How COMPUTE! Readers Use Their Computers

COMPUTE!

$2.95
February
1984
Issue 45
Vol. 6, No. 2

$2.25 UK $3.25 Canada
C2193
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

Special Games Issue

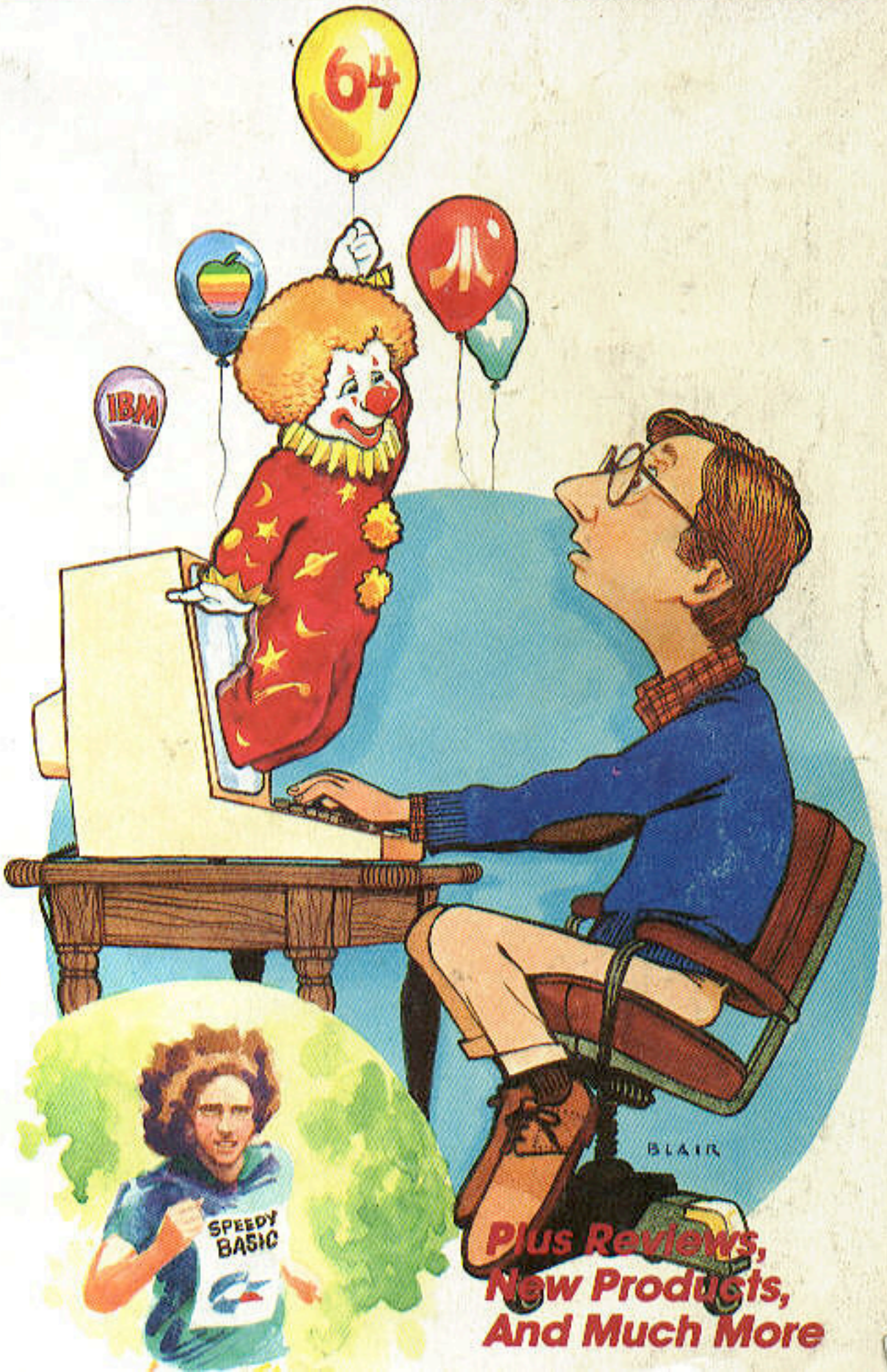
Now Including The IBM PC

Circus For VIC-20, Commodore 64, Atari, And TI-99/4A

3-D Drawing Master For Commodore 64, Apple, And VIC-20

Quatrainment For IBM PC, Atari, TI-99/4A, VIC-20, Commodore 64, Apple, And Radio Shack Color Computer

Speedy BASIC For VIC-20 And Commodore 64



Plus Reviews, New Products, And Much More

0 71486 02193 02

February 1984 Vol. 6, No. 2

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 180 before typing in programs.

GUIDE TO ARTICLES AND PROGRAMS

	•
	•
	•
V/64/AT/TI	
AT/V/64/TI/AP/PC/C	
V/64/AT	
P/64	
AT/V/TI	
AT/64	
AP/AT	
	•
	•
	•
	•
	•
	•
	•
	•
	•
AT	
TI	
64	
V/64/AP	
C/AP	
64/V	
P/V/64	
64	

AP Apple **AT** Atari, **P** PET/CBM, **V** VIC-20, **C** Radio Shack Color Computer, **64** Commodore 64, **TS** Timex/Sinclair, **TI** Texas Instruments, **PCjr** IBM PCjr, *All or several of the above.

TOLL FREE Subscription Order Line
800-334-0868 (In NC 919-275-9809)

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies

One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published 12 times each year by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 505 Edwardia Drive, Greensboro, NC 27409. Domestic Subscriptions: 12 issues, $24.00. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 914, Farmingdale, NY 11737. Second class postage paid at Greensboro, NC 27403 and additional mailing offices.

ISSN 0194-357X.

on your computer. This will cause the disk to boot (load and run automatically). If you turn on the computer and drive without a disk in the drive, the computer will keep trying to read the nonexistent disk until you insert a bootable disk.

Will The VICmodem Work With PET And 64?

I have a VICmodem. I was wondering what would happen if I plugged the modem into a 64 or Commodore PET. If it wouldn't work, what are the differences between the VIC's User Port and the others?

Seth Major

The VICmodem will work with either the VIC-20 or the Commodore 64. The User Ports of these two computers are nearly identical. The PET/CBM series computers have a User Port also, and that port has the same edge connector as the VIC and 64. The bottom row of pins (A–N) has essentially the same configuration on all Commodore products. However, the top row of pins (1–12) is totally different on the PET/CBM. Also, the VIC and 64 have routines in their operating system ROMs which support serial communication through the User Port. The PET/CBM does not support in ROM any type of communication through its User Port. The VICmodem cannot be used with PET/CBM models.

Whiz Kids And The Real World

I have an Atari 800 computer and I wanted to know if I could connect my 800 with other computers to make one big system. I would also like to make a computer system that would be able to talk and do many programs like they do on the show *Whiz Kids*.

David Smith

The TV show Whiz Kids *is sometimes unrealistic. One episode had them turning on the sprinkler system in an effort to escape a room they had been locked into (of course the room had a computer terminal in it). There seemed to be no concern that the water cascading from the ceiling would almost instantly short out the terminal they were using.*

Nevertheless, the show does feature current technology and trends. With a telephone modem and terminal software, you can call up other computers and exchange information. There are many small bulletin boards which distribute information and even free programs. You can also subscribe to information utilities, such as The Source and CompuServe. With a telephone and a modem, most computers can communicate with each other.

You can also buy a speech synthesizer, such as the Alien Group's VoiceBox, SAM *from Don't Ask Software, or the Votrax* Type 'n Talk. *These devices let you program your computer to talk, but do not give the computer any capability for real conversation. Even the most advanced artificial intelligence experts haven't solved that problem yet.*

TI-99/4A Disk Drives

Is there any company that makes a TI-99/4A disk drive that does not require accessories?

Eric Chet

TI, in fact, manufactures a "stand-alone" disk drive which has a built-in disk controller, and thus doesn't require the Peripheral Expansion System. This particular drive actually costs a little more than the disk controller card and the disk drive which are housed in the Peripheral Expansion System. If you are thinking of later expanding your system, you may want to spend the difference on the Peripheral Expansion System since TI no longer manufactures any other peripherals which work independently of this unit.

VIC-20 Static

I own a VIC-20. I use it on a black-and-white television. No matter how much fine-tuning I do, I can't get sound other than static. Is there any way this can be corrected? I have the RF Box without any adjusting screw inside.

Chris Nelson

Could be a bad RF modulator. Try another.

64 Lockup Bug

I am having a problem with my Commodore 64 which occurs when I use the DEL key to edit a BASIC program. While I am deleting, the program that I am working on RUNs! Even if I try to use the RUN/STOP key, it keeps going. Then, when it gets to an INPUT statement, the computer will not accept any keys. When I hit a key, even RETURN, nothing happens. The cursor just stays still and blinks.

R. Kasturi

You are describing the infamous 64 keyboard lockup bug. It occurs when you enter a long line at the bottom of the screen, then back up to edit it. There is no way to recover from this lockup short of shutting your machine off, then on again. If you scroll your line up before you try to DEL any characters, there should be no problem, as this lockup will only occur at the very bottom of the screen.

De Re Atari Error

Attention! Anyone owning *De Re Atari,* I have found an error in it. In the section on Display List

CIRCUS

Craig Setera

You don't have to run away to join the circus. Here's your chance to be Head Clown and pop clouds of balloons by catapulting your fellow clowns all around the big top. Originally written for the unexpanded VIC; versions also are included for 64, Atari, and TI-99/4A. Joystick required for the VIC, 64, and Atari.

The circus has closed for the evening. It's your job to help the clowns remove the cloud of balloons from the ceiling of the big top, by catapulting them into the air so they can pop all the brightly colored spheres. But a prankster has released one balloon filled with laughing gas. If one of the clowns pops the laughing gas balloon, he's out of the game. Even a clown can't bounce and pop and giggle at the same time.

When the laughing gas balloon is yellow, it's safe to pop, and you are awarded 250 points. But if it's black, watch out. Each row of balloons has a different point value. The blue (bottom) row is worth 50 points, the green (middle) row is worth 75 points, and the red (top) row has a point value of 100.

Whenever you catch a clown on your board, you get 5 points. An extra man is awarded for every 2000 points.

To start the game, press the fire button on the joystick. As you play, you will notice a block moving left to right below the balloons. This block will cause your clown to rebound in the direction from which he came. There also are two platforms, one on each side, that keep the clown within reach of your teeter board. The platforms can be thought of as constructed with upside-down trap doors. When a clown lands on top, he will bounce back, but if he hits the bottom, he will pass through.

"Circus" is a two-part program for the unexpanded VIC. The first program contains the custom character data and play instructions. When it is finished running, it will NEW itself, so be sure to SAVE it before you RUN it. The second program contains the game. You must LOAD and RUN the second program after running the first program, because the first program sets up the special graphics characters for the second program.

Program 1: Circus, VIC Loader

```
20 PRINT"{CLR}{8 DOWN}{8 SPACES}{RED}C
   {CYN}I{PUR}R{GRN}C{BLU}U{RED}S{BLK}":C
   =0
30 PRINT"{DOWN}{3 RIGHT}PLEASE HOLD ON...
   ":GOTO50000
40 GOSUB1000
45 POKE36869,255
50 PRINT"{CLR}{11 DOWN}{8 RIGHT}{BLK}CIRC
   US"
60 PRINT"{HOME}{9 DOWN}{6 RIGHT}########
   #":PRINT"{3 DOWN}{6 RIGHT}#########"
70 PRINT"{HOME}{10 DOWN}{6 RIGHT}#{DOWN}
   {LEFT}#{DOWN}{LEFT}#{2 UP}{8 RIGHT}#
   {DOWN}{LEFT}#{DOWN}{LEFT}#"
80 GOSUB2000
85 FORT=1TO3:FORP=1TO7:S=P
90 FORI=38604TO38614:POKEI,S
100 S=S+1:IFS=1THENS=2
110 IFS>7THENS=0
120 NEXT
130 FORI=38635TO38679STEP22:POKEI,S
140 S=S+1:IFS=1THENS=2
150 IFS>7THENS=0
160 NEXT
170 FORI=38702TO38692STEP-1:POKEI,S
180 S=S+1:IFS=1THENS=2
190 IFS>7THENS=0
200 NEXT
210 FORI=38670TO38626STEP-22:POKEI,S
220 S=S+1:IFS=1THENS=2
230 IFS>7THENS=0
240 NEXT
250 NEXTP,T
252 GOSUB1000:PRINT"{CLR}{9 DOWN}";
253 POKE36869,240:PRINT"{RIGHT}INSTRUCTIO
    NS (Y/N)?";:GOSUB2000
254 GETA$:IFA$=""THEN254
255 IFA$="Y"THEN260
```

```
1290 IF PEEK(764)=255 THEN 1290
1291 GRAPHICS 0
1295 GOTO 5
1300 DIM T(2),TA$(120),TB$(120),TC$(
     120),TD$(120),G$(25)
1301 IF PEEK(106)=155 THEN CHSET=(PE
     EK(106)+1)*256:GRAPHICS 17:POKE
      756,CHSET/256:RETURN
1305 POKE 106,PEEK(106)-5:GRAPHICS 1
     7
1307 POSITION 5,5:? #6;"redefining"
1308 POSITION 5,10:? #6;"CHARACTERS"
1309 POSITION 4,15:? #6;"PLEASE  WAI
     T"
1310 CHSET=(PEEK(106)+1)*256
1315 POKE 756,CHSET/256
1320 FOR X=0 TO 1023:POKE CHSET+X,PE
     EK(57344+X):NEXT X
1330 FOR I=24 TO 111:READ X:POKE CHS
     ET+I,X:NEXT I
1370 RETURN
1380 DATA 28,62,47,63,63,126,96,0
1390 DATA 58,58,18,124,16,56,68,68
1400 DATA 128,64,32,16,24,28,38,37
1410 DATA 128,64,32,16,8,4,2,1
1420 DATA 1,2,4,8,24,56,100,164
1430 DATA 1,2,4,8,16,32,64,128
1440 DATA 0,0,0,255,24,24,36,36
1450 DATA 0,0,0,255,0,0,0,0
1460 DATA 170,85,170,85,170,85,170,8
     5
1470 DATA 255,255,255,255,255,255,25
     5,255
1480 DATA 170,85,170,85,170,85,170,8
     5
```

Program 5: Circus, TI-99/4A Version

```
10 DIM D1(20),E(20):: RANDOMIZE ::
   GOTO 110
20 REM BOING!
30 FOR VOL=1 TO 30 STEP 6 :: CALL S
   OUND(-50,VOL+110,VOL):: NEXT VOL
    :: RETURN
40 REM SCORE
50 CALL HCHAR(ROW+DY,COL+DX,32):: C
   ALL SOUND(10,-5,1):: SC=SC+(H=12
   0)*-50+(H=112)*-75+(H=104)*-100+
   ((H=128)*(M1=1)*250):: BAL=BAL+1
60 IF BAL=84 THEN 150
70 IF (M1=0)*(H=128)THEN GOSUB 820
   ELSE DISPLAY AT(1,10):STR$(SC);
80 M1=INT(RND*2):: IF M1=1 THEN CAL
   L COLOR(13,11,1)ELSE CALL COLOR(
   13,2,1)
90 IF (SC>(2000*VAR))THEN LIFE=LIFE
   +2 :: VAR=VAR+1 :: G1=1 :: GOSUB
    820 :: G1=0
100 RETURN
110 GOSUB 430
120 G$=" abc "
130 VAR=1 :: SC=0
140 LIFE=3 :: V(0)=-1 :: V(1)=0 ::
    V(2)=1
150 Y=23 :: X=13 :: COL=16 :: BAL=0
     :: CALL CLEAR :: CALL SCREEN(1
    5)
160 M1=INT(RND*2):: IF M1=1 THEN CA
    LL COLOR(13,11,1)ELSE CALL COLO
    R(13,2,1)
170 CALL HCHAR(2,2,100,30):: CALL H
    CHAR(24,2,102,30):: CALL VCHAR(
    3,2,101,21):: CALL VCHAR(3,31,1
    01,21)
180 CALL HCHAR(3,3,104,28):: CALL H
    CHAR(4,3,112,28):: CALL HCHAR(5
    ,3,120,13):: CALL HCHAR(5,19,12
    0,12)
190 DISPLAY AT(1,3):"SCORE:";SC;TAB
    (18);"LIVES:";LIFE
200 CALL HCHAR(17,3,103,7):: CALL H
    CHAR(17,24,103,7)
210 DISPLAY AT(Y,X):G$
220 CALL MAGNIFY(3):: FOR T=1 TO 3
    :: CALL SPRITE(#T,136,2+RND*12,
    RND*90+50,120,0,RND*20+10):: NEX
    T T
230 F=RND*27+3 :: CALL HCHAR(3,F,12
    8)
240 FOR I=19 TO 5 STEP -1 :: CALL H
    CHAR(I+2,COL,32,3):: CALL HCHAR
    (I+1,COL,32,3):: CALL HCHAR(I,C
    OL,120,3):: CALL HCHAR(I+1,17,9
    6):: NEXT I
250 DISPLAY AT(8,4):"PRESS ANY KEY
    TO START"
260 CALL KEY(0,K,S):: IF S=0 THEN 2
    60
270 FOR G5=4 TO 27 :: CALL HCHAR(8,
    G5,32):: FOR J1=1 TO 10 :: NEXT
     J1 :: NEXT G5
280 ROW=6 :: COL=COL+1 :: DY=1 :: D
    X=0
290 CALL KEY(0,K,S):: IF (K<>44)*(K
    <>46)THEN 320
300 X=X+(K=44)+SGN(24-X)*(K=46)*-1
310 DISPLAY AT(23,X):G$
320 CALL GCHAR(ROW+DY,COL+DX,H):: I
    F H=32 THEN 400
330 IF H=101 THEN CALL HCHAR(ROW,CO
    L,32):: DX=-DX :: GOTO 290
340 IF (H=103)*(DY=-1)THEN CALL HCH
    AR(ROW,COL,32):: ROW=ROW-2 :: C
    ALL GCHAR(ROW,COL+2*DX,H):: COL
    =COL+2*DX+SGN(DX)*(H=101):: GOT
    O 320
350 IF (H=100)+((H=103)*(DY=1))THEN
     DY=-DY :: DX=V(INT(RND*3)):: G
    OSUB 30 :: GOTO 320
```

A flock of birds provides a distraction in the TI version of "Circus."

```
360 IF (H>96)*(H<100)THEN DX=V(H-97
    ):: DY=-DY :: GOSUB 30 :: GOTO
    400
370 IF (H=102)THEN GOSUB 820
380 IF ((H=104)+(H=112)+(H=120)+(H=
    128))*(FL=1)THEN GOSUB 50 :: GO
    TO 290
390 IF (H=104)+(H=112)+(H=120)+(H=1
    28)THEN GOSUB 50 :: DY=1 :: GOT
    O 290
400 CALL HCHAR(ROW,COL,32):: ROW=RO
    W+DY :: COL=COL+DX
410 CALL HCHAR(ROW,COL,96)
420 GOTO 290
430 FOR I=0 TO 3 :: CALL CHAR(104+I
    *8,"1C3E2F3F3F7E6000"):: NEXT I
440 CALL COLOR(10,7,1):: CALL COLOR
    (11,13,1):: CALL COLOR(12,5,1)
450 FOR I=96 TO 99 :: READ A$ :: CA
    LL CHAR(I,A$):: NEXT I
460 DATA 3A3A127C10384444,FF3030484
    8848484,FF0000000000000
470 DATA FF0C0C1212212121
480 CALL CHAR(136,"1C0F0703C1EF7F7F
    3F0F03070F1E3800000C0DEFAFEFFF
    CF0E0C0C080000000")
490 FOR I=100 TO 103 :: CALL CHAR(I
    ,"AA55AA55AA55AA55"):: NEXT I
500 FOR T=1 TO 20 :: READ D1(T),E(T
    ):: NEXT T
510 DATA 200,523,200,494,100,466,10
    0,494,100,466,100,440,200,415,2
    00,392,200,370,200,392
520 DATA 200,440,200,392,100,370,10
    0,392,100,370,100,349,200,330,2
    00,311,200,294,200,311
530 F=1 :: F1=7 :: F2=13 :: F3=5 ::
     T=14
540 CALL CLEAR :: CALL SCREEN(15)
550 DISPLAY AT(8,10):"hpxhpxhpxh" :
    : DISPLAY AT(9,10):"p
    (8 SPACES)x" :: DISPLAY AT(10,1
    0):"h CIRCUS p"
560 DISPLAY AT(11,10):"x(8 SPACES)h
    " :: DISPLAY AT(12,10):"hpxhpxh
    pxh"
570 FOR R=1 TO 20 :: CALL COLOR(10,
    F1,F,11,F2,F,12,F3,F)
580 T=F1 :: F1=F2 :: F2=F3 :: F3=T
    :: CALL SOUND(D1(R),E(R),2):: N
    EXT R
590 DISPLAY AT(22,3):"INSTRUCTIONS
    (Y/N)?" :: ACCEPT AT(22,23)VALI
    DATE("YN"):A$
600 IF A$="N" THEN RETURN
610 CALL CLEAR :: PRINT "THIS IS TH
    E GAME OF CIRCUS."
620 PRINT :: PRINT "THE OBJECT OF T
    HE GAME IS"
630 PRINT :: PRINT "TO POP ALL OF T
    HE BALLOONS"
640 PRINT :: PRINT "WITH THE MAN. T
    O CATCH THE"
650 PRINT :: PRINT "MAN, POSITION T
    HE TRAMPOLINE"
660 PRINT :: PRINT "WITH THE '<' AN
    D '>' KEYS."
670 PRINT :: PRINT "ALL BORDERS, BU
    T THE BOTTOM,"
680 PRINT :: PRINT "WILL BOUNCE THE
     MAN BACK."
690 PRINT :: PRINT :: PRINT "AN EXT
    RA MAN WILL BE AWARDED"
700 PRINT :: PRINT "FOR EVERY 2000
    POINTS." :: PRINT :: PRINT :: P
    RINT "PRESS ANY KEY TO CONTINUE
    "
710 CALL KEY(0,K,S):: IF S=0 THEN 7
    10
720 CALL CLEAR
730 PRINT :: PRINT "BE CAREFUL OF T
    HE BALLOON ON"
740 PRINT :: PRINT "THE TOP ROW WHI
    CH CHANGES"
750 PRINT :: PRINT "COLOR. WHEN THI
    S BALLOON IS" :: PRINT :: PRINT
     "BLACK, IT WILL BLOW UP IN"
760 PRINT :: PRINT "YOUR MAN'S FACE
    . IF IT IS"
770 PRINT :: PRINT "YELLOW, YOU WIL
    L RECEIVE 250"
780 PRINT :: PRINT "POINTS." :: PRI
    NT :: PRINT :: PRINT :: PRINT T
    AB(10);"GOOD LUCK!"
790 PRINT :: PRINT :: PRINT :: PRIN
    T TAB(4);"PRESS ANY KEY TO STAR
    T"
800 CALL KEY(0,K,S):: IF S=0 THEN 8
    00 ELSE RETURN
810 REM LIFE-1
820 LIFE=LIFE-1 :: DISPLAY AT(1,25)
    :STR$(LIFE):: IF LIFE=0 THEN 87
    0
830 IF G1=0 THEN DISPLAY AT(Y,X):"
    (3 SPACES)" :: CALL HCHAR(ROW,C
    OL,32):: ROW=6 :: COL=17 ::DX=0
    :: X=13 :: CALL HCHAR(ROW,COL,9
    6):: DISPLAY AT(Y,X):G$
840 IF G1=1 THEN RETURN
850 CALL KEY(0,K,S):: IF S=0 THEN 8
    50
860 RETURN
870 DISPLAY AT(8,7):"G A M E
    (3 SPACES)O V E R" :: DISPLAY A
    T(11,7):"PLAY AGAIN (Y/N)?"
880 ACCEPT AT(11,25)BEEP VALIDATE("
    YN"):H$ :: IF H$="Y" THEN 130 E
    LSE CALL DELSPRITE(ALL):: CALL
    CLEAR :: STOP
```

Computer War For Atari, VIC, And TI

Dan Gutman

The success of the movie *WarGames* has spawned several post-*Missile Command* "end of the world" games, most notably Probe 2000's *War Room* and Thorn E.M.I.'s *Computer War*. Coleco, which owns the rights to the title *WarGames*, has yet to be heard from. There will certainly be others.

Computer War is a game in three parts. In the first phase, you see a map of the United States with four American missile bases highlighted. Also highlighted is the computer at NORAD (North American Air Defense System). Suddenly, small white blips move into view, indicating that nuclear missiles are headed for American targets.

But wait! Upon closer examination (of the instructions, that is), you realize that the missiles aren't enemy missiles—somebody has tapped into NORAD to activate a nuclear war simulation program. Since the computer can't tell the difference between real missiles and fake ones, it's going to launch a volley of American missiles as soon as NORAD headquarters is in danger. You've got to knock out the missiles in the computer's memory banks and crack the code to shut down the bases.

Find The Missiles

This first section consists merely of zooming from the map of the United States to individual missiles. The joystick controls an onscreen cursor. When the cursor overlaps the missile blip and the fire button is pressed, that area of the map zooms into view. This exercise is fairly easy. In fact, I would prefer that the blips move a little faster to make this part of the game more challenging. As it is now, zooming in on the missiles is merely a formality. The graphics on the map screen, however, are the most impressive of the game.

You will zoom to a close-up view of the missiles' target area. Aside from the mountains in the distance, the landscape is totally barren. In fact, you may wonder just what it is you are defending. There are no people or buildings around. Why not just let the missiles harmlessly explode and avoid all the complications?

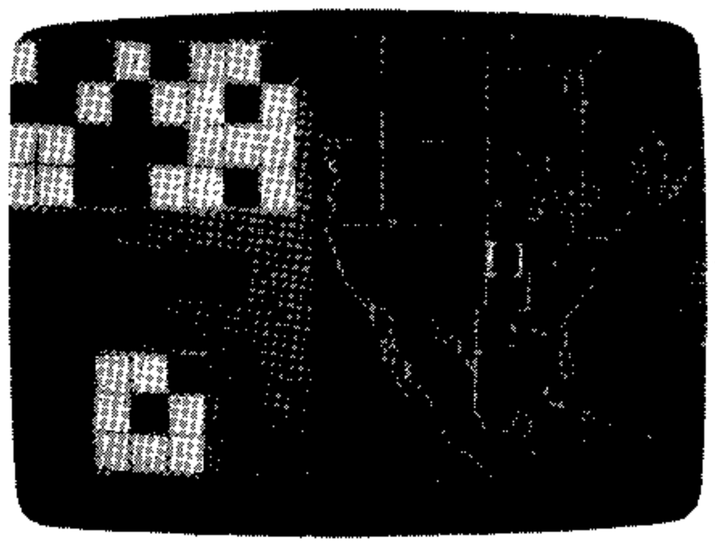
No, the security of the world is in your hands. The joystick can swivel your gunsight left and right and move it up and down also. A little box on one side of the screen indicates which direction to turn to see the missile, and rapid beeping tells you it is about to appear. The fire button launches your fire—two small rockets that arc across the sky and converge in the middle.

The missile will zip back and forth across the screen much faster than you can turn your gunsight. This means that to hit it, you have to fire before it appears onscreen. If you miss, you'll turn far past the target and have to wait for your slow-moving gunsight to change directions. The gunsight should probably move faster, or the missiles slower.

If you fail at this task, the missiles will reach U.S. bases and the DEFCON (Defense Condition) count will deteriorate. If it reaches DEFCON 1 (it starts at DEFCON 5), global war will begin.

Crack The Code

But if you succeed, you reach the third and most interesting part of the game. There are two banks of flashing squares on the left of the screen. Suddenly they freeze in random checkerboard patterns. With your joystick, you have to match the pattern of the smaller box with a section of the pattern of the larger box—kind of like fitting the peg in the correct hole. You have just a few seconds to do this, and you may have to rotate the box to complete the task.



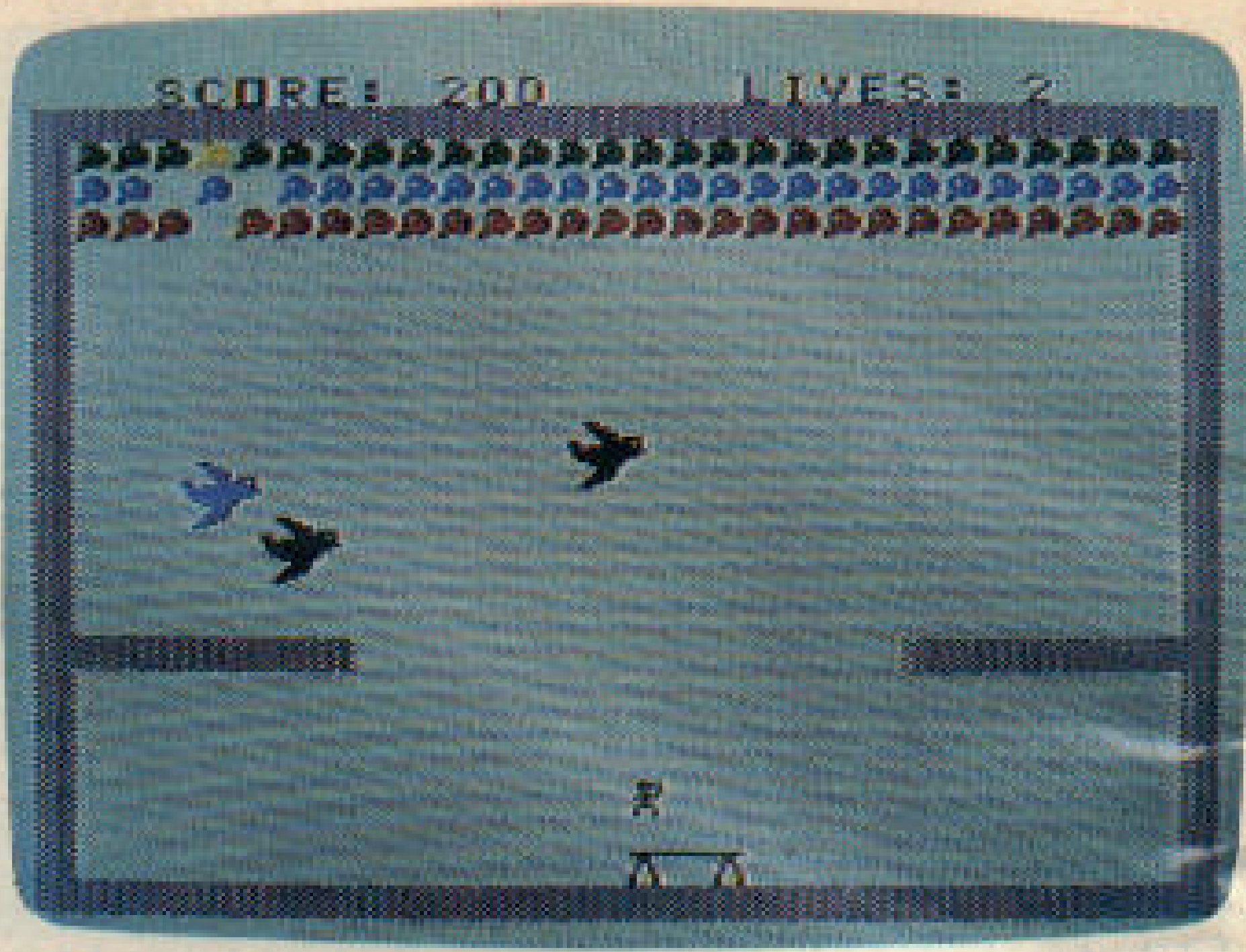
Incoming missiles are shown at right in opening screen of Computer War *(Atari version).*

If you match the two patterns (cracking the code), you have earned the right to shut down one of the U.S. missile bases. Of course, there are three more ready to launch everything they've got, so you've got to blow up more missiles, crack more codes, and so on.

Even though I love shooting games, I found that the last part of *Computer War*—cracking the code—was the most intriguing. Since *Space Invaders*, we have shot down so many enemies that the whole ordeal has become a little routine. But when you have five seconds to find a way to fit one pattern into another pattern—*that* can get the adrenaline flowing again. Mental challenges like this can stand up as games by themselves, and they should—they're exciting and they provide the brain with a little exercise.

Computer War is a single-player, single-difficulty level game. It captures the overall feel of *WarGames* without attempting the complexity of *War Room*, in which you must not only stop the missiles, but also rebuild cities, control production of goods and services, and pick up enemy spies.

Computer War's graphics are sometimes good (locating the



A flock of birds provides a distraction in the TI version of "Circus."

C. Regena

Foreign Languages

As you may already know, Texas Instruments has disbanded its home computer division. It's unfortunate that the TI-99/4A will no longer be manufactured and sold, because it's such a good computer.

However, there are still a lot of people out there who own TIs. I will keep writing and programming for the TI as long as there is a demand for it. Also, several third-party software companies have announced that they will continue to publish software as long as there is a market for it. If you are looking for further support, I suggest that you get involved with a local user group. If you are not aware of any in your area, or would like to start one, you may contact:

Charles LaFara
International 99/4A User Group
P.O. Box 67
Bethany, OK 73008

I have had several requests for programs for teaching or translating foreign languages. Letters have come from Southern California specifically requesting help in using Spanish accents and the tilde plus the opening exclamation and interrogation symbols.

In my high school days, we used headphones with an audio system that taught us a dialogue as we repeated phrases. With the TI computer and the Speech Synthesizer, you can imitate this. However, the computer adds branching capabilities to learning processes. The computer can determine when you are ready to continue to the next learning unit—or you can repeat one unit as long as you wish.

To use the Speech Synthesizer, you will need a command module that has speech capabilities. The Terminal Emulator II command module allows "unlimited" speech—there is no set vocabulary of words—so it is an ideal module for foreign speech. With the command module in place, press 1 for TI BASIC as usual (not 2 for the particular module). Any words that you want spoken you can spell phonetically in your program. Warning: Allow plenty of time to experiment with different sounds and spellings. The Spanish program included here presents the option to use speech.

To The Screen

Now to print the language on the screen. You probably noticed that the TI-99/4A keyboard has a tilde on one of the keys (FCTN W). It's the little curvy mark that belongs above the N in many Spanish words. The tilde is important enough in the language to change the pronunciation and the meaning of words. In Spanish writing you cannot just ignore the tilde or you may convey the wrong meaning. For example, Segundo P. I. Acuña writes, "A MONO is a MONKEY whereas a MOÑO is variously a bun, a crest, a chignon, a tuft.... You would wear a bun in your hair, but never a monkey!"

The problem with the tilde alone on a key is that it really should be above an N. To print an N with the tilde on the screen you need to print the tilde on one line, then the N directly below it. The solution is to design our own characters with the N and the tilde together and the accents with the appropriate vowels.

For this example program, I am leaving all the lowercase letters as is. (They really are small capital letters, not true lowercase letters, but it would take too much memory to redefine all of them to look like the normal lowercase letters.) To be able to print the N with the tilde and the vowels with the accent marks, I have redefined several characters.

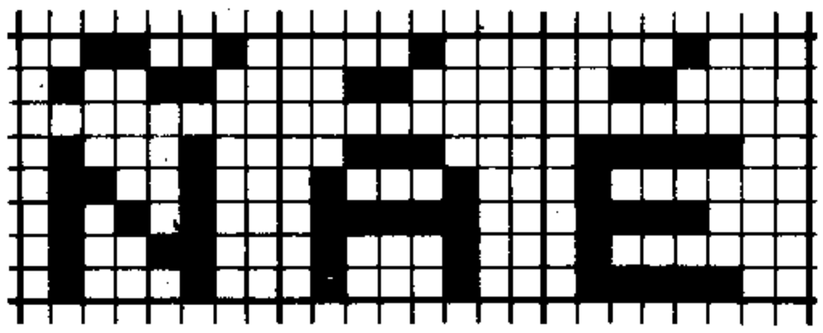
In your own programming, choose regular characters between 33 and 127 that you would not otherwise be printing in the program. I chose to use Characters 91, 92, 93, and 94 for a, e, i, and o. I redefined the underline, Character 95, to be the ñ. I also redefined Character 35 to be the upside-down exclamation point and Character 36 to be the upside-down question mark. Later when you PRINT "$" you won't see the dollar sign, but the upside-down question mark.

If you have the TI Extended BASIC command module, you can find out how the computer defines the characters by using the CHARPAT function. Rather than draw my own little letters, I used Extended BASIC and my printer to print a list of the definitions. If you have a printer and Extended BASIC, you can try the following

program. Change line 100 for your own printer configuration.

```
100 OPEN #2:"RS232.BA=600"
110 PRINT #2:"CHARACTER","PATTERN CO
    DE": :
120 FOR N=33 TO 126
130 CALL CHARPAT(N,C$)
140 PRINT #2: :N;"    ";CHR$(N),C$
150 NEXT N
160 CLOSE #2
170 END
```

This program showed me that for the lowercase n, the character definition is 0000004464544C44. The next step was to get out the graph paper and draw a tilde above the given n:



The new character definition is 324C004464544C44. Similarly, you can draw the accents above the lowercase a, e, i, and o.

The redefined Spanish characters all have the accents going the same way. In French you will need the regular e plus é and è. In German you will need to define vowels with the umlaut marks above. (Sure, use the same idea for Chinese or Japanese characters—but I'll leave that up to you!)

The Special Characters

After you have redefined the characters in your program, you can print them on the screen—just remember which symbols correspond to which regular characters. For example, in the Spanish program, to print the word "niño", remember that the ñ is the underline (FCTN U) and use the command PRINT "ni_o". As you are programming, you will see the underline, but when the program is RUN, the underline will be redefined and you will see the ñ.

The main purpose of this Spanish program is to illustrate how to print the special characters. All of the Spanish is written with the lowercase letters—release the ALPHA LOCK key to type the lowercase letters. To type any symbols on the face of the keys, use FCTN and the key. Any phrase in a PRINT #1 statement is spoken with the Speech Synthesizer. You may want to experiment and change these pronunciations. If you do not have the Terminal Emulator II module or the Speech Synthesizer, make choice number 2 at the beginning of the program for no speech. The variable SP will then equal 2 and all commands involving speech will be by-passed.

The first part of the program draws pictures and shows the Spanish word or phrase. If you have speech, the computer will say the word or phrase and you may repeat it. If you want to hear the phrase again, press the space bar. To continue after each presentation, press the ENTER key. The last section of the program presents Spanish phrases with the English translation.

If you prefer to avoid the typing, you can receive a copy of "Spanish" by sending $3, a blank cassette or diskette, and a stamped, self-addressed mailer to:

C. Regena
P.O. Box 1502
Cedar City, Utah 84720

Program Explanation

110–200 Print title screen; define special Spanish characters.
210–250 Print option for speech and receive a 1 or 2. If "no speech" is chosen, SP will equal 2.
260–290 Print instructions.
300–310 If speech option is chosen, OPEN the device to allow speech. You must have the TI Speech Synthesizer and Terminal Emulator II to use speech.
320–370 Define graphics characters for characters numbered 128 to 137. Be sure to type all the commas as shown. If you RUN the program and get an error message for line 330 or 340, there is probably a typing error in lines 360–370.
380–390 Define colors for graphics.
400–420 Wait for user to press any key to begin.
430–570 Draw a boy on the screen and present the Spanish phrase for "I am a boy."
580–690 Draw a girl and say the phrase for a girl.
700–1470 Clear screen; draw face. CALL CHAR statements define graphics, and CALL HCHAR and CALL VCHAR statements draw on the screen. W$ contains the Spanish word to be printed. PRINT #1 statements use the speech synthesizer to say the word.
1480–1890 Present Spanish phrases with English translations.
1900–1950 Subroutine used for each word or phrase. If the space bar is pressed, R = 2 and the phrase is repeated. If ENTER is pressed, the program continues.
1960–1990 Subroutine to print the word on the screen without scrolling. W$ is the Spanish word, and X is the row for printing.
2000–2030 Clear screen; close speech device; end program.

Learning Spanish

```
100 REM  SPANISH
110 CALL CLEAR
120 PRINT TAB(10);"SPANISH"
130 CALL CHAR(95,"324C004464544C44"
    )
140 PRINT :TAB(10);"espa_ol":::::
150 CALL CHAR(91,"0830003844 7C4444"
    )
160 CALL CHAR(92,"083000 7C4078407C"
    )
170 CALL CHAR(93,"0830003810101038"
    )
180 CALL CHAR(94,"083000 7C4444447C"
    )
190 CALL CHAR(35,"0010001010101010"
    )
200 CALL CHAR(36,"0003000810202210"
    )
```

```
210 PRINT "DO YOU HAVE THE SPEECH
    (6 SPACES)SYNTHESIZER AND":"TER
    MINAL EMULATOR II?"
220 PRINT :"  1 YES, INCLUDE SPEECH
    ":"  2 NO SPEECH"
230 CALL KEY(0,K,S)
240 IF (K<49)+(K>50)THEN 230
250 SP=K-48
260 CALL CLEAR
270 IF SP=2 THEN 290
280 PRINT "PRESS THE SPACE BAR TO H
    EAR THE WORD OR PHRASE AGAIN.":
    :
290 PRINT "PRESS <ENTER> TO CONTINU
    E":"AFTER EACH WORD OR PHRASE."
    :::
300 IF SP=2 THEN 320
310 OPEN #1:"SPEECH",OUTPUT
320 FOR C=128 TO 137
330 READ C$
340 CALL CHAR(C,C$)
350 NEXT C
360 DATA 3C7EC3C3C3C37E3C,000000FFF
    F,181818FFFF181818,1818181818
    3C3C,0000000000101030
370 DATA 6642C38181,0000000008080C04
    ,06040C183870E0C,6020301810E07
    03,FFFFFFFFFFFFFFFF
380 CALL COLOR(13,5,1)
390 CALL COLOR(14,5,1)
400 PRINT "PRESS ANY KEY TO BEGIN."
410 CALL KEY(0,K,S)
420 IF S<1 THEN 410
430 CALL CLEAR
440 CALL HCHAR(17,16,128)
450 CALL HCHAR(18,15,129,3)
460 CALL HCHAR(18,16,130)
470 CALL HCHAR(19,16,131)
480 CALL HCHAR(20,15,132)
490 CALL HCHAR(20,16,133)
500 CALL HCHAR(20,17,134)
510 CALL HCHAR(21,15,135)
520 CALL HCHAR(21,17,136)
530 PRINT TAB(9);"Soy un ni_o."
540 IF SP=2 THEN 560
550 PRINT #1:"SOY UN ^NEEN YO."
560 GOSUB 1900
570 IF R=2 THEN 540
580 CALL COLOR(13,7,1)
590 CALL COLOR(14,7,1)
600 CALL CHAR(133,"7E7EFFFFFFFFFFFF
    ")
610 CALL CHAR(135,"07070F1F3F7FFFFF
    ")
620 CALL HCHAR(20,16,137)
630 CALL CHAR(136,"E0E0F0F8FCFEFFFF
    ")
640 CALL HCHAR(23,11,32,12)
650 PRINT TAB(8);"Soy una ni_a."
660 IF SP=2 THEN 680
670 PRINT #1:"SOY OONA ^NEEN YUH."
680 GOSUB 1900
690 IF R=2 THEN 660
700 CALL CLEAR
710 CALL CHAR(144,"EFDFFAA7FFEDFE7F
    ")
720 CALL CHAR(145,"8040201008040201
    ")
730 CALL CHAR(146,"0102040810204080"
    )
740 CALL CHAR(147,"FF")
750 CALL HCHAR(8,19,144,7)
760 CALL HCHAR(9,18,144)
770 CALL HCHAR(9,26,144)
780 CALL VCHAR(10,17,144,8)
790 CALL VCHAR(10,27,144,8)
800 CALL HCHAR(18,18,145)
810 CALL HCHAR(18,26,146)
820 CALL HCHAR(19,19,145)
830 CALL HCHAR(19,25,146)
840 CALL HCHAR(20,20,147,5)
850 PRINT TAB(17);"cabeza"
860 IF SP=2 THEN 880
870 PRINT #1:"^CA BAY TSA."
880 GOSUB 1900
890 IF R=2 THEN 860
900 CALL CHAR(152,"0F1020408083878F
    ")
910 CALL CHAR(153,"F00804020 1C1E0F1
    ")
920 CALL CHAR(154,"4F4F4F2F2F2F708"
    )
930 CALL CHAR(155,"F2F2F2F4F4F40701
    ")
940 CALL COLOR(16,5,1)
950 FOR I=20 TO 23 STEP 3
960 CALL HCHAR(11,I,152)
970 CALL HCHAR(11,I+1,153)
980 CALL HCHAR(12,I,154)
990 CALL HCHAR(12,I+1,155)
1000 NEXT I
1010 W$="ojos"
1020 X=12
1030 GOSUB 1960
1040 IF SP=2 THEN 1060
1050 PRINT #1:"^O HOES."
1060 GOSUB 1900
1070 IF R=2 THEN 1040
1080 CALL CHAR(136,"0004040808101 02
    ")
1090 CALL CHAR(137,"2040408080 8C936
    ")
1100 CALL COLOR(14,10,1)
1110 CALL HCHAR(13,22,136)
1120 CALL HCHAR(14,22,137)
1130 W$="nariz"
1140 X=14
1150 GOSUB 1960
1160 IF SP=2 THEN 1200
1170 PRINT #1:" NAR ^DHIZ."
1180 GOSUB 1900
1190 IF R=2 THEN 1160
1200 CALL CHAR(128,"2040A0100806 01"
    )
1210 CALL CHAR(129,"0000000000000817
    E")
1220 CALL CHAR(130,"0402050810608")
1230 CALL HCHAR(16,21,128)
1240 CALL HCHAR(16,22,129)
1250 CALL HCHAR(16,23,130)
1260 W$="boca"
1270 X=16
1280 GOSUB 1960
1290 IF SP=2 THEN 1310
1300 PRINT #1:"^BO CA."
1310 GOSUB 1900
1320 IF R=2 THEN 1290
1330 CALL CHAR(148,"0000000C4222222
    2")
1340 CALL CHAR(149,"0000006010101 02
    ")
```

```
1350 CALL CHAR(150,"0000000608080804")
1360 CALL CHAR(151,"0000003042444444")
1370 CALL HCHAR(10,20,148)
1380 CALL HCHAR(10,21,149)
1390 CALL HCHAR(10,23,150)
1400 CALL HCHAR(10,24,151)
1410 W$="pestu_a"
1420 X=10
1430 GOSUB 1960
1440 IF SP=2 THEN 1460
1450 PRINT #1:"PES ^TUNE YA."
1460 GOSUB 1900
1470 IF R=2 THEN 1440
1480 CALL CLEAR
1490 PRINT "Buenos dJas, se_or.":::"Good day, Sir."
1500 IF SP=2 THEN 1520
1510 PRINT #1:"^BWAY NOSE THEE AS, _SEEN YOR."
1520 GOSUB 1900
1530 IF R=2 THEN 1500
1540 PRINT ::"Buenas tardes, se_ora.":::"Good afternoon, Madam."
1550 IF SP=2 THEN 1570
1560 PRINT #1:"^BWAY NAS. TAR DES, ^SEEN _YO RA."
1570 GOSUB 1900
1580 IF R=2 THEN 1550
1590 PRINT ::"Buenas noches, se_orita.":::"Good evening, Miss."
1600 IF SP=2 THEN 1620
1610 PRINT #1:"^BWAY NAS NO CHES, _SEEN YO _REE TA."
1620 GOSUB 1900
1630 IF R=2 THEN 1600
1640 PRINT :::"$Habla usted espa_ol?":::" Do you speak Spanish?"
1650 IF SP=2 THEN 1670
1660 PRINT #1:"^ABLA OO _STED _S PAN YOLE?"
1670 GOSUB 1900
1680 IF R=2 THEN 1650
1690 PRINT :::"#Yo hablo espa_ol!"::" I speak Spanish!"
1700 IF SP=2 THEN 1720
1710 PRINT #1:"^YO ABLO _S PAN YOLE!"
1720 GOSUB 1900
1730 IF R=2 THEN 1700
1740 PRINT ::::"$C^mo est[ usted?"::" How are you?"
1750 IF SP=2 THEN 1770
1760 PRINT #1:"^COE MOE  _S TAW  _USTED?"
1770 GOSUB 1900
1780 IF R=2 THEN 1750
1790 PRINT :::"No s\.":::"I do not know."
1800 IF SP=2 THEN 1820
1810 PRINT #1:"^NO SAY."
1820 GOSUB 1900
1830 IF R=2 THEN 1800
1840 PRINT ::::"#Adi^s!":::"Good bye!":::
1850 IF SP=2 THEN 1870
1860 PRINT #1:"^AWDHEE OSE!"
1870 GOSUB 1900
1880 IF R=2 THEN 1850
1890 GOTO 2000
1900 R=1
1910 CALL KEY(0,K,S)
1920 IF K=13 THEN 1950
1930 IF K<>32 THEN 1910
1940 R=2
1950 RETURN
1960 FOR I=1 TO LEN(W$)
1970 CALL HCHAR(X,4+I,ASC(SEG$(W$,I,1)))
1980 NEXT I
1990 RETURN
2000 CALL CLEAR
2010 IF SP=2 THEN 2030
2020 CLOSE #1
2030 END
```

CAPUTE!

Modifications Or Corrections To Previous Articles

Chopperoids

Here's how to produce a working version of the "Chopperoids" game described in the December 1983 issue (p. 122):

1. Load the MLX program shown on p. 216 of the December issue.
2. Add or change the following lines in the MLX program. (Note: These changes are for fixing Chopperoids only; they do *not* alter the MLX program. Be sure not to make any permanent changes to MLX.)

For tape users:

```
940 CLOSE #2:TRAP 32767:? "Finished.":?
950 IF NOT READ THEN END
955 BUFFER$(FIN-BEG+24)=CHR$(0):BUFFER$(
    25)=BUFFER$(55):LET READ=0:GOTO 360
```

For disk users:

```
1185 BUFFER$(31)=BUFFER$(61)
```

3. RUN MLX using the starting, ending, and run/init addresses specified in the Chopperoids article. Disk users should again choose to make a boot disk. Use the MLX Load function (CTRL-L) to load the Chopperoids data from the December issue. If you use the Display command (CTRL-D), you will see that all the data has been moved up five lines. That is, the data in lines 3584-3608 has been eliminated, so the data formerly at line 3614 is now at line 3584, and so forth.
4. Use the MLX New Address command (CTRL-N) to begin typing at line 6092. Add the following lines:

```
6092:197,020,208,252,169,000,026
6098:133,148,076,146,023,160,128
6104:005,166,142,169,000,157,087
6110:130,025,232,136,208,249,178
6116:141,005,208,141,006,208,169
6122:096,000,000,000,000,000,074
```

5. After you type the last line, MLX should create a boot tape or disk which is a working version of Chopperoids.

Atari Gas Mileage

In the Atari version of this utility from the December 1983 issue (p. 86), delete lines 280, 290, and 450 and change the following lines:

```
270 ? "{CLEAR}"
440 A=130-MG*2
```

TI Get The Gold

To load Program 2 of this two-part game from the December issue (p. 132), type in NEW, then OLD CS1. Reader Mark Leair suggests these improvements, which allow Program 1 to load Program 2:

For console BASIC:

```
790 PRINT "loading":"After load type RUN
    then Enter"
800 OPEN #1:"CS1",INTERNAL,OUTPUT,FIXED
810 CLOSE #1
```

For Extended BASIC:

```
790 PRINT "Loading"
800 RUN "CS1"
810 REM
```

Goodbye Charlie For 64 And VIC

In both these versions of this game from the November 1983 issue (p. 68), change the S=CS+10 in line 515 to SC=SC+10.

64 Crazy Climber

The logical AND in line 1440 of this game from the November 1983 issue (p. 80) should be replaced with an OR.

64 Sound Tester

The final Release stage of the ADSR envelopes generated in this program (November 1983, p. 187) is not realized because the program ends each note by POKEing the frequency to zero rather than by turning off the gate bit. To correct this, change the POKE W, in lines 250–280 to X= and change the following lines:

```
310 FORI=1TO15STEP2: POKEW,X: POKEHF,SO((
    I,A(2)): POKELF,SO(I+1,A(2))
311 O=O+1: FORN=1TOD(0): NEXT: POKEW,X-1:
    NEXT: FORI=1TO10000: NEXT
```

Then eliminate the NEXT in line 315. The new FOR-NEXT loop in 311 allows time for the Release to be heard at the end of the tune. Our thanks to Arthur Hunkins for this correction.

Stock Market Analyzer For VIC/64 And Atari

In both versions of this utility from the November 1983 issue (p. 54), the following line must be changed to plot stock prices higher than $10 per share:

```
620 FORT=1TOINT(15/HI*TP(X)):PRINT"{UP}";
    :NEXTT
```

In line 652 of the VIC/64 version, insert a colon after the first semicolon.

Timex/Sinclair Making Change

A typographical error in the machine language loader (Program 1) for this article from the September 1983 issue (p. 252) causes the program to crash. The twentieth character in A$ in line 20 should be 8 rather than 6.