drive to access 1440 sectors per disk. It will format either 40 or 80 tracks but you have to use the TI DiskManager II to initialize the disks. Half height drives listed as 96 tpi or quad density can be used to take advantage of the increase in storage capacity. It should be noted that this modification is NOT compatible with either the Myarc or CorComp double density format. The price for the EPROMs is $45.00.

Oh yes, there is also a new disk of programs in our library compliments of the Hunter Valley UG in Australia. The programs are all original, written by the UG members, and very interesting. They range in scope from "kiddie games" to some that are a bit more advanced. Best of all, just check the disk out at the next meeting.

Well, this is a little short this month but I'll try to make up for it next month. 'Til next time . . . (Contributed by Bob Carmany).

LOGO TIMES

Having spent the better part of the year trying to convince you that LOGO isn't just a cute kids language, it's about time for some regression. That is some plain, simple, cute fun.

The following is a program which I wrote in about an hour, using tips and portions of procedures provided by Mrs. Elaine Bologna, a computer teacher at Summitt School in Winston-Salem. She, you may recall, graciously provided us with a TI-LOGO lecture and demonstration about a year ago in Kernersville.

When you type this program in, take a few moments after each line or procedure to think about what it might or should do. If you've done little or no LOGO programming, but do have the LOGO package, just follow the instructions in the front of the manual to load LOGO. Type the program exactly as you see it here. You can consult the index of commands at the rear of the manual for their definitions. It doesn't matter what order you type the procedures. Remember, each procedure begins with "TO" and ends with "END." Each stands alone. Thus, in the following program, it doesn't matter where the procedure THUMPER is. As the main procedure of the program, it "calls" the others regardless of what order they were written or entered.

Also, contemplate a while about what you, or a youngster developing this program, might learn about space and geometry. LOGO never ceases to amaze and getting it to do what you want is the best part. In fact, you might try altering a procedure, say one like ARC which is called several times, to see what the effect is.

Once you know how to get into and out of the editor (covered in section 1.4 of the manual), interaction is simple and the results easily and quickly viewed. Experimentation is a LOGO hallmark, and its one of LOGO's finest features.

Note also that procedure "TAIL" is called to make something other than a tail at times. Nomenclature is entirely in the hands of the programmer in LOGO.

Here you are, so have at it. And have fun.

```
TO THUMPER
TELL TURTLE
CS
HT
BODY
TAIL
MOVE
BODY
MOVE2
EAR
MOVE3
EAR2
MOVE4
TAIL
MOVE5
TAIL
MOVE6
ARC
MOVE7
MESSAGE
HT.END


TO MESSAGE
FD 40 BK 20 RT 90
FD 10 LT 90 FD 20
```

```
BK 40 RT 90 PU
FD 10 LT 90 PD
FD 30 PU
FD 8 FD FD 2
END


TO MOVE7
PU SH 270
FD 45 PD
SH 0
END


TO MOVE6
PU LT 150
FD 29 PD
SH 240
END


TO MOVE5
PU RT 80
FD 15 PD
END


TO MOVE4
PU LT 90
FD 10 PD
END


TO EAR2
EAR
END


TO MOVE3
PU SH 0 RT 90
FD 15 LT 60
PD
END


TO EAR
ARC RT 75 ARC
END


TO ARC
REPEAT 12 ▀FD 3 RT 8▬
END


TO MOVE2
PU FD 25 RT 90
FD 10 LT 90 PD
```

```
END


TO MOVE
PU FD 45 PD
END


TO TAIL
REPEAT 36 ▀FD 1 LT 10▄
END


TO BODY
REPEAT 36 ▀FD 4 RT 10▄
END
```

(Contributed by Larry Spohn)

FORTH FORUM

This month, we are going to discuss portability ----what it is and what not to do when you write Forth applications. You will find that it can make a big difference in how you structure your Forth coding as to whether you will be able to use the same definition in more than one Forth environment.

Portability is the ability of a Forth application to be used in more than one Forth environment without modification. That is, being able to use a particular Forth definition in both TI-Forth or Wycove Forth without modifying it. Portability can even refer to being able to use a definition in more than one version of the same Forth language (ie Version 2.1 and 3.0 of Wycove Forth).

Let's take a look at how it all works. For example, let's create a definition for GOTOXY in Wycove Forth. There is no equivalent for it in Wycove as the system is delivered. GOTOXY is the same as the XB DISPLAY AT. There are several ways that it can be done:

```
: GOTOXY SCREEN-WIDTH @ * + CURSOR-POS ! ;

: GOTOXY SCREEN-WIDTH @ * + >C9C8 ! ;

: GOTOXY SCREEN-WIDTH @ * + ▀CURSOR-POS▄ LITERAL ! ;
```

All three of these definitions will work in Wycove Forth Versions 2.0 or 2.1. The problem arises when you want to use the definition in Wycove Forth Version 3.0, however. The second of the definitions uses an absolute Hexadecimal memory location. When the Forth version changed, it made the code invalid for version 3.0. Either the first or the third version will work in Wycove Forth Version 3.0 but the third is the more efficient of the two choices remaining. The lesson to be learned here is not to use absolute memory locations. If the version changes, it is likely that the code will no longer be valid for the newer version.

The problem is even more acute when you try to transfer a definition from one Forth language to another. For example, we can write the code for PICK in a couple of ways in TI-Forth:

```
CREATE PICK (item# -- item) HEX C019 , 0A10 , A009 , C650 , 045F , DECIMAL

: PICK 2 * SP@ + @ ;
```

Again, the first definition uses absolute Hexadecimal memory locations (which is basically a bad idea). Since there are several variations of TI-Forth "floating around", the code may not be valid for all of the versions in existence. The second definition is much more suitable because it doesn't use any unique memory locations.

Even better, the second definition of PICK uses words that are common to the vocabularies of both TI-Forth and Wycove Forth. The advantage of a definition that is constructed of common vocabulary words is that it is portable between Forth versions and the two Forth languages available for the TI.

The obvious lesson to be learned here is to try to use those words that are common to both languages when you are considering a truly portable application of Forth. There are significant differences between TI-Forth and Wycove Forth but they are certainly not insurmountable and effort, definitions can be constructed that can be used in both Forth

MODEM TALK

First, the sad news. Dan Post has given up the ROS bulletin board. As of late he had been plagued with telephone line problems and mysterious bugs in the version of the BBS software that he was using.

Dan had tried to get a better software version for months but no success. Well, to make a long story short, since the board was just not working as well as it should, Dan decided to give up.

In a way it really should come as no surprise. To run a bulletin board for the world at large is really the utmost in altruism. It takes an investment of well over $2500 to set up a system complete with a large capacity fixed disk, a sophisticated modem, etc. which must be fully dedicated to just this one use. Then there is the recurring expense for electrical power, phone bills for a separate line and so on and so forth. The reward? Some boards solicit contributions or membership fees to defray some of the expense but to my knowledge, nobody has gotten rich running a BBS system yet. Unfortunately there is very little appreciation from the users for this wonderful service. On the contrary, Sysops have to take a lot of abuse if things just don't happen to work right and then there are the malicious nitwits who try the crash the system, making life even more difficult for the Sysop. Running a BBS can be a very shortlived thrill and the novelty wears off very quickly with the daily grind of maintaining the system and keeping things backed up.

All we can do now is thank Dan for his efforts and the help he has given us in the past. All the TI files have been given to Benn Mann of the Opus board and Benn and I will have to tackle the task what to move over to OPUS.

New uploads are as follows:

CATARC.TIP (88K) See Bob Carmany's comments elsewhere in this issue about Marty Kroll's latest version of Disk Cataloguer.

MONARC.TI (86K) Yet another version of the Monopoly game. This one has extensive assembler support.

OMARC.TI (30K) A new twist to a communications program. The specialty of this one is that RLE pictures can be viewed on-line. If you need to look at aviation weather forecasts or need pictures of the FBI's ten-most-wanted, then Omega is just great.

TOOLARC.TIP (71K) Tools for the XB programmer. We covered this one at our last meeting.

MENU6 (10K) Support software (Menus) for the HRD. Thanks George M. for the upload. (Contributed by H. Geschwind)

BASIC CORNER

Our program this month is a very delightful keyboard drill program. There has been a lot of talk about the lack of computer literacy but in my book what is an even more fundamental problem is the lack of familiarity with the QWERTY keyboard. I have seen some very smart people with college degrees to prove become very frustrated with a computer simple because they just did not find the proper key in time.

Learning to use a keyboard is a fundamental skill just like swimming or steering a car which is best acquired at an early age. Our program, which runs equally well in Console as well as Extended Basic takes a non-threatening, game like approach to reinforce basic keyboarding. Enjoy.

```
100 REM THIS IS A TYPING IMP
ROVEMENT PROGRAM
110 DEF RD(X)=INT(X*100+.5)/
100
120 DIM LTRS$(41),MISSES(41)
,ESCAPES(41),TRIES(41)
130 RANDOMIZE
140 CALL CLEAR
150 GOSUB 1000
160 ASUB=INT(RND*42)
170 ALPH=ASC(LTRS$(ASUB))
180 TRIES(ASUB)=TRIES(ASUB)+
1
190 DIR=INT(RND*4)+1
200 COUNT=COUNT+1
210 IF COUNT<>NEXTEST THEN 2
30
220 GOSUB 5000
230 CALL HCHAR(CROW,CCOL,ALP
H)
240 TDLY=0
250 CALL SOUND(-150,1760,0)
260 ON DIR GOSUB 2000,2500,3
000,3500
270 IF STRIKE=15 THEN 290
280 IF (COUNT<MAXT)*(ESCAPEE
S<MAXESC)THEN 160
290 CALL CLEAR
300 IF ESCAPEES<MAXESC THEN
350
310 PRINT "You have allowed
";ESCAPEES;" DANGEROUS Alpha
 Bettis"
320 PRINT " to escape.": :"T
his is a TERRIBLE record":"
for a person"
330 PRINT " in your position
!!!"
340 PRINT : :"Pack up your c
omputer!!!":"  YOU'RE FIRED!
!!"
350 PRINT : :"Your average s
core is ";RD(TIMER/COUNT)
360 GOSUB 7000
370 PRINT : :"Try again?(Y/N
)->";
380 CALL KEY(0,ALPH,ST)
390 IF ST=0 THEN 380
400 PRINT CHR$(ALPH)
410 IF CHR$(ALPH)="Y" THEN 1
50
420 IF CHR$(ALPH)<>"N" THEN
370
428 CALL CLEAR
```

```
430 STOP
1000 TIMER=0
1010 COUNT=0
1020 ESCAPEES=0
1030 LSTIMER=0
1040 SC=5
1050 NEXTEST=20
1060 LROW=2
1070 HROW=22
1080 LCOL=4
1090 HCOL=28
1100 FOR I=0 TO 41
1110 TRIES(I)=0
1120 MISSES(I)=0
1130 ESCAPES(I)=0
1140 NEXT I
1150 IF LTRS$(0)="A" THEN 16
10
1160 NEXTLEV(1)=10
1170 NEXTLEV(2)=12
1180 NEXTLEV(3)=14
1190 NEXTLEV(4)=16
1200 NEXTLEV(5)=18
1210 NEXTLEV(6)=20
1220 CCOL=16
1230 CROW=12
1240 CALL CHAR(128,"017F4549
516141FF")
1250 CALL CHAR(129,"81C3A599
99A5C381")
1260 CALL CHAR(130,"80FEC2A2
928A86FF")
1270 CALL CHAR(131,"FF422418
182442FF")
1280 CALL CHAR(132,"FF615149
45437F01")
1290 CALL CHAR(133,"FFB68A92
A2C2FEB0")
1300 CALL CHAR(136,"01020408
13244893")
1310 CALL CHAR(137,"00804030
10CC2211")
1320 CALL SCREEN(SC)
1330 FOR I=1 TO 12
1340 CALL COLOR(I,2,3)
1350 NEXT I
1360 CALL COLOR(13,12,2)
1370 CALL COLOR(14,15,7)
1380 FOR I=0 TO 41
1390 READ LTRS$(I)
1400 NEXT I
```

```
1402 PRINT : : : : : : :TAB(
1);"Welcome to Extraterresti
al": :TAB(8);"Typing Program
": : : : : : : : : : : : :
1403 PRINT : :"Do you want"
1404 INPUT "instructions(Y/N
)->":Y$
1406 IF (SEG$(Y$,1,1)="N")+(
SEG$(Y$,1,1)="n")THEN 1610
1408 IF (SEG$(Y$,1,1)<>"Y")*
(SEG$(Y$,1,1)<>"y")THEN 1403
1410 CALL CLEAR
1418 PRINT : : : :
1420 PRINT "You are Head Kee
per of the":"T. Y. P. Qwerty
 Extraterres-"
1430 PRINT "trial Exotic Bio
logy Center."
1440 PRINT "Your job is to c
are for the":"non-native ani
mals that are"
1450 PRINT "brought to Earth
 by the far":"ranging StarSh
ips."
1460 PRINT :"The ExoBiology
exploration":"and research s
hip Zoologi-"
1470 PRINT "cal Gardens has
just return-":"ed from the n
ewly discovered"
1480 PRINT "planet Alpha Bet
ti with a":"cargo of unusual
, and unpre-"
1490 PRINT "dictable, animal
s native to":"that planet.":
 : : :
1500 GOSUB 8000
1505 CALL CLEAR
1510 PRINT : : :"You must ke
ep the animals":"within the
confines of their"
1520 PRINT "fenced area.  If
 any escape":"the area, you
must activate"
1530 PRINT "the teleportatio
n switch on":"your control p
anel that cor-"
1540 PRINT "responds to the
animal that":"has escaped. T
his immediate-"
1550 PRINT "ly brings that a
nimal back":"to the fenced a
rea."
```

```
1560 PRINT :"BUT, teleportat
ion equip-":"ment has a limi
ted range."
1570 PRINT "You MUST hit the
 proper"
1580 PRINT "switch BEFORE th
e animal":"reaches the mount
ains."
1590 PRINT "If you let too m
any animals":"escape, YOU'RE
 FIRED!!!": : : : :
1600 GOSUB 8000
1610 CALL CLEAR
1620 CALL HCHAR(23,1,129,32)
1630 PRINT :"Make sure the A
LPHA LOCK key":"is DOWN!!!"
1640 PRINT :" "
1650 CALL HCHAR(23,3,129,28)
1660 PRINT : :
1670 PRINT " 1 Super Expert"
:" 2 Touch Typist":" 3 Two H
anded"
1680 PRINT " 4 Two Fingered"
:" 5 One Fingered":" 6 Worki
ng Blindfolded"
1690 PRINT : : : :"Select yo
ur level -> ";
1700 CALL KEY(0,LEV,ST)
1710 IF ST=0 THEN 1700
1720 IF (LEV<49)+(LEV>54)THE
N 1660
1730 PRINT CHR$(LEV)
1740 LEV=LEV-48
1750 DELAY=LEV
1755 CALL CLEAR
1760 PRINT : : : : :" 1 Shor
t Game":" 2 Medium Game":" 3
 Long Game"
1770 PRINT : : : :"Select ga
me length -> ";
1780 CALL KEY(0,MAXT,ST)
1790 IF ST=0 THEN 1780
1800 IF (MAXT<49)+(MAXT>51)T
HEN 1760
1810 PRINT CHR$(MAXT)
1820 MAXT=MAXT-48
1830 MAXT=MAXT*75
1840 MAXESC=INT(LEV*MAXT/10)
1850 FOR I=1 TO 24
1860 CALL HCHAR(I,1,0,32)
1870 NEXT I
1880 CALL HCHAR(11,15,128)
1890 CALL HCHAR(11,16,129)
1900 CALL HCHAR(11,17,130)
```

```
1910 CALL HCHAR(12,15,131)
1920 CALL HCHAR(12,17,131)
1930 CALL HCHAR(13,15,132)
1940 CALL HCHAR(13,16,129)
1950 CALL HCHAR(13,17,133)
1960 CALL HCHAR(12,16,32)
1970 GOSUB 6000
1980 RETURN
2000 CALL HCHAR(CROW,CCOL,32
)
2010 GOTIT=0
2020 ROW=10
2030 CALL HCHAR(10,16,ALPH)
2040 GOSUB 4000
2050 IF GOTIT THEN 2180
2060 FOR ROW=9 TO LROW STEP
-1
2070 CALL HCHAR(ROW+1,16,0)
2080 CALL HCHAR(ROW,16,ALPH)
2090 GOSUB 4000
2100 IF GOTIT THEN 2180
2110 NEXT ROW
2120 ESCAPEES=ESCAPEES+1
2130 ESCAPES(ASUB)=ESCAPES(A
SUB)+1
2140 TIMER=TIMER+50
2150 CALL HCHAR(LROW,16,0)
2160 CALL SOUND(300,131,2,26
2,4,-2,3)
2170 RETURN
2180 CALL HCHAR(ROW,16,0)
2190 CALL SOUND(120,440,5,10
47,0,1319,3,-1,7)
2200 RETURN
2500 CALL HCHAR(CROW,CCOL,32
)
2510 GOTIT=0
2520 ROW=14
2530 CALL HCHAR(14,16,ALPH)
2540 GOSUB 4000
2550 IF GOTIT THEN 2680
2560 FOR ROW=15 TO HROW
2570 CALL HCHAR(ROW-1,16,0)
2580 CALL HCHAR(ROW,16,ALPH)
2590 GOSUB 4000
2600 IF GOTIT THEN 2680
2610 NEXT ROW
2620 ESCAPEES=ESCAPEES+1
2630 ESCAPES(ASUB)=ESCAPES(A
SUB)+1
2640 TIMER=TIMER+50
2650 CALL HCHAR(HROW,16,0)
2660 CALL SOUND(300,131,2,26
2,4,-2,3)
```

```
2670 RETURN
2680 CALL HCHAR(ROW,16,0)
2690 CALL SOUND(120,440,5,10
47,0,1319,3,-1,7)
2700 RETURN
3000 CALL HCHAR(CROW,CCOL,32
)
3010 GOTIT=0
3020 COL=14
3030 CALL HCHAR(12,14,ALPH)
3040 GOSUB 4000
3050 IF GOTIT THEN 3180
3060 FOR COL=13 TO LCOL STEP
 -1
3070 CALL HCHAR(12,COL+1,0)
3080 CALL HCHAR(12,COL,ALPH)
3090 GOSUB 4000
3100 IF GOTIT THEN 3180
3110 NEXT COL
3120 ESCAPEES=ESCAPEES+1
3130 ESCAPES(ASUB)=ESCAPES(A
SUB)+1
3140 TIMER=TIMER+50
3150 CALL HCHAR(12,LCOL,0)
3160 CALL SOUND(300,131,2,26
2,4,-2,3)
3170 RETURN
3180 CALL HCHAR(12,COL,0)
3190 CALL SOUND(120,440,5,10
47,0,1319,3,-1,7)
3200 RETURN
3500 CALL HCHAR(CROW,CCOL,32
)
3510 GOTIT=0
3520 COL=18
3530 CALL HCHAR(12,18,ALPH)
3540 GOSUB 4000
3550 IF GOTIT THEN 3680
3560 FOR COL=19 TO HCOL
3570 CALL HCHAR(12,COL-1,0)
3580 CALL HCHAR(12,COL,ALPH)
3590 GOSUB 4000
3600 IF GOTIT THEN 3680
3610 NEXT COL
3620 ESCAPEES=ESCAPEES+1
3630 ESCAPES(ASUB)=ESCAPES(A
SUB)+1
3640 TIMER=TIMER+50
3650 CALL HCHAR(12,HCOL,0)
3660 CALL SOUND(300,131,2,26
2,4,-2,3)
3670 RETURN
3680 CALL HCHAR(12,COL,0)
```

```
3690 CALL SOUND(120,440,5,10
47,0,1319,3,-1,7)
3700 RETURN
4000 CALL KEY(0,STRIKE,ST)
4010 IF ST<>0 THEN 4040
4020 TDLY=TDLY+1
4030 IF TDLY=DELAY THEN 4150
 ELSE 4000
4040 IF STRIKE=15 THEN 4130
4050 IF STRIKE=ALPH THEN 411
0
4060 CALL SOUND(-150,-3,0)
4070 MISSES(ASUB)=MISSES(ASU
B)+1
4080 TIMER=TIMER+5
4090 TDLY=TDLY+1
4100 GOTO 4030
4110 TIMER=TIMER+TDLY
4120 TDLY=0
4130 GOTIT=1
4140 RETURN
4150 TIMER=TIMER+TDLY
4160 TDLY=0
4170 RETURN
5000 PFACT=(TIMER-LSTIMER)/2
0
5010 RANDOMIZE
5020 LSTIMER=TIMER
5030 IF PFACT<NEXTLEV(LEV)TH
EN 5170
5040 IF PFACT<NEXTLEV(LEV)*1
.3 THEN 5260
5050 IF HROW=22 THEN 5150
5060 HROW=HROW+1
5070 LROW=LROW-1
5080 HCOL=HCOL+1
5090 LCOL=LCOL-1
5100 CALL HCHAR(LROW,LCOL,0,
HCOL-LCOL)
5110 CALL HCHAR(HROW,LCOL,0,
HCOL-LCOL)
5120 CALL VCHAR(LROW,LCOL,0,
HROW-LROW)
5130 CALL VCHAR(LROW,HCOL,0,
HROW-LROW+1)
5140 GOTO 5260
5150 DELAY=DELAY+1
5160 GOTO 5260
5170 IF HROW=15 THEN 5240
5180 HROW=HROW-1
5190 LROW=LROW+1
5200 HCOL=HCOL-1
5210 LCOL=LCOL+1
5220 GOSUB 6000
```

```
5230 GOTO 5260
5240 IF DELAY=1 THEN 5260
5250 DELAY=DELAY-1
5260 NEXTEST=COUNT+20
5270 RETURN
6000 FOR I=1 TO LCOL-1 STEP
2
6010 CALL VCHAR(1,I,136,24)
6020 NEXT I
6030 FOR I=2 TO LCOL-1 STEP
2
6040 CALL VCHAR(1,I,137,24)
6050 NEXT I
6060 MCHAR=137
6070 IF INT(LCOL/2)*2=LCOL T
HEN 6090
6080 MCHAR=136
6090 FOR I=LCOL TO HCOL
6100 CALL VCHAR(1,I,MCHAR,LR
OW-1)
6110 CALL VCHAR(HROW+1,I,MCH
AR,24-HROW)
6120 MCHAR=273-MCHAR
6130 NEXT I
6140 MCHAR=136
6150 IF INT((HCOL+1)/2)*2=HC
OL THEN 6170
6160 MCHAR=137
6170 FOR I=HCOL+1 TO 32
6180 CALL VCHAR(1,I,MCHAR,24
)
6190 MCHAR=273-MCHAR
6200 NEXT I
6210 RETURN
7000 PRINT : :"Do you want t
o see":" your detailed score
? (Y/N)->";
7010 CALL KEY(0,ALPH,ST)
7020 IF ST=0 THEN 7010
7030 PRINT CHR$(ALPH)
7040 IF CHR$(ALPH)="Y" THEN
7070
7050 IF CHR$(ALPH)<>"N" THEN
 7000
7060 RETURN
7070 GOSUB 7180
7080 FOR I=0 TO 18
7090 PRINT TAB(3);LTRS$(I);T
AB(7);TRIES(I);TAB(14);MISSE
S(I);TAB(22);ESCAPES(I)
7100 NEXT I
7110 GOSUB 8000
7120 GOSUB 7180
7130 FOR I=19 TO 37
```

```
7140 PRINT TAB(3);LTRS$(I);T
AB(7);TRIES(I);TAB(14);MISSE
S(I);TAB(22);ESCAPES(I)
7150 NEXT I
7160 GOSUB 8000
7162 GOSUB 7180
7164 FOR I=38 TO 41
7166 PRINT TAB(3);LTRS$(I);T
AB(7);TRIES(I);TAB(14);MISSE
S(I);TAB(22);ESCAPES(I)
7167 NEXT I
7168 PRINT : : : : :
7169 GOSUB 8000
7170 RETURN
7180 CALL CLEAR
7185 PRINT :TAB(12);"PRESSED
":TAB(6);"ESCAPE";TAB(13);"W
RONG";TAB(20);"REACHED"
7190 PRINT " KEY";TAB(6);"TR
IES";TAB(14);"KEY";TAB(21);"
MTNS"
7200 RETURN
8000 PRINT "PRESS ANY KEY TO
 CONTINUE."
8010 CALL KEY(0,ALPH,ST)
8020 IF ST=0 THEN 8010
8030 RETURN
10000 DATA A,B,C,D,E,F,G,H,I
10020 DATA J,K,L,M,N,O,P,Q,R
10040 DATA S,T,U,V,W,X,Y,Z
10060 DATA ",",.,;,0,1,2,3,4
10080 DATA 5,6,7,8,9,/,"""",
=
```

FOR SALE

*Peripheral Expansion System with Disk Drive, Disk Controller card, disk manager module, memory expansion.........$380.00

*Smith Corona - TPI Daisy Wheel printer with RS 232 card and printer cable...$220.00

*Disk Controller Card with disk manager module..$45.00

*Many TI programs (i.e. TI Writer, TI Logo, Microsoft Multiplan, Extended Basic and many others).

Call George Jordan (evenings) 299-4999

Navarone Widget cartridge extender........$10.00
(will have at next meeting - H. Geschwind).