YESTERDAY'S NEWS

Volume 5 Number 12 Established 2016 December 2020

INSIDE INFORMATION

30 Years Ago...

Historical Information taken from Bill Gaskills TIMELINE

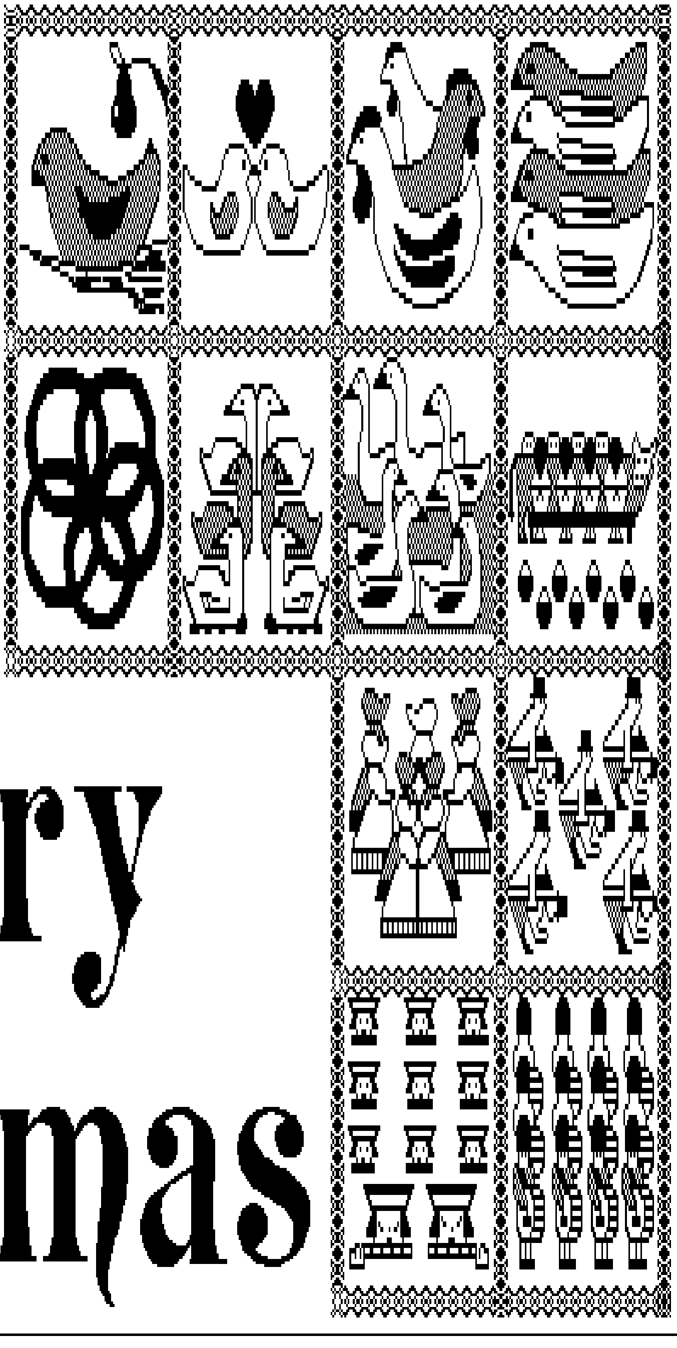
December 1990:

Myarc founder Lou Phillips is said to be severely curtailing his TI-99/4A support activities due to a new job and family demands.

John Birdwell, author of Disk Utilities and contributing programmer to Myarc's DM5 disk manager, and project programmer for Myarc's Tape Streamer software for HFDC backups, is reported to be terminally ill with cancer of the liver. He dies on December 27th at age 41.

Noted TI-99 author and writer Cheryl Regena Whitelaw celebrates her 10th anniversary as a TI-99 owner and user.

MICROpendium reports that Asgard's new YAPP (Yet Another Paint Program) drawing program authored by Alexander Hulpke does not work properly with the Myarc Mouse, but does work correctly with the Asgard mouse. YN



Merry Christmas

TIPS FROM THE TIGERCUB NUMBER 19

By Jim Peterson

Danny Michael has improved his graphics screen dump to include rotate and double size! It is in assembly, very fast, and runs out of XBasic, E/A module or Mini Memory. He has also written an assembly Neatlist program which lists an XBasic program to a printer in single line statements, indented, expanded, etc., very useful for debugging, setting up pre-scan, etc.

These are freeware, pay if you want and whatever you want. Just send an initialized disk for either one, or two disks (or SSDD or flippy) for both, in a returnable mailer with ENOUGH RETURN POSTAGE, to

Danny Michael,
Rt 9 Box 460
Florence, AL 35630.

John Hamilton of the Central Iowa Users Group will send you his 22-page boklet of "99 Tips" for the TI-99/4A, for just $4.00. The address is

John Hamilton,
4228 E. Clinton, Des Moines IA 50317.

I have been experimenting with TI-Writer, and this issue of the Tips is being printed in 4 columns, right justified directly from the printer. Here's how -

Use TI-Writer, editor mode, in any line length you want. The first line should be .RM 27;FI;AD but don't use any other formatter codes. Don't indent paragraphs. Use some other character as a temporary substitute for any ^, @, & or * in the text. Don't include any program listings, yet.

Save the file as DSK1.TEXT. Print an edit copy. Then go into formatter mode. Select DSK1.TEXT to be printed, but instead of your printer spec, type DSK1.TEXT2. Your file will now be in 28-column format and right justified, and indented.

If the text is to include any program listings, run them through my 28-Column converter (see Tips #18), using the Editor option of that program.

Go back to TI-Writer editor and load DSK1.TEXT2. Merge in the program listings. Then PF to print file, but instead of a printer spec, type C DSK1.TEXT3. When it has printed to disk, LF the DSK1.TEXT3 and you will find that all control characters are gone.

Now for a bit of editing. Delete the 3 blank lines at the beginning, and the 6 blank lines that have appeared after every 60th line. Center the title by erasing with the space bar and retyping - do NOT use FCTN 2! Also replace any temporary characters with the ^, @, & or *.

You will print 4 columns of 60 lines per page, so the total lines in your file must be a multiple of 240. Add enough blank lines to the end of the file to reach that count.

Save that file back to disk as DSK1.TEXT3. Now go into XBasic, key in this program and RUN!

```
100 OPEN #1:"DSK1.TEXT3",INP
UT :: OPEN #2:"PIO",VARIABLE
 255 :: PRINT #2:CHR$(15);CH
R$(27);CHR$(69):: DIM B$(240
)
110 FOR A=1 TO 2 :: FOR B=1
TO 240 :: LINPUT #1:B$(B)::
NEXT B
120 FOR C=1 TO 60 :: PRINT #
2:TAB(10);B$(C);TAB(41);B$(C
+60);TAB(72);B$(C+120);TAB(1
03);B$(C+180):: NEXT C :: PR
INT #2:CHR$(27);CHR$(97);CHR
$(6):: NEXT A :: CLOSE #1 ::
 CLOSE #2 :: END
```

The A loop is for a 2-page printout of 480 lines, of course.

You can modify this routine to print in 2 or 3 columns, adjust the margins, change the type font or size, rewrite for your own printer, etc. And the column width can be anything you want, just change that .RM 27 in the first line of the text (don't forget that the left margin is set at 0, not 1).

If you want a 2-column page, you can dump the file back to disk instead, and then print it out of TI-Writer editor. Use this routine, modified as you wish.

```
100 !Opens a file TEXT3 of 2
40 lines 35 char long and co
nverts it into a file which
can be printed out of TI-wri
ter Editor as 2 pages in 2 c
olumns
110 OPEN #1:"DSK1.TEXT3",INP
UT :: OPEN #2:"DSK1.TEXT4",O
UTPUT :: DIM B$(120)
120 FOR A=1 TO 2 :: FOR B=1
TO 120 :: LINPUT #1:B$(B)::
NEXT B
130 FOR C=1 TO 60 :: PRINT #
2:"     "&B$(C)&RPT$(" ",38-
LEN(B$(C)))&B$(C+60):: NEXT
C :: FOR D=1 TO 6 :: PRINT #
2:" " :: NEXT D :: NEXT A ::
 CLOSE #1 :: CLOSE #2
```

It is best to run a program to set up your printer, and leave it turned on, before printing that file out of the Editor. It is not at all easy to imbed control characters in the file, because they affect the line in all columns and also shift the lines out of alignment.

I understand that there a couple of kids who wait every month for their dad to key them in a bit of nonsense from the Tigercub, so -

```
100 !KEYZAP - by Jim Peterso
n
110 DISPLAY AT(6,11)ERASE AL
L:"KEYZAP" :: DISPLAY AT(12,
1):"   Zap the Zprite by typ
ing the key in the correspon
dingposition on the keyboard
."
120 DISPLAY AT(24,10):"Press
 any key" :: CALL KEY(0,K,S)
:: IF S=0 THEN 120
130 RANDOMIZE
140 CALL CHAR(47,"817EA58199
A5423C")
150 CALL CLEAR :: T=0 :: CAL
L FLASH(T)
160 CALL KEY(3,K,ST):: IF ST
=0 THEN 180
170 C=C+1 :: IF C=101 THEN 1
90 ELSE CALL KEYBOARD(K,T)
180 CALL MOTION(#1,25*RND-25
*RND,25*RND-25*RND):: CALL C
OINC(#1,#2,16,A):: IF A=0 TH
EN 160 ELSE CALL FLASH(T)::
GOTO 160
190 CALL DELSPRITE(ALL):: DI
SPLAY AT(12,9):"GAME OVER" :
```

```
: DISPLAY AT(14,9):"SCORE";T
 :: DISPLAY AT(16,9):"PLAY A
GAIN?"
200 CALL KEY(3,K,S):: IF S<1
 THEN 200
210 IF K=89 THEN C=0 :: GOTO
 150 ELSE END
220 SUB KEYBOARD(K,T)
230 IF FLAG=1 THEN 250 :: FL
AG=1Jim Peterson
240 KEY$="1234567890=QWERTYU
IOP/ASDFGHJKL;"&CHR$(13)&"ZX
CVBNM,."
250 IF (K=47)+(K=61)+(K=13)T
HEN SUBEXIT ELSE X=POS(KEY$,
CHR$(K),1):: Y=ABS(X>11)-(X>
22)-(X>33)+1 :: R=Y*6 :: C=(
(X+(Y>1)*(Y-1)*11)*3)
260 CALL SPRITE(#2,42,16,R*8
-7,C*8-7):: CALL COINC(#1,#2
,16,N):: IF N=0 THEN SUBEXIT
270 CALL FLASH(T):: SUBEND
280 SUB FLASH(T):: FOR W=1 T
O 10 :: CALL SCREEN(16):: CA
LL SCREEN(8):: NEXT W :: CAL
L SPRITE(#1,47,2,1,1):: T=T+
1 :: DISPLAY AT(1,20):T :: S
UBEND
```

And here's another -

```
100 ! QUICK & DIRTY DOODLER
      by Jim Peterson
Use joystick #1. Press fire
button to change color or
pattern, Enter to clear the
screen.
110 DATA FFFFFFFFFFFFFFFF,FF
,0101010101010101,0000000000
0000FF,8080808080808080,01020
4081020408,8040201008040201,
FF818181818181FF
120 CALL CLEAR :: FOR J=1 TO
 8 :: READ CH$(J):: NEXT J
130 FOR CH=32 TO 136 STEP 8
:: FOR CN=CH TO CH+7 :: X=X+
1 :: CALL CHAR(CN,CH$(X))::
NEXT CN :: X=0 :: NEXT CH ::
 CALL CHAR(32,"0")
140 CALL SCREEN(16):: FOR S=
2 TO 14 :: CALL COLOR(S,S+1,
1):: NEXT S :: R=12 :: C=16
:: CH=33
150 CALL HCHAR(R,C,CH):: CAL
L FASTJOY(C,R,Q):: IF Q=18 T
HEN CH=CH+1+(CH=143)*110
160 CALL KEY(0,K,S):: IF K=1
3 THEN CALL CLEAR :: GOTO 15
0 ELSE 150
170 SUB FASTJOY(C,R,Q):: CAL
L JOYST(1,X,Y):: CALL KEY(1,
Q,S):: X=SGN(X):: Y=-SGN(Y):
: C=C+X+(C=32)-(C=1):: R=R+Y
+(R=24)-(R=1):: SUBEND
```

And a pretty one -

```
100 CALL CLEAR :: CALL SCREE
N(2):: FOR S=2 TO 8 :: CALL
COLOR(S,15,1):: NEXT S :: DI
SPLAY AT(12,7):"KALEIDOSQUAR
ES" ! by Jim Peterson
110 FOR CH=40 TO 136 STEP 8
:: FOR L=1 TO 4 :: RANDOMIZE
 :: X$=SEG$("0018243C425A667
E8199A5BDC3DBE7FF",INT(16*RN
D+1)*2-1,2)
120 B$=B$&X$ :: C$=X$&C$ ::
NEXT L :: CALL CHAR(CH,B$&C$
):: B$,C$=NUL$ :: NEXT CH
130 FOR S=2 TO 14 :: X=INT(1
5*RND+2)
140 Y=INT(15*RND+2):: IF (Y=
X)+(Y=8)THEN 140
150 CALL COLOR(S,X,Y):: NEXT
 S
160 AR,R,AVR,VR=1 :: AC,C,AH
C,HC=4 :: TT=24 :: XX,XT=13
170 FOR L=1 TO 12 :: T=TT ::
 XT=XX :: R=AR :: VR=AVR ::
C=AC :: HC=AHC
180 FOR J=1 TO XT :: X=INT(1
3*RND+2)*8+24 :: CALL HCHAR(
R,HC,X,T):: CALL HCHAR(25-R,
HC,X,T):: CALL VCHAR(VR,C,X,
T)
190 CALL VCHAR(VR,31-C,X,T):
: T=T-2 :: HC=HC+1 :: VR=VR+
1
200 NEXT J :: AR=AR+1 :: AVR
=AVR+1 :: AC=AC+1 :: AHC=AHC
+1 :: TT=TT-2 :: XX=XX-1 ::
NEXT L
210 IF INT(2*RND)<>0 THEN 23
0
220 FOR S=INT(12*RND+2)TO 14
 :: CALL COLOR(S,1,1):: NEXT
 S
230 FOR J=1 TO INT(20*RND+1)
:: S=INT(13*RND+2):: X=INT(1
5*RND+2):: Y=INT(15*RND+2)::
 CALL COLOR(S,X,Y):: NEXT J
240 CALL SCREEN(INT(15*RND+2
)):: ON INT(5*RND+1)GOTO 130
,160,220,230,240
```

The challenge in Tips #16 was - how can you store a hundred or more values of any size, positive or negative, integer or non-integer, even in exponential notation, without dimensioning an array or opening a file, and then link to another program with a RUN statement and recover those values - not by reading them from the screen? I had just one reply! Was it too easy, too hard, or doesn't anyone care? Anyway -

```
20591 SUB CHARSAVE2(CH,N)::
N$=STR$(N):: N$=RPT$("0",16-
LEN(N$))&N$
20592 IF POS(N$,".",1)=0 THE
N 20593 :: N$=SEG$(N$,1,POS(
N$,".",1)-1)&"A"&SEG$(N$,POS
(N$,".",1)+1,LEN(N$))
20593 IF POS(N$,"+",1)=0 THE
N 20594 :: N$=SEG$(N$,1,POS(
N$,"+",1)-1)&"B"&SEG$(N$,POS
(N$,"+",1)+1,LEN(N$))
20594 IF N<0 THEN N$=SEG$(N$
,1,POS(N$,"-",1)-1)&"F"&SEG$
(N$,POS(N$,"-",1)+1,LEN(N$))
20595 CALL CHAR(CH,N$):: SUB
END
```

And to recover the values -

```
20596 SUB READCHAR(CH,N):: C
ALL CHARPAT(CH,CH$)
20597 IF POS(CH$,"A",1)=0 TH
EN 20598 :: CH$=SEG$(CH$,1,P
OS(CH$,"A",1)-1)&"."&SEG$(CH
$,POS(CH$,"A",1)+1,LEN(CH$))
20598 IF POS(CH$,"B",1)=0 TH
EN 20599 :: CH$=SEG$(CH$,1,P
OS(CH$,"B",1)-1)&"+"&SEG$(CH
$,POS(CH$,"B",1)+1,LEN(CH$))
20599 IF POS(CH$,"F",1)<>0 T
HEN CH$="-"&SEG$(CH$,POS(CH$
,"F",1)+1,LEN(CH$))
20600 N=VAL(CH$):: SUBEND
```

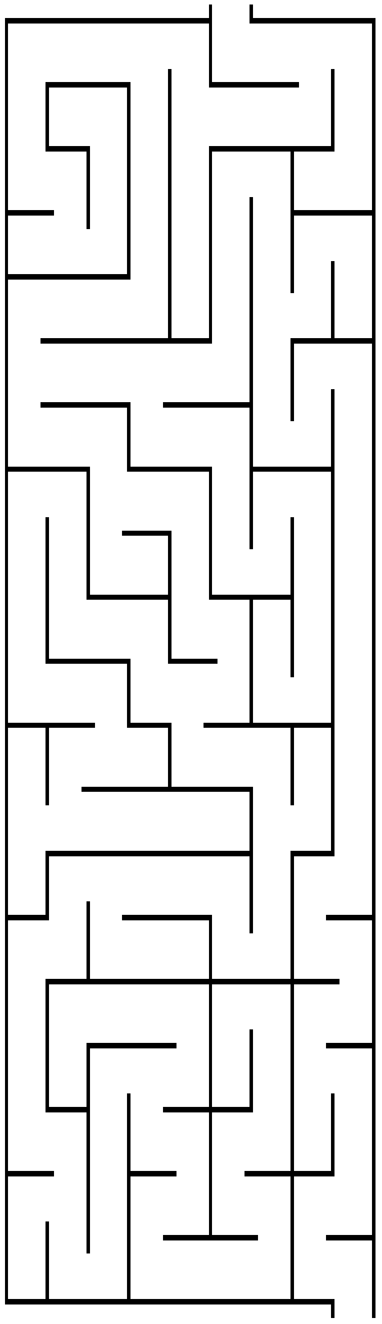
Here's a jewel of a routine from Danny Michael, to avoid those lockups and other foul-ups that occur when you CALL INIT after you have already CALLed INIT -

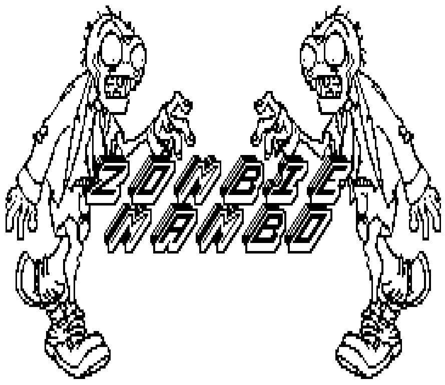
```
CALL PEEK(8198,A):: IF A<>17
0 THEN CALL INIT
```

The best way to edit a program is to type NUM and the first line number, then Enter will take you through line by line with no danger of accidentally deleting a line. The edit functions will still work, and FCTN 4 gets you out of the NUM mode.

MEMORY FULL!

Jim Peterson YN





By
Judy Sanoian

99'ER
October 1983
Volume 2
Number 12

The title "Zombie Mambo" may conjure up a macabre image of pale, hollow-eyed corpses shuffling in time to a Latin beat. But this game fulfills only half of its title's promise. There is, to be sure, a coterie ofsomnambular spirits, but these zombies don't exactly boogie to a disco beat, or even a sedate tango. They plod along slowly, as if they might be sporting leg irons.

Those who are wise in the ways of cyber-tainment will immediately nod and say "Ah, typical sluggish BASIC game." It can indeed be difficult to find Basic games that are fast, complex and challenging. In my opinion, Zombie Mambo meets all three of these criteria. First, it is a complex game, offering multiple screens and two separate scenarios. It even has elements of an adventure game in that you must progress through interconnecting chambers of a crypt, using clues to find your way out. Second, the game is extremely challenging on even its easiest level of difficulty. (I challenge anyone to make it to the crypt their first time through!) As for speed, Zombie Mambo's one game in which slow motion is appropriate for the scenario.The slow, jerky motion of Basic is perfectly suited to simulate the dead zombie shuffle.

Cantankerous Cadavers

There are two separate games in Zombie Mambo. Part 1 begins in the graveyard, a garish pink and green garden with a crypt reminiscent of the Taj Mahal in the background. Your assignment is to dig through the graves in search of three golden keys that will gain you access to the sorcerers crypt. As you dig, you will hear the realistic crunch of the shovel (and see a growing mound of dirt) each time you hit the fire hutton.

The sound effects are, as a matter of fact, my favorite feature in Zombie Mambo. Muffled, echoing footsteps evoke the feeling of the empty tomb. A door clangs shut to signal your entrance into the crypt. Best of all is the sound of the zombies themselves a strange cacophony of unearthly grunts, pig-like oinks, and, of course, the constant, dragging footsteps. There is also a syncopated, DEVO-esque tune (the mambo?) which serves as a nice counter point to the zombie grunts.

Once you have dug eight scoops of dirt, either a key, weapon or zombie will appear. If it's a weapon or key, hang on to it. Most likely though, it will be a zombie. The stumbling stiff will raise himself from his not-so-final resting place and begin stalking you. The zombie (and also your own character) can move either quickly or slowly, depending on the level of difficulty (1-9) you choose at the beginning of the game. This type of adjustable difficulty level is generally a good option to provide; it ensures that the game will remain challenging as you reach higher levels of proficiency. In the case of Zombie Mambo, however, I'm not sure it is really necessary. Part 1 was too difficult for us, even on level 9 (the easiest level). We played it all afternoon and never made it to the crypt. In fact, I was so frustrated after losing nine rounds in a row that I wanted a Zombie - the kind with rum in it!

Dead on Your Feet

Part of the problem may have been our limited mobility. It is impossible to move diagonally or turn about-face. To go from facing north to south, you must first make a quarter-turn. (Of course the zombies are similarly hampered.) Also, in Part 1, you are hemmed in by the walls of the graveyard. There is, however, one point in your favor: The zombies kill one another on contact. So if you are cornered, you can simply hover above the meandering mummies (they usually move in horizontal paths) and wait for one to "re-kill" his brother. Also, the zombies can stalk only while you are active. . . but remember they can attack while you are digging. So don't let down your guard when you pick up the shovel!

There are only three keys hidden in the cemetery, so your grave/gold-digger may have to rifle through every plot before finding them. You will usually also find a weapon which kills only one type of monster. We tried this "weapon" twice and it never worked. Instead, we got the message "Your weapon is ineffective." Even though we are warned in the manual that this might happen, it is frustrating to lose the game every time your weapon fails. It would have been less devastating to deduct a sum from our cash pile when our gun missed. As it was, we decided gambling on the weapon wasn't worth the risk.

Let s Make a Deal

Once you gather all three keys, you can enter. . . the crypt. This is the really engrossing part of the game. To start Part 2 you must reload the tape. Be sure to read the loading instructions or you might, as I did, flip the cassette before loading it. Part 2 offers the game's more cerebral challenge. You must find your way out of the labyrinthine tomb. On the way, you progress through several chambers (on separate screens), opening vaults to reveal either valuable cash prizes (yea!) or stalking monsters (boo). A different type of monster resides in

each chamber. You can enter and exit these rooms freely, but if you re-enter a room with monsters in it, you will he Killed instantly. You must discover a complex, secret pattern in order to escape the tomb. The documentation recommends making a map to Keep track of money, weapons and monsters as you explore the chambers. I would take this recommendation as an order. And even with a map, solving this maze will test your logic, patience and visual memory to the fullest.
By Jim Peterson

Gruesome Graphics

Zombie Mambo's graphics may not be spectacular, but they are unusual. The pale, grey zombies do look like dead shadows of former selves. Unfortunately, the piles of earth you dig from the grave are the same shade of grey, so the zombies and dirt piles tend to blend together. I'd prefer some rich "earth tones" for the unearthed debris. In Part 2, the tomb's monsters are imaginative, if somewhat odd. One resembles a pot-bellied woodstove, another a Hawaiian tiki god. I thought the graphics were OK, but I have one suggestion. It would be helpful (at least for those who have no adventure gaming experience) to somehow distinguish the tomb's chambers from one another in the initial, easy levels of the game. As it is, it s impossible to differentiate between the rooms without opening a vault and you are immediately Killed if you re-enter a room with remaining monsters! I'm not saying it can t be done; it's just a bit too much of a challenge for beginners.

Crypt-ic Documentation

The documentation is - excuse the expression - strictly "bare bones." Loading and running are adequately explained, as is how to maneuver your grave-digger. But Part 2 of the game baffled me, and the instructions offered no clues. How, I wondered, do I distinguish one room from another? Do I have to get all the money before I can exit? For those who have never played an adventure game, the modus operandi can be difficult to figure out. For these novices, a short explanation of how to find your way through the tomb would cut down on some of the frustrating trial and error. The game is certainly challenging enough without concealing how it's done. The instructions also failed to note that your disk drive must be disconnected before the program will run. It seems that many manuals fail to mention this important point.

Despite these minor flaws, I recommend Zombie Mambo highly, especially for those of you who are looking for a game in BASIC. Even though Part 1 proved impossible for any of us to master, Part 2 more than made up for the disappointment. It is a really tough game which will remain a challenge, both physically (dodging zombies) and mentally (navigating the tomb), long after your first foray into the crypt. There is a lot to Zombie Mambo - multiple screens, multiple levels of difficulty, and multiple goals. The game is also successful in conveying its scenario, The clanging door of the tomb, echoing footsteps, unearthly grunts, and pale, plodding figures all evoke the home of the ungrateful undead. So if Halloween has you hankering for some ghouling around, check out Zombie Mambo. But be forewarned - the game is Difficult. Whatever you do, don't vow to play until you win........ or you may experience your own night of the living dead and end up feeling a lot more zombie than mambo. YN

October 1991 - Vol. 8, No. 9
By Stan Krajewski

This is a program that had good intentions but didn't quite turn out. System requirements are disk drive, 32K, Extended BASIC and a joystick.

4-Wheelin' is a one player game where you are a monster truck out to beat the computer-operated truck through a mass of other vehicles. The object of the game is to get to the finish line first so you can earn another truck and go to the next level. Your lives, score, level and distance are displayed on the right side of the screen.

The reason I gave two stars is that the level of difficulty is high enough to make the game a bit of a challenge. However, the sounds are annoying and sound as if the console is malfunctioning. The graphics are better than some but there are flaws as they jump around the screen instead of flowing smoothly. Also, when the computer truck is crashed into it remains partially on the screen as a new one is generated.

The game operates at a decent speed at restart and is easy to continue with a press of the joystick button. This comes in handy as you will find yourself restarting quite often and do not have to reach for the Keyboard.

The programming expertise is obvious but the program could have benefitted from a longer stay on the drawing board. This game is suitable for younger members of the household for the price as is. The adults might want to take a go at it. To order send $4 plus $1.50 S&H to: Baker Software.

KNOW-WARE SYSTEM TESTS

CONSOLE TEST

The test begins with the VDP RAM test. All ram bits are turned on and off and checked for both conditions. A special address error test is used to make sure there aren't any addressing errors. When the test begins the screen will flash to blue then yellow then transparent. If an error is found during this time the computer screen will turn red and will print out the board layout number of the RAM in question. The rams are numbered in sequence from U102 thru U109. The ram nearest the edge of the board is U102, and the ram furthest from the edge of the board is U109. If all is well the computer will print "VDP RAMS good".

Next the computer checks the console GROMS by combining the data in each GROM and getting a signature unique to that GROM. If the signature is not correct the computer will print out the chip number for the GROM. This will be either 2155, 2156, or 2157 GROM gives bad data. If no error is found the computer will print "Console GROMS good".

Next is the scratch-pad RAM test. Every bit in the pad is turned on and off and verified. If an error is found the computer will print either LSB or MSB scratchpad RAM gives bad recall.

Next is the console ROM test. The computer combines all the data in the console ROM to get a signature. If the signature is not correct the computer prints either LSB or MSB ROM gives bad data. If the signature is correct the computer prints "Console ROMS good".

After these tests the computer begins the graphics and sound tests. First the computer fills the screen with ASCII characters and then shows the same number of characters in text mode. The computer will hold this screen for a few seconds. If you want to examine this screen longer you can press the space bar and the computer will hold this screen until you press the space bar again.

Next comes the multicolor test. The multicolor test consists of 8 rows of alternating colors. These 8 rows are repeated 8 times until the screen is filled. The first row is medium red and transparent. The second row is light red and black. The third row is medium green and dark yellow. The fourth row is light green and light yellow. The fifth row is dark green and dark blue. The sixth row is light blue and magenta. The seventh row is dark red and gray. the eighth row is cyan and white. The background color will change to all the possible colors during this test. You can hold any screen by pressing the space bar during the screen you want. Pressing the space bar again will move the computer to the next test.

Next is the bit map test. A light green diamond bordered on the top two sides by white and the bottom two sides by blue will fill every screen position. A transparent dot will also be visible in the center of the diamond. The screen background color will switch to all the screen colors. Pressing the space bar during this time will hold the screen. Pressing the space bar again will cause the computer to move to the next test.

The next test is the sound test. The computer checks all three sound and both noise channels at descending amplitudes. The graphics and sound tests require user verification to determine correctness.

The last test is a sprite test which checks both coincidence and fifth sprite. A vertically descending white sprite will coincide with a stationary magenta sprite and then will move between four sprites colored blue, red, green and yellow to become a fifth sprite. The word "coincidence" will be displayed during the coincidence. The words "fifth sprite" will be displayed when the white sprite is the fifth sprite. This sequence will be completed two times. After this the computer will go back to the beginning and start over with the test. The test will continue to cycle unless an error is found in GROM, ROM, or RAM. If an error is found the screen will turn red and the error will be reported.

SPEECH TEST

The test begins by speaking 3 words from its resident memory. The 3 words are "BLACK", "FROM", and "ZERO". The words "THAT IS RIGHT" are then spoken by using the external speech function of the speech synthesizer. The data in the two phrase ROMs is then checked. The words are used to check the ability to generate recognizable speech output. A problem with a word or part of a word generally indicates a faulty synthesizer chop. The external speech function is used to check the ability to input external speech data and generate recognizable speech. A problem with external speech generally indicates a problem with the synthesizer chip or the computer-synthesizer interface.

The entire speech synthesizer ROM memory is then checked for correctness of data. If the data read from the ROM memory is zero at all locations, then it is likely that the synthesizer chip is faulty and unable to properly address and read from the ROM memory chips and the words "UNABLE TO READ" will be displayed. The previous test of the words should have resulted in no speech output for the first 3 words. If the data is incorrect, but not zero, then it is likely that the ROM memory chips are defective and the word "BAD" will be displayed. If the memory check is good then the word "GOOD" will be displayed.

MEMORY EXPANSION TEST

The memory expansion test is available only if you are using a Mini Memory. It will be the third option available on the menu. The memory expansion test turns every bit in the 32K memory expansion on and off and checks for both conditions. It also checks for addressing errors. Peterson

In most memory expansions there are two block of 8 16K bit rams. One block is for memory addresses >2000 thru >3FFF and >A000 thru >BFFF. The other block is for addresses >C000 thru >FFFF.

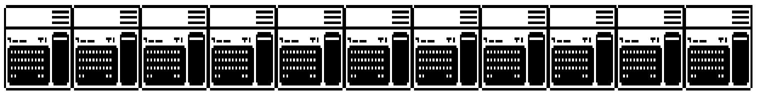
There are two test options on the memory expansion test. One continuously loops checking every bit in the memory expansion. While it loops a counter at the center of the screen counts the cycles in hexadecimal. The other option is the refresh test. When this option is chosen a prompt will appear asking "HOW MANY MINUTES?". Enter a four digit decimal number from 0001 to 9999. Every bit in the memory will be turned on and the computer will then wait the chosen number of minutes until it checks to see if any bits have decayed to 0. In most cases entering 0005 minutes will be sufficiant to test the memory expansion.

If you have a memory expansion that you feel looses a bit only on occasion, then you can put in enough time to last over night. At any time if an error occurs when the memory is checked the screen will turn red, the address the error was found at, and the expected and actual data are displayed on the screen. Also the words "POSSIBLE BAD RAM BIT _" is displayed. The bad bit can be from bit 0 to bit 7. Keeping with the TI tradition bit 0 is the most significant bit and bit 7 is the least significant bit. Here the address at which it failed is important because that determines which bank of rams is at fault.

THE PURPOSE OF THESE TESTS

If the computer passes these tests then the user can be reasonably sure that his or her system is functioning properly. If any test fails and a diagnostic indication is given then the chances are good that the diagnostic information is right. However if certain compponents fail an erroneous diagnosis may result. An example would be a bad microprocessor or a bad ram in the memory expansion (where the program resides) and some hardware problems.

We believe these tests are the best that can be done with software alone. Also the computer must be able to load and run programs before the tests can even begin to work.



BLOCK BUSTER

```
10 @=1 :: _=2 :: CALL CLEAR
:: GOSUB 240
20 A=@ :: B=9 :: C=[ :: FOR
D=5 TO 7 :: CALL COLOR(D,16,
@):: NEXT D
30 FOR E=3 TO 4 :: CALL COLO
R(E,16,@):: NEXT E :: CALL C
OLOR(9,15,@,11,16,@):: CALL
CLEAR :: A$="FFFFFFFFFFFF" :
: CALL CHAR(39,A$,42,A$)
40 CALL CHAR(94,A$,96,A$,97,
"FFFFFFFFFFFFFFFF",114,"3C7E
FFFFFFFF7E3C")
50 DISPLAY AT(3,4):"' ' ' '
' ' ' ' ' ' '" :: CALL COLOR
(@,13,@,_,7,@):: DISPLAY AT(
6,4):" * * * * * * * * * * *
" :: DISPLAY AT(9,4):"^ ^ ^
^ ^ ^ ^ ^ ^ ^ ^"
60 DISPLAY AT(12,4):" ^ ^ ^
^ ^ ^ ^ ^ ^ ^ ^" :: CALL COL
OR(8,5,@):: F=23 :: G=15 ::
H=3 :: I=12 :: J,K=@ :: CALL
 HCHAR(F,G,96,3):: CALL HCHA
R(24,@,97,32)
70 DISPLAY AT(@,4):"BALLS:";
B;TAB(17);"SCORE:";C
80 CALL KEY(@,L,M):: IF L=_
THEN 90 ELSE IF L=3 THEN 110
 ELSE 130
90 G=G-_ :: IF G<@ THEN G=@
100 CALL HCHAR(F,G+_,32,3)::
 CALL HCHAR(F,G,96,3):: GOTO
 130
110 G=G+_ :: IF G>29 THEN G=
29
120 CALL HCHAR(F,G-_,32,3)::
 CALL HCHAR(F,G,96,3)
130 H=H+J :: I=I+K :: CALL G
CHAR(I,H,N):: IF N=96 THEN 1
70 ELSE IF N=97 THEN 180 ELS
E IF N=94 OR N=39 OR N=42 TH
EN 200
140 IF H<_ OR H>30 THEN 160
ELSE IF I<_ THEN 170
150 CALL HCHAR(I,H,114):: CA
LL HCHAR(I,H,32):: GOTO 80
160 J=-J :: CALL SOUND(30,38
0,_):: IF I>@ THEN 80
170 K=-K :: CALL SOUND(30,38
0,_):: GOTO 80
180 B=B-@ :: I=12 :: H=3 ::
FOR O=@ TO 500 :: NEXT O ::
IF B=[ THEN 220
190 DISPLAY AT(@,4):"BALLS:"
;B;TAB(17);"SCORE:";C :: GOT
O 80
200 CALL HCHAR(I,H,32):: C=C
+21 :: A=A+@ :: IF A=45 THEN
 CALL HCHAR(F,G,32,3):: A=@
:: GOTO 50
210 DISPLAY AT(@,23):C :: GO
TO 170
220 DISPLAY AT(@,4):"BALLS:"
;B;TAB(17);"SCORE:";C :: CAL
L HCHAR(_,@,32,568):: FOR D=
_ TO 8 :: CALL COLOR(D,16,@)
:: NEXT D :: DISPLAY AT(23,@
):" PRESS REDO(R) OR QUIT(Q)
."
230 CALL KEY([,L,M):: FOR O=
@ TO 100 :: NEXT O :: DISPLA
Y AT(12,7):"G A M E  O V E R
" :: FOR O=@ TO 100 :: NEXT
O :: DISPLAY AT(12,7):: IF L
=81 THEN STOP ELSE IF L=82 T
HEN 20 ELSE 230
240 CALL SCREEN(_):: CALL CH
AR(42,"FFFFFFFFFFFF"):: PRIN
T "***##*#****#****#*":"*##*
#*#*##*#*##*#*##*":"*##*#*#*
##*#*####*#*":"***##*#*##*#*
####**"
250 PRINT "*##*#*#****#*####
*#*":"*##*#*#####*##*#*##*"
:"***#*****#***#*###*": :
 :"***##*##*#***#*****#***#*
**":"*##*#*##*#*#####*###*##
#*"
260 PRINT "*##*#*##*#***###*
###**##*":"***##*##*###*###*
###*####*":"*##*##**##***###*
###***":"*##*":"***": : : :
:: CALL COLOR(_,16,@):: FOR
D=@ TO 20 :: CALL SOUND(-100
,-3,D):: NEXT D
270 FOR D=20 TO @ STEP -@ ::
 CALL SOUND(-1000,-3,D):: NE
XT D :: FOR H=3 TO 16 :: CAL
L COLOR(_,H,@):: FOR I=@ TO
50 :: NEXT I :: NEXT H :: RE
TURN
```

Yesterday's News Information

Yesterday's News is a labor of love offered as a source of pleasure & information for users of the TI-99/4A and Myarc 9640 computers.

TI-99/4A HARDWARE

TI99/4A COMPUTER
MODIFIED PEB
WHT SCSI AND SCSI2SD
MYARC DSQD FDC
MYARC 512K MEMORY
HORIZON 1.5 MEG HRD
TI RS232
CORCOMP TRIPLE TECH
1 360K 5.25 DRIVE
1 360K 3.50 DRIVE
1 720K 5.25 DRIVE
1 720K 3.50 DRIVE

TI-99/4A SOFTWARE

PAGEPRO 99
PAGEPRO COMPOSER
PAGEPRO FX
PAGEPRO HEADLINER
PAGEPRO GOFER
PAGEPRO FLIPPER
PAGEPRO ROTATION
PIXPRO
PICASSO PUBLISHER
BIG TYPE
TI ARTIST PLUS
GIF MANIA

PC HARDWARE

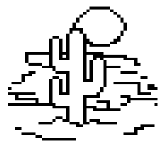
COMPAQ ARMADA 7800
COMPAQ ARMADASTATION
SAMSUNG SYNCMASTER

PC SOFTWARE

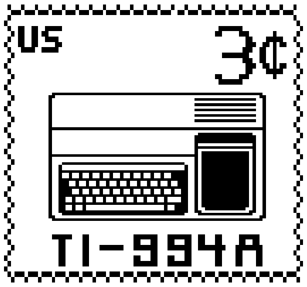
DEAD WINDOWS 98SE
FILECAP
PRN2PBNS
IRFANVIEW
ADOBE DISTILLER
ADOBE ACROBAT

Yesterday's News is composed entirely using a TI-99/4A computer system. It consists of 11 PagePro pages which are "printed" via RS232 to PC to be published as a PDF file.

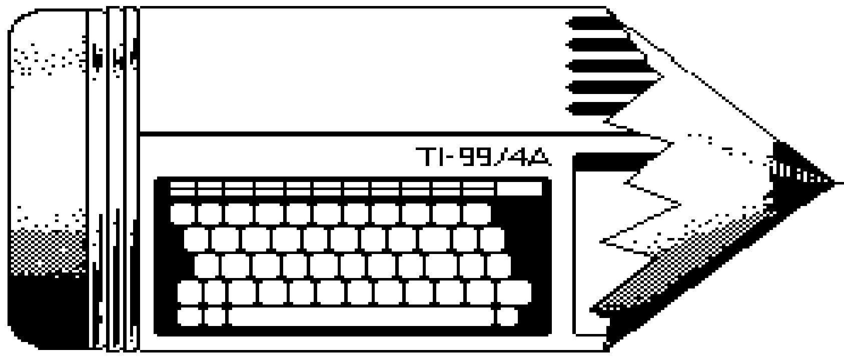
TI-99/4A



Yesterday's News
c/o SparKdrummer
AtariAge Forum
Phoenix, AZ 85027



TI-99/4A Computer User
1234 What Me Worry Lane
Any City,Any State,Any Country



COMING NEXT MONTH

HOPPER
TAG TOM AND FIRE
WILD CATTING
DADDIE'S HOT ROD
STAR VENTURE