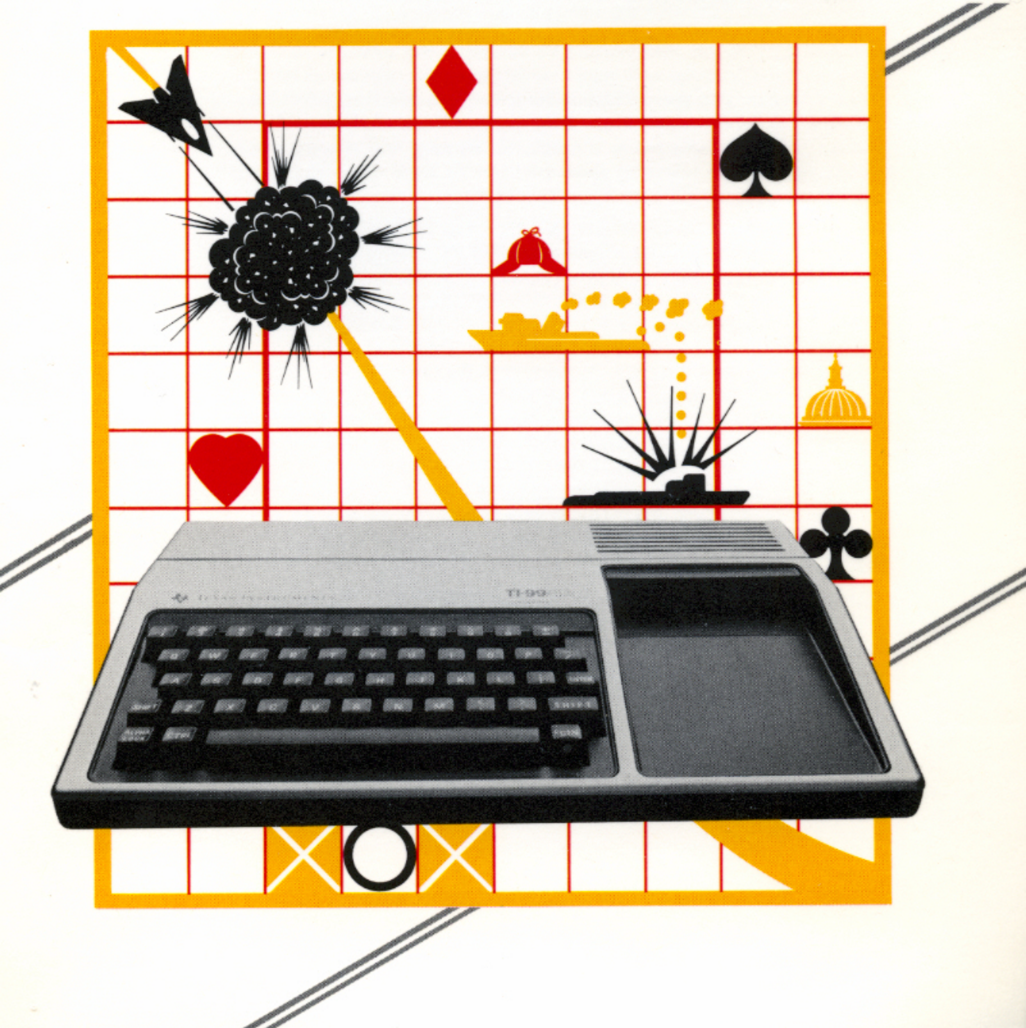
SAMS 22398

TI-99/4A™ Games

Includes 13 Fun and Educational Programs

Allen L. Wyatt



TI-99/4A™ Games

by

Allen L. Wyatt

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

International Standard Book Number: 0-672-22398-8
Library of Congress Catalog Card Number: 84-51462

Edited by *Susan Pink*

Printed in the United States of America.

TI-99/4A™ Games

Allen Wyatt has been actively involved with the microcomputer industry for more than six years and is currently software development supervisor for Sams Software in Indianapolis, Indiana. Mr. Wyatt has had extensive experience in computer consulting and software development.

He has written several commercial software packages utilizing many of the same techniques detailed in *TI-99/4A Games.* The broad range of computer programs runs the gamut from small system data bases to games and utilities.

In addition to being a computer author, Allen is a devoted family man and active church member. He uses his personal computers to assist him in all of these areas. At home, his family spends many hours using the computer.

Preface

This book is an expression of the simple things in life. We often face challenges, particularly in technological areas, that are so complex they "boggle the mind." *TI-99/4A Games* goes beyond that or short of that, depending on your perspective. It is short, simple, and sweet. This book shows you how several complete, ready-to-run, and entertaining games were conceived, designed, and implemented.

In addition to game programs, there are two miscellaneous programs. *Sprite/Character Generator* can be used to develop TI-99/4A sprites and characters for your own graphics programs. *Master Catalog* can be used to bring order to your collection of programs.

TI-99/4A Games contains intermediate-level programs written in TI Extended BASIC. There are complete listings and explanations for every program. You must have a TI Extended BASIC cartridge for the programs to run.

Now, your question might be, "What can I do with these games?" Well, you could play them, but that is not their greatest value. Because the concepts used in these games are documented and explained, they can be used in your programming. In this way, you will not only have fun, you will learn.

Take time to study the program listings. Then, if you feel adventurous, "poke around" and change lines to see "what happens." You may be surprised at what you can do.

One of the easiest ways to test your new programming skills is to change a program so it uses different input. For example, all of the programs use the keyboard for input. You can, however, change the input routines to use a joystick. By changing appropriate sections, you can learn more about programming.

Each chapter is comprised of several parts. The first part is a general statement about the program. The second is the rules or instructions for using the game or program. The third is a set of programming notes that will explain the operation of different parts of the program. Finally, each chapter contains the program listing for the game or program detailed.

If you purchased this book as part of a Combo Pack, you have the programs on a cassette tape. If you bought the book separately, you have to type in the listings. The first program on the tape is the *Main Menu.* It serves as a reminder of the available programs. Also, if your computer has a disk drive, you may want to "link" the programs so they are selectable from the menu and so they exit to the menu upon completion. However, the programming to do this is beyond the scope of this book.

As you play the games, study the programs, and use, explore and learn from this book, I hope you will have more than a little fun. After all, that is part of what computers and games are all about.

ALLEN WYATT

This book is dedicated to those who strive for a common cause— **quality.** *"By the fruits of their labors they shall be known."*

A NOTE TO THE READER

The programs in this book were not written as applications software but as educational examples of what your personal computer can do. All of the programs have been tested and work on the machine configuration for which they were designed. The programs are unprotected. This means that you can modify them to better understand how they work or to fit a different machine configuration.

What Is a Combo Pack?

A Combo Pack, like this package, is a step beyond your average technical book. While most books give you programming examples through printed listings (which we do here), Combo Packs provide the book and the listings recorded on magnetic media, either disk, cassette tape, or both.

Every effort has been made to be clear, concise, and informative about how these programs and routines work. If you experience any difficulty with the software operations, the solution can be found in the book or in your computer manuals.

We are rather proud of the time and effort that went into preparing the Combo Pack. If you have purchased the Combo Pack and have enjoyed using it, let us know your thoughts. Your comments will be valuable in preparing future Combo Packs.

LOADING INSTRUCTIONS

The cassette accompanying this Combo Pack contains the program listings printed in the book.

To load a cassette file from this tape, perform the following steps:

1. Put the cassette tape into the cassette recorder.
2. Position the tape at the beginning of the program you want to load.
3. Type **OLD CSI**
   Press <ENTER>
4. Follow the directions as they appear on your video screen.
5. When the cursor reappears on the screen, type **RUN** and press <ENTER> to run the program.

6. If you rewound the tape completely, the first program on the tape *(Main Menu)* is loaded into memory and will appear on the screen. For more information on this program, refer to Section 1.
7. To load another program, simply repeat steps 2 through 5.

For detailed instructions on any program on the tape, refer to the appropriate section in the book.

The following list shows the program names and tape counter positions for the contents of the cassette tape. These numbers are approximate and may vary from recorder to recorder. They should, however, assist you in locating the programs you are searching for. Each side of the tape contain the same programs in the same order.

Tape Directory

Listing #	Program Name	Counter Location
1	Main Menu	2
2	Flip Flop	14
3	Mastermind	26
4	Towers	39
5	Sherlock's Home	52
6	Attack of the Zargons	90
7	Phaser Practice	106
8	Acey-Ducey	126
9	Big Government	143
10	Tic-Tac-Toe	183
11	Qubic	200
12	Depth Charge	222
13	Sprite/Character Generator	239
14	Master Catalog	256

Contents

Section 1
MAIN MENU PROGRAM

Section 2
GAME PROGRAMS

Section 3
MISCELLANEOUS PROGRAMS

Section 1

MAIN MENU PROGRAM

1. Main Menu

Most programs, if they contain more than one function, are menu driven; the available options can be selected from a list of choices that are clearly and conveniently displayed on the screen. *Master Catalog* (Program 14) uses menus extensively.

Main Menu is actually two programs in one. The first is the demonstration screen. It utilizes letters as sprites to spell **TI-99/4A Games** (and get your attention).

The second part of the program is the menu for *TI-99/4A Games.* It presents 13 choices on two screens. The games are listed on the first screen and the remaining utility programs are on the second.

Program Instructions

1. Load and run the Main Menu program. The demonstration screen will appear.
2. Type **Q** and press <ENTER> to exit the demonstration screen. In a few moments, the menu will be displayed.
3. Type **M** and press <ENTER> to see the other screen of menu choices.
4. Type **E** and press <ENTER> to exit the menu.
5. When you exit the menu the computer returns to the TI Extended BASIC command level.

Programming Notes

In the demonstration, each letter in the title moves from a randomly chosen start position to a set ending position. At the same time, a randomly chosen musical note is heard. After the title is positioned, the color and size of each letter is randomly changed. Then the entire process is repeated. You may want to study the listing to learn about sprite manipulation techniques.

This portion of the program, with minor alterations, can be used as a "sign board" for many different types of messages. Note the DATA lines in the Programming Notes section. Change these lines for a different message.

10-50	Program initialization
60-80	Screen setup
90	Read first character to be displayed
100-130	Determine start and end locations for character
140	Call display routine
150	Loop for another character
160-170	Call special effects routine
180-200	Check for keypress and quit if **Q** is pressed
210-280	Randomly change sprites color and size
290-440	Move character step-by-step
450-640	Data for characters to be printed in demo
650-710	Program initialization
720-740	Print first page of menu choices
750	Wait for your input
760-770	Print second page of menu choices
780	Wait for your input
790	Loop to first-page display
800-890	Prompt and receive your input
900-1050	Program name data

Program Listing

```
10 REM      MENU
20 REM
30 REM  OPENING DEMO
40 REM
50 RANDOMIZE
60 CALL CLEAR :: CALL DELSPRITE(ALL)
70 DISPLAY AT(23,1):"PRESS 'Q' TO QUIT"
80 RESTORE 510 :: MF=2 :: CALL MAGNIFY(MF):: SN=0
90 READ A$ :: IF A$="END" THEN 160
100 CV=ASC(A$):: SN=SN+1
110 READ X2 :: READ Y2
120 X1=INT(RND*190+1):: Y1=INT(RND*250+1)
130 IF X1=X2 OR Y1=Y2 THEN 120
140 GOSUB 340
150 GOTO 90
160 GOSUB 210
170 GOTO 80
180 CALL KEY(0,A,S):: IF S=0 THEN 200
190 IF A=81 THEN 680
200 RETURN
210 FOR J=1 TO 500
```

cont. on next page

Program Listing—cont.

```
220 Q=INT(RND*(SN+1)):: IF Q<1 OR Q>SN THEN 220
230 SC=INT(RND*17):: IF SC<1 OR SC>16 THEN 230
240 CALL COLOR(#Q,SC)
250 IF INT(J/10)*10=J THEN GOSUB 180
260 IF RND>.5 THEN MF=3-MF :: CALL MAGNIFY(MF)
270 NEXT J
280 RETURN
290 REM
300 REM  MOVE LETTER FROM
310 REM  BEGINNING LOCATION
320 REM  TO ENDING LOCATION
330 REM
340 SL=(X2-X1)/(Y2-Y1):: C=5*SGN(Y2-Y1)
350 SC=INT(RND*17):: IF SC<2 OR SC>16 OR SC=8 THEN 350
360 CALL SPRITE(#SN,CV,SC,X1,Y1)
370 FOR Y3=Y1 TO Y2 STEP C
380 IF INT((Y3-Y1)/10)*10=(Y3-Y1)THEN GOSUB 180
390 X3=X1+(Y3-Y1)*SL
400 CALL LOCATE(#SN,X3,Y3)
410 CALL SOUND(RND*100+5,RND*5000+110,3)
420 NEXT Y3
430 CALL SPRITE(#SN,CV,SC,X2,Y2)
440 RETURN
450 REM
460 REM  DATA FOR LETTERS
470 REM  TO USE AS SPRITES
480 REM  AND THEIR ENDING
490 REM  LOCATIONS
500 REM
510 DATA T,1,10
520 DATA I,13,25
530 DATA -,25,40
540 DATA 9,37,55
550 DATA 9,49,70
560 DATA /,61,85
570 DATA 4,72,100
580 DATA A,84,115
590 DATA G,109,145
600 DATA A,121,160
610 DATA M,133,175
620 DATA E,145,190
630 DATA S,157,205
640 DATA END
650 REM
660 REM  MENU PROGRAM
670 REM
680 CALL DELSPRITE(ALL):: CALL CLEAR
690 NP=13 :: G=11 :: M=13
700 DIM P$(13)
710 RESTORE 930 :: FOR X=1 TO NP :: READ P$(X):: NEXT X
720 DISPLAY AT(1,6):"TI-99/4A GAMES"
730 DISPLAY AT(3,1):"GAME PROGRAMS:"
740 FOR X=1 TO G :: DISPLAY AT(X+5,2):USING "##) ######
    ###############":X,P$(X):: NEXT X
```

```
750 GOSUB 850
760 CALL CLEAR :: DISPLAY AT(1,6):"TI-99/4A GAMES" :: D
    ISPLAY AT(3,1):"MISCELLANEOUS PROGRAMS:"
770 FOR X=G+1 TO M :: DISPLAY AT(X+5-G,2):USING "##) ##
    ###################":X,P$(X):: NEXT X
780 GOSUB 850
790 CALL CLEAR :: GOTO 720
800 REM
810 REM DISPLAY COMMON
820 REM PROMPTS AND GET
830 REM USER RESPONSE
840 REM
850 DISPLAY AT(20,1):"THIS IS A LIST OF PROGRAMS","ON T
    HIS TAPE."
860 DISPLAY AT(23,1):"<M> FOR MORE, <Q> TO QUIT:"
870 ACCEPT AT(23,27)VALIDATE("MQ"):CH$ :: IF CH$<>"M" A
    ND CH$<>"Q" THEN 870
880 IF CH$="Q" THEN CALL CLEAR :: PRINT "HAVE A NICE DA
    Y!" :: END
890 RETURN
900 REM
910 REM PROGRAM NAMES
920 REM
930 DATA FLIP FLOP
940 DATA MASTERMIND
950 DATA TOWERS
960 DATA SHERLOCK'S HOME
970 DATA ATTACK OF THE ZARGONS
980 DATA PHASER PRACTICE
990 DATA ACEY-DUCEY
1000 DATA BIG GOVERNMENT
1010 DATA TIC-TAC-TOE
1020 DATA QUBIC
1030 DATA DEPTH CHARGE
1040 DATA SPRITE GENERATOR
1050 DATA MASTER CATALOG
```

Section 2

GAME PROGRAMS

2. Flip Flop

This is an easy-to-play game of chance. You guess what the computer's next move will be. As you will soon discover, this can be very aggravating.

The object of the game is to change a row of 10 X's to 10 O's in as few moves as possible. You do this by choosing the position of the character you want to change. Sounds easy, right? Well, there's a catch—the computer may also randomly change a character. The computer could change the position you selected in the previous move, thus negating your choice. Take heart, however, because the odds are in your favor.

You may find *Flip Flop* not only aggravating, it may also be addicting. Your chance of winning depends largely on whether or not the Powers-That-Be recognize the sincerity of your efforts. Look sincere and don't berate the computer if it takes you longer to win than you first thought. After all, your computer is only a tool for the Powers-That-Be!

Game Rules

1. Load and run the Flip Flop program.
2. The directions will be displayed.
3. Press a 0 through 9 to change the corresponding X to an O, or O to X.
4. Try to change all 10 X's to O's.
5. Pressing **M** at any time ends the game.
6. When the game is finished, you can play again or exit.

Programming Notes

10-20	Program initialization
30-190	Instructions
200-210	Game initialization
220-240	Get string and print it

250-430	Get your input and randomly change X's to O's or O's to X's
440	Increment turn counter
450-460	Check for winner and loop if none
470	Delay and print final string value
480-490	End of game message
500-540	Play again or exit
550	Concatenate game array to a string
560-580	Print a string while playing random sounds

Program Listing

```
10 REM FLIP FLOP
20 CALL CLEAR
30 Z$="THE OBJECT OF THE ANCIENT" :: Y=2 :: GOSUB 560
40 Z$="GAME OF FLIP FLOP" :: Y=3 :: GOSUB 560
50 Z$="IS TO CHANGE THIS:" :: Y=4 :: GOSUB 560
60 Z$="X X X X X X X X X X" :: Y=6 :: GOSUB 560
70 Z$="TO THIS:" :: Y=8 :: GOSUB 560
80 Z$="O O O O O O O O O O" :: Y=10 :: GOSUB 560
90 Z$="IN AS FEW MOVES AS POSSIBLE." :: Y=12 :: X=1 ::
   GOSUB 570
100 Z$="TO CHANGE A POSITION, PRESS" :: Y=14 :: X=1 ::
   GOSUB 570
110 Z$="THAT POSITION'S NUMBER.  IT" :: Y=15 :: X=1 ::
   GOSUB 570
120 Z$="WILL THEN BE CHANGED.  HOW-" :: Y=16 :: X=1 ::
   GOSUB 570
130 Z$="EVER, THE GODS MAY NOT LOOK" :: Y=17 :: X=1 ::
   GOSUB 570
140 Z$="KINDLY UPON YOUR EFFORTS," :: Y=18 :: X=1 :: GO
   SUB 570
150 Z$="AND THE POSITIONS MAY" :: Y=19 :: X=1 :: GOSUB
   570
160 Z$="CHANGE ON THEIR OWN." :: Y=20 :: X=1 :: GOSUB 5
   70
170 Z$="P.S.: TO END THE GAME AT ANY" :: Y=22 :: X=1 ::
    GOSUB 570
180 Z$="TIME, PRESS <Q>." :: Y=23 :: X=1 :: GOSUB 570
190 FOR X=1 TO 1500 :: NEXT X :: CALL CLEAR
200 FOR X=1 TO 10 :: A$(X)="X" :: NEXT X
210 DISPLAY AT(8,4):"1 2 3 4 5 6 7 8 9 0"
220 GOSUB 550
230 DISPLAY AT(9,1)SIZE(27):" " :: DISPLAY AT(10,1)SIZE
   (27):" " :: Y=10 :: X=4 :: GOSUB 570
240 DISPLAY AT(20,1):"TO CHANGE:"
250 CALL KEY(0,CH,S):: IF S<>1 THEN 250
260 IF CH=81 THEN CALL CLEAR :: GOTO 500
270 IF CH<48 OR CH>57 THEN 250
280 DISPLAY AT(20,12)SIZE(5):CH-48
```

cont. on next page

Program Listing—cont.

```
290 N=CH-48 :: IF N=0 THEN N=10
300 M=N
310 IF A$(N)="O" THEN 350
320 A$(N)="O"
330 N=INT(RND*11):: IF N<1 OR N>10 THEN 330
340 IF A$(N)="X" THEN A$(N)="O" :: GOTO 440
350 A$(N)="X"
360 IF M=N THEN 330
370 GOTO 440
380 IF A$(N)="O" THEN 420
390 A$(N)="O"
400 N=INT(RND*11):: IF N<1 OR N>10 THEN 400
410 IF A$(N)="X" THEN A$(N)="O" :: GOTO 440
420 A$(N)="X"
430 IF M=N THEN 400
440 C=C+1
450 FOR X=1 TO 10 :: IF A$(X)<>"O" THEN 220
460 NEXT X
470 GOSUB 550 :: DISPLAY AT(9,1)SIZE(27):" " :: DISPLAY
     AT(10,1)SIZE(27):" " :: Y=10 :: X=4 :: GOSUB 570
480 IF C<13 THEN DISPLAY AT(19,1):"VERY GOOD.  YOU GUES
    SED IT","IN ONLY "&STR$(C)&" TURNS!" :: GOTO 500
490 DISPLAY AT(19,1):"GOOD TRY...BUT YOU COULD DO","BET
    TER.  IT TOOK YOU "&STR$(C),"TURNS THIS TIME."
500 DISPLAY AT(23,1):"TRY AGAIN (Y/N):"
510 ACCEPT AT(23,18)VALIDATE("YN"):CH$
520 IF CH$="Y" THEN CALL CLEAR :: C=0 :: GOTO 200
530 IF CH$<>"N" THEN 510
540 CALL CLEAR :: DISPLAY AT(8,1):"THANKS FOR THE GAME!
    " :: END
550 Z$="" :: FOR X=1 TO 10 :: Z$=Z$&A$(X)&" " :: NEXT X
     :: RETURN
560 X=INT(14-LEN(Z$)/2)
570 FOR Z=1 TO LEN(Z$):: DISPLAY AT(Y,X+Z-1):SEG$(Z$,Z,
    1):: IF SEG$(Z$,Z,1)<>" " THEN CALL SOUND(1,RND*400
    0+200,4)
580 NEXT Z :: RETURN
```

3. Mastermind

Most people are familiar with the game of *Mastermind,* right? Well, this version is an adaptation of that popular game. The object is to guess the correct combination of four elements of a set. Some versions of *Mastermind* use only numbers, but this version also uses color.

After the initial instructions are displayed, little text is used. If you have a monochrome monitor, the number value that corresponds with the color that is being displayed is shown below the color block. When you finish guessing the color and position of a row of squares, the results are displayed to the right of the input line (see game rule #6). Because this game uses numbers and colors instead of text, it is well-suited for small children who can count and know color names, but have not begun to read. However, it is also a refreshing change of pace for adults and older children.

Game Rules

1. Load and run the Mastermind program.
2. The directions will be displayed.
3. The position that currently requires you to choose a color is indicated with a blinking rectangle.
4. Press a number (1-9) to select the desired color for each position.
5. Press N to move to the next position. The next position will begin blinking or, if you have completed a row, the results will be displayed.
6. Two numbers will be displayed to the right of your four guesses. The first number represents the number of colors that were correct, but in the wrong positions. The second number indicates the number of colors that were correct and also in the right positions.
7. When you have guessed the colors for the first row, try again on the next row.

8. Press **Q** at any time to end the game.
9. You win when you have matched the position and color of every rectangle in the row.
10. After you have won, you can play again or exit the program.

Programming Notes

This game illustrates the use of multiple sprites on the screen. Each position within a set is displayed using one magnified sprite. Consequently, up to 16 sprites can be on the screen at one time.

10-80	Instructions
90-110	Prompt and wait for <ENTER> to be pressed
120-160	Additional instructions
170-190	Prompt and wait for <ENTER> to be pressed
200-240	Game initialization
250	Display input line and get your input
260-320	Check for correct input
330	Display results
340	Check for winner and loop if no winner yet
350-400	Announce the winner
410	End the game
420-480	Display input line
490-590	Get input from the keyboard; **Q** to end the game
600-640	Randomly select the game elements
650	Blank out graphics line (the row of squares)

Program Listing

```
10 REM     MASTERMIND
20 DISPLAY AT(1,9)ERASE ALL:"MASTERMIND"
30 RANDOMIZE
40 DISPLAY AT(3,1):"THE OBJECT OF MASTERMIND IS","TO GU
  ESS THE CORRECT COLOR","COMBINATION OF FOUR SQUARES
  ."
50 DISPLAY AT(6,1):"THE COMPUTER WILL ALLOW YOU","TO CH
  ANGE THE COLOR OF EACH","POSITION BY USE OF THE"
60 DISPLAY AT(9,1):"NUMBER KEYS (1-9).  USING","THE <N>
   KEY WILL ALLOW YOU","TO MOVE FROM ONE LOCATION"
70 DISPLAY AT(12,1):"TO ANOTHER.  WHEN ALL FOUR","SQUAR
  ES ARE COMPLETED, YOU","WILL SEE HOW MANY SQUARES"
80 DISPLAY AT(15,1):"WERE THE RIGHT COLOR, AND","IF THE
  Y WERE IN THE PROPER","POSITION."
90 DISPLAY AT(22,1):"PRESS <ENTER> TO CONTINUE:"
100 ACCEPT AT(22,28):CH$
```

```
110 IF CH$<>"" THEN 100
120 CALL CLEAR
130 DISPLAY AT(1,1):"TWO NUMBERS WILL BE SHOWN TO","THE
     RIGHT OF YOUR GUESSES,","THE FIRST IS THE NUMBER O
    F"
140 DISPLAY AT(4,1):"COLORS THAT WERE CORRECT,","BUT IN
     THE WRONG POSITION.","THE SECOND IS THE NUMBER OF"
150 DISPLAY AT(7,1):"COLORS THAT WERE CORRECT AND","IN
    THE RIGHT POSITION."
160 DISPLAY AT(10,1):"PRESSING <Q> AT ANY TIME","ALLOWS
     YOU TO QUIT THE GAME."
170 DISPLAY AT(22,1):"PRESS <ENTER> TO CONTINUE:"
180 ACCEPT AT(22,28):CH$
190 IF CH$<>"" THEN 180
200 CALL CLEAR :: CALL SCREEN(2):: CALL COLOR(1,16,1)::
     CALL COLOR(3,16,1):: CALL COLOR(4,16,1)
210 GOSUB 600 :: CALL CLEAR
220 Q=-4 :: T=0
230 Q=Q+5 :: IF Q>21 THEN Q=1
240 T=T+1
250 GOSUB 650 :: GOSUB 420 :: GOSUB 490
260 RW=0 :: RR=0
270 FOR X=1 TO 4 :: F(X)=0
280 IF G(X)=M(X)THEN RR=RR+1 :: F(X)=1
290 NEXT X
300 FOR X=1 TO 4 :: FOR Y=1 TO 4
310 IF G(Y)=M(X)AND Y<>X AND F(X)=0 THEN RW=RW+1 :: F(X
    )=1
320 NEXT Y :: NEXT X
330 DISPLAY AT(Q+1,17):RW :: DISPLAY AT(Q+1,20):RR
340 IF RR<4 THEN 230
350 CALL DELSPRITE(ALL):: CALL CLEAR
360 CALL SCREEN(8):: CALL CHARSET
370 PRINT "CONGRATULATIONS, YOU DID IT" :: PRINT "IN";T
    ;"TURNS!" :: PRINT :: PRINT :: PRINT
380 DISPLAY AT(23,1)SIZE(27):"PLAY AGAIN (Y/N):"
390 ACCEPT AT(23,19)SIZE(1)VALIDATE("YN"):CH$ :: IF CH$
    <>"Y" AND CH$<>"N" THEN 390
400 IF CH$="Y" THEN RUN
410 CALL CLEAR :: END
420 CALL CHAR(96,"FFFFFFFFFFFFFFFF")
430 CALL MAGNIFY(2)
440 FOR B=1 TO 4 :: B1=((B-1)*24+16):: B2=INT(Q/5)*4+B
450 CALL SPRITE(#B2,96,15,Q*8,B1)
460 NEXT B
470 DISPLAY AT(Q+1,17):"      "
480 RETURN
490 FOR X=1 TO 4 :: G(X)=0 :: C=15 :: B2=INT(Q/5)*4+X
500 CALL KEY(0,A,S):: IF S<>1 THEN A=0
510 IF A=81 THEN CALL DELSPRITE(ALL):: CALL CLEAR :: CA
    LL CHARSET :: CALL SCREEN(8):: GOTO 380
520 CALL COLOR(#B2,2)
530 FOR Z=1 TO 10 :: NEXT Z
```

cont. on next page

Program Listing—cont.

```
540 IF A>48 AND A<58 THEN C=A-46 :: DISPLAY AT(Q+3,(X-1
    )*3+1):C-2
550 CALL COLOR(#B2,C)
560 FOR Z=1 TO 10 :: NEXT Z
570 IF A<>78 OR C=15 THEN 500
580 G(X)=C
590 NEXT X :: RETURN
600 FOR X=1 TO 4 :: M(X)=0
610 Z=INT(RND*RND*165):: IF Z<3 OR Z>11 THEN 610
620 FOR Y=1 TO X :: IF Z=M(Y)THEN 610
630 NEXT Y
640 M(X)=Z :: NEXT X :: RETURN
650 DISPLAY AT(Q,1)SIZE(27):" " :: DISPLAY AT(Q+1,1)SIZ
    E(27):" " :: RETURN
```

4. Towers

You've heard of the famous Leaning Tower of Pisa? That's good, even though it has nothing to do with this game.

The proper name for this game is *The Towers of Hanoi.* The game begins with a screen showing several disks of different sizes, located on the leftmost of three towers. The object of the game is to move all the disks to tower three, the rightmost tower, in as few moves as possible. However, you can only move one disk at a time and you cannot place a larger disk on a smaller one. Try playing with two disks and then progress to three.

If you take too many moves, you lose the game. Besides that, you may not get invited back to Hanoi.

Game Rules

1. Load and run the Towers program.
2. The directions will be displayed.
3. Choose the number of disks to use in the game. You may select one through six disks.
4. Press <ENTER> to begin.
5. Select a disk to move. Disks are represented with 3, 5, 7, 9, 11, or 13 * (asterisks).
6. You can only move the top disk on a tower.
7. Select which tower to move the disk to. Towers are specified by 1, 2, and 3, 1 is the leftmost tower.
8. You cannot place a large disk on a smaller one.
9. Try to get all the disks to tower 3, the rightmost one, in as few moves as possible.
10. When you have won or taken the allotted number of moves, you can play again or exit the program.

Programming Notes

This game demonstrates screen control techniques. The towers and disks appear to remain static, while the comments and prompts scroll on the bottom portion of the screen. This is done using the unique text display capabilities of the TI-99/4A, and can be examined in the display routine in lines 540 through 580.

10-40	Initialization
50-70	Instructions
80-110	Determine number of disks to use by prompting player
120-160	Additional instructions
170-180	Prompt for <ENTER> to be pressed
190-210	Game initialization
220-290	Determine disk to move and check if valid move
300-370	Determine tower to move to and check if valid move
380-430	Move disk
440	Refresh screen (by redrawing the display)
450	Increment move counter and stop if too many moves
460-470	Check if winner, loop if none yet
480	Print game move results
490-530	Prompt for new game or exit
540-580	Display towers and disks on screen
590	Exit game

Program Listing

```
10 REM      TOWERS
20 DIM F(3,7)
30 D$="***************"
40 FOR Q=1 TO 3 :: FOR R=1 TO 7 :: F(Q,R)=0 :: NEXT R :
   : NEXT Q :: CALL CLEAR
50 DISPLAY AT(1,6):"TOWERS OF HANOI"
60 DISPLAY AT(3,1):"YOU MUST TRANSFER THE DISKS","FROM
   THE LEFT TO THE RIGHT","TOWER, ONE AT A TIME, NEVER
   "
70 DISPLAY AT(6,1):"PUTTING A LARGER DISK ON A","SMALLE
   R ONE."
80 DISPLAY AT(9,1):"HOW MANY DISKS (1-6):"
90 ACCEPT AT(9,23)VALIDATE(DIGIT)SIZE(1):CH$ :: IF CH$<
   "1" OR CH$>"6" THEN 90
100 S=VAL(CH$)
110 M=0
120 CALL CLEAR :: DISPLAY AT(1,1):"IN THIS GAME, WE WIL
    L REFER","TO DISKS BY A NUMERICAL","CODE."
130 DISPLAY AT(5,1):"3 WILL REPRESENT THE SMALL-","EST
    DISK, 5 THE NEXT SIZE,","7 THE NEXT, AND SO ON, UP"
140 DISPLAY AT(8,1):"TO 13.  YOU CAN IDENTIFY A","DISK
    BY THE NUMBER OF ASTER-","ISKS WITH WHICH IT IS MAD
    E."
150 DISPLAY AT(12,1):"THE TOWERS ARE NUMBERED FROM","LE
    FT TO RIGHT, 1 TO 3.  WE","WILL START WITH THE DISK
    S"
```

```
160 DISPLAY AT(15,1):"ON TOWER 1, AND ATTEMPT TO","MOVE
     THEM TO TOWER 3.","                    ","GOOD LUCK!"
170 DISPLAY AT(22,1):"PRESS <ENTER> TO BEGIN:"
180 ACCEPT AT(22,25):CH$ :: IF CH$<>"" THEN 180
190 CALL CLEAR
200 Y=7 :: FOR Q=S TO 1 STEP -1 :: F(1,Y)=Q*2+1 :: Y=Y-
    1 :: NEXT Q
210 GOSUB 540
220 DISPLAY AT(23,1):"DISK TO MOVE:"
230 GOTO 250
240 CALL SOUND(100,1000,1)
250 ACCEPT AT(23,15)VALIDATE(DIGIT)SIZE(2):CH$ :: IF CH
    $="" THEN 240
260 D=VAL(CH$):: IF ((D-1)/2)<1 OR((D-1)/2)>S THEN 240
270 FOR Q=1 TO 3 :: FOR R=1 TO 7 :: IF F(Q,R)=D THEN 29
    0
280 NEXT R :: NEXT Q :: GOTO 240
290 IF F(Q,R-1)>0 THEN 240
300 DISPLAY AT(23,1):"TO WHICH TOWER:"
310 GOTO 330
320 CALL SOUND(100,1000,1)
330 ACCEPT AT(23,17)VALIDATE(DIGIT)SIZE(1):CH$ :: IF CH
    $<"1" OR CH$>"3" THEN 320
340 N=VAL(CH$)
350 FOR Q=1 TO 7 :: IF F(N,Q)<>0 THEN 370
360 NEXT Q :: GOTO 380
370 IF D>=F(N,Q)THEN CALL SOUND(100,1000,1):: GOTO 220
380 FOR U=1 TO 7 :: IF F(N,U)<>0 THEN 400
390 NEXT U
400 U=U-1
410 FOR Q=1 TO 3 :: IF Q=N THEN 430
420 IF F(Q,R)=D THEN F(N,U)=F(Q,R):: F(Q,R)=0 :: Q=3
430 NEXT Q
440 GOSUB 540
450 M=M+1 :: IF M>128 THEN CALL CLEAR :: DISPLAY AT(5,1
    ):"SORRY, BUT I HAVE ORDERS TO","STOP IF YOU MAKE M
    ORE THAN","128 MOVES." :: GOTO 490
460 FOR Q=1 TO 2 :: FOR R=1 TO 7 :: IF F(Q,R)<>0 THEN 2
    20
470 NEXT R :: NEXT Q
480 CALL CLEAR :: DISPLAY AT(5,1):"YOU HAVE PERFORMED T
    HE TASK","IN "&STR$(M)&" MOVES."
490 DISPLAY AT(20,1):"ANOTHER GAME (Y/N):"
500 ACCEPT AT(20,21)VALIDATE("YN")SIZE(1):CH$
510 IF CH$<>"Y" AND CH$<>"N" THEN 500
520 IF CH$="N" THEN 590
530 RUN
540 FOR Q=1 TO 3 :: Z=Q*7 :: FOR R=1 TO 7
550 DISPLAY AT((Q-1)*7+R,1)SIZE(27):" "
560 DISPLAY AT((Q-1)*7+R,Z-INT(F(Q,R)/2)):SEG$(D$,1,F(Q
    ,R))
570 IF F(Q,R)=0 THEN DISPLAY AT((Q-1)*7+R,Z):"*"
580 NEXT R :: NEXT Q :: RETURN
590 CALL CLEAR :: END
```

5. Sherlock's Home

Elementary, my dear Watson! In this game, you're the master detective, and you have to discover "who dunnit" before he does you in. *Sherlock's Home* uses one- and two-word commands, similar to many adventure or fantasy games.

Each time you are called in on a new case there is a different victim. But the suspects remain the same. Not surprising to a trained detective like yourself, is it?

The only fact the police have turned up is that the suspect resides in Sherlock's home. But you have a few hunches of your own and one of them may prove correct—he's out to get you too. You can move from room to room, pick up weapons, and announce who you think committed the dire deed. If any of your guesses (as to the suspect, room, or weapon) are incorrect, you'll discover the reason why, but nothing else. And time isn't on your side—each step you take brings your nemesis closer and closer.

Game Rules

1. Load and run the Sherlock's Home program.
2. The instructions will be displayed if desired.
3. Move from room to room to determine who committed the crime and how it was done.
4. When proposing a solution, you must be in the room where you think the crime was committed.
5. You must be holding the correct weapon to win.
6. You must deduce who committed the crime, where it happened, and what weapon was used.
7. You can move from room to room by typing **GO** followed by the name of the room. The rooms are:

   LIVING ROOM
   STUDY
   BEDROOM
   BASEMENT
   DEN
   KITCHEN
   ATTIC
   DINING ROOM
   WATERCLOSET
   GUEST ROOM

8. If a weapon is in the room, you can pick it up by typing **TAKE** followed by its name, but you can carry only one. Possible weapons are:

   GUN
   KNIFE
   ROPE
   POISON
   CANDLESTICK
   WRENCH

9. To drop a weapon, type **DROP** followed by its name.

10. The suspects and their occupations are:

MR. MISER	Millionaire
MRS. SIPPY	Housewife
CISSI SIPPY	Their daughter
COL. KENTUCKY	Restaurateur
OLIVE PITTS	Fruit grower
HEAVES	Family butler
SPOT	Family dog

11. If you think you have solved the case, make an announcement by entering * (an asterisk) as your command.

12. The game ends when you get the criminal or he gets you.

13. Upon completion you can play again (if you weren't "done in") or exit.

Programming Notes

10-310	Initialization	
320-340	Prompt for displaying instructions	
350-770	Display instructions	
780-900	Game initialization	
910-1000	Main program loop	
	930	Prompt for command
	940	Check detective's announcement
	950	Check command to go to another room
	960	Check command to take a weapon
	970	Check command to drop a weapon
1010	You're too late, you've been "done in"	
1020-1050	Announce "who dunnit"	

1060-1370	Routine for detective's announcement	
	1060	Check for no weapon in hand
	1070-1130	Display announcement screen
	1140-1150	Get your guess as to "who dunnit"
	1160-1210	Check theory in order of room, weapon, and person
	1220	You solved it!
	1230-1260	Play again or exit
	1270	Sorry, wrong room
	1280	Sorry, wrong weapon
	1290	Sorry, wrong person
	1300-1330	Check theory in order of weapon, person, room
	1340-1370	Check theory in order of person, room, weapon
1380-1500	Routine for going to a room	
	1380-1430	Check for valid room
	1440	In that room already
	1450-1480	Special routine for watercloset
	1490	Change room
1510-1650	Routine for taking a weapon	
	1510-1550	Check for valid weapon
	1560	Valid weapon, but not in room
	1570-1590	Check for weapon in room
	1600	Weapon not in room
	1610-1640	Transfer weapon from in the room to in the detective's possession
1660-1740	Routine for dropping a weapon	
	1670	Sorry, not carrying anything to drop
	1680-1710	Check for valid weapon
	1720	Not carrying that weapon
	1730	Transfer weapon from in the detective's possession to in the room
1760	Get random number between 1 and 10	
1770-1790	Check for a keypress to continue; clear the screen	
1800-1840	Display information (room, weapons visible, etc.)	
1850-1870	Check for a keypress to exit the program	

Program Listing

```
10 REM SHERLOCK'S HOME
20 RANDOMIZE
30 DATA FRED,JOHN,CARL,ZEKE,ARCHIBALD,FRANCINE,AGATHA,W
   ILLIAM,MORTIMER,FANNY
40 DATA SMITH,ABERCROMBIE,HARKENFARKER,SNERD,WHOSITS,SP
   ITZINPOT,PEEZINPEPPER,BELEAKE,GYTOPE,PEON
50 DATA LIVING ROOM,DEN,DINING ROOM,STUDY,KITCHEN,WATER
   CLOSET,BEDROOM,ATTIC,GUEST ROOM,BASEMENT
60 DATA GUN,WRENCH,ROPE,CANDLESTICK,KNIFE,POISON
70 DATA MR. MISER,MRS. SIPPY,CISSI SIPPY,COL. KENTUCKY,
   OLIVE PITTS,HEAVES,SPOT
80 FOR X=1 TO 10 :: READ F$(X):: NEXT X :: FOR X=1 TO 1
   0 :: READ L$(X):: NEXT X
90 FOR X=1 TO 10 :: READ P$(X):: NEXT X :: FOR X=1 TO 6
    :: READ W$(X):: NEXT X :: FOR X=1 TO 7 :: READ S$(
   X):: NEXT X
100 GOSUB 1760 :: N$=F$(R):: GOSUB 1760 :: N$=N$&" "&L$
   (R)
110 DIM RM(10,6)
120 V$(1)=" O.K. EVERYONE IS HERE..... WHAT'S THE BIG A
   NNOUNCEMENT?"
130 V$(2)=" LAY IT ON US OH EXALTED    DETECTIVE....."
140 V$(3)=" THINK YOU GOT THE ANSWER?  WE'LL SEE....."
150 V$(4)=" I HOPE YOU ARE NOT WASTING OUR TIME....."
160 V$(5)=" OH, NO.....NOT ANOTHER     THEORY!"
170 V$(6)=" THE VILLIAN IS GETTING     CLOSER.....I HOP
   E YOU HAVE  THE ANSWER....."
180 V$(7)=" SOMEONE IN THIS ROOM MAY DOYOU IN IF YOU DO
   N'T HAVE THECORRECT ANSWER....."
190 V$(8)=" TIME IS RUNNING OUT.....DO YOU HAVE THE ANS
   WER?"
200 V$(9)=" GETTING CLOSE, HUH?  COULD BE DANGEROUS ARO
   UND HERE    SOON....."
210 V$(10)=" THE SURGEON GENERAL HAS    WARNED THAT TOO
    MANY        THEORIES COULD BE HAZARDOUS TO YOUR HE
   ALTH."
220 X$(1)="NOT QUITE..."
230 X$(2)="WRONG AGAIN..."
240 X$(3)="ARE YOU KIDDING?"
250 X$(4)="WRONGO!"
260 X$(5)="SINCE WHEN?"
270 X$(6)="THINK AGAIN..."
280 X$(7)="GOOD TRY, BUT..."
290 X$(8)="THINK SO, HUH?"
300 X$(9)="I DOUBT IT..."
310 X$(10)="BAD JUDGEMENT..."
320 CALL CLEAR :: PRINT " WELCOME TO SHERLOCK'S HOME":
   : : : : : : :
330 PRINT " DO YOU WANT TO REVIEW THE  INSTRUCTIONS (Y/
   N)?": : :
340 CALL KEY(3,K,S):: IF K=78 THEN 780 ELSE IF K<>89 TH
   EN 340
```

cont. on next page

Program Listing—cont.

```
350 CALL CLEAR :: PRINT " WELCOME TO SHERLOCK'S HOME":
    : :
360 PRINT " THE OBJECT OF THE GAME IS  TO GUESS ";CHR$(
    34);"WHO DUNNIT";CHR$(34);".  THE":"COMPUTER WILL C
    HOOSE THE"
370 PRINT "VILLIAN, WHERE THE CRIME WASCOMMITTED, AND T
    HE WEAPON.  BECAUSE THE LOCAL POLICE ARECOMPLETELY
    BAFFLED BY THIS"
380 PRINT "COMPUTER-ASSISTED CRIME,    YOU, AS THE WORL
    D'S FOREMOSTAUTHORITY ON COMPUTER CRIME (WHAT WE IN
     THE TRADE REFER"
390 PRINT "TO AS 'C.C.'), HAVE BEEN    CALLED IN TO DIS
    COVER THE   ANSWER.": : :
400 GOSUB 1770 :: PRINT " THE ONLY ESTABLISHED FACT  IS
     THAT THE CRIME TOOK PLACE"
410 PRINT "SOMEWHERE IN AN OLD DESERTEDMANSION CALLED "
    ;CHR$(34);"SHERLOCK'S":"HOME";CHR$(34);" (ONCE OWNE
    D BY THE"
420 PRINT "RICH AND FAMOUS MYSTERY     AUTHOR, JOHN SPE
    NCER        SHERLOCK III).": : :
430 GOSUB 1760 :: Q$="" :: IF R<4 THEN 460
440 Q1$=Q$&"POOR " :: IF R<7 THEN Q$=Q1$ :: GOTO 460
450 Q$=Q$&"HAPLESS "
460 PRINT " ";N$;", ";"THE ";Q$;"VICTIM, ";"REFUSES ";"
    TO ";"ANSWER ";"ANY ";"OF ";"THE ";"QUESTIONS ";"PO
    SED ";"BY ";"THE ";"POLICE, ";
470 PRINT "SO ";"YOU ";"HAVE ";"NOTHING ";"MORE ";"TO "
    ;"GO ";"ON ";"EXCEPT ";"YOUR ";"WIT, ";"COURAGE, ";
    "AND ";"CUNNING.": : :
480 GOSUB 1770 :: PRINT " THE ROOMS IN THE HOUSE ARE AS
     FOLLOWS:": : :
490 FOR X=1 TO 9 STEP 2 :: PRINT TAB(2);P$(X);TAB(16);P
    $(X+1):: NEXT X
500 PRINT : :" YOU MAY ENTER ANY ROOM IN  THE HOUSE BY
    TYPING 'GO '   AND THE NAME OF THE ROOM.": : :
510 GOSUB 1770
520 PRINT " THE WEAPONS ARE:": : :
530 FOR X=1 TO 6 :: PRINT TAB(11);W$(X):: NEXT X
540 PRINT : :" THESE ITEMS LAY SCATTERED  THROUGHOUT TH
    E HOUSE.": : :
550 GOSUB 1770
560 PRINT " THE POSSIBLE SUSPECTS (AND THEIR OCCUPATION
    S) ARE:": : :
570 PRINT "MR. MISER";TAB(15);"MILLIONAIRE":"MRS. SIPPY
    ";TAB(15);"HOUSEWIFE":"CISSI SIPPY";TAB(15);"THEIR
    DAUGHTER"
580 PRINT "COL. KENTUCKY";TAB(15);"RESTAURANTEUR":"OLIV
    E PITTS";TAB(15);"FRUIT GROWER"
590 PRINT "HEAVES";TAB(15);"FAMILY BUTLER":"SPOT";TAB(1
    5);"FAMILY DOG": : :
600 PRINT " ALL OF THE ABOVE HAD A     MOTIVE TO DO AWA
    Y WITH" :: PRINT N$;".": : :
610 GOSUB 1770
```

```
620 PRINT " YOUR JOB IS TO GO FROM ROOMTO ROOM AND DECI
    DE WHO DID  IN ";N$;".": : :
630 PRINT " WHEN YOU WISH TO VENTURE A GUESS AS TO 'WHO
     DUNNIT',  ENTER '*' AS YOUR COMMAND.  EVERYONE WI
    LL THEN COME TO"
640 PRINT "THE ROOM YOU ARE IN FOR THE BRILLIANT ANNOUN
    CEMENT.": : :
650 PRINT " YOU MUST ANNOUNCE YOUR     THEORY IN THE RO
    OM YOU THINKTHE CRIME HAPPENED IN.  FOR EXAMPLE, IF
     YOU THINK IT"
660 PRINT "TOOK PLACE IN THE DEN, YOU  MUST ANNOUNCE IT
     IN THE DEN.": :
670 GOSUB 1770
680 PRINT " ALSO, YOU MUST BE CARRYING THE WEAPON YOU T
    HINK WAS    USED TO DO THE DIRTY DEED.": : :
690 PRINT " YOU MAY PICK UP WEAPONS BY TYPING 'TAKE ' A
    ND THE NAME OF THE WEAPON.  FOR EXAMPLE,TO PICK UP
    THE GUN, THE"
700 PRINT "COMMAND WOULD BE 'TAKE GUN'.": : :" YOU CAN
    ONLY CARRY ONE     WEAPON AT A TIME, BUT CAN   DROP
     ANY WEAPON WITH THE"
710 PRINT "COMMAND 'DROP ' AND THE NAMEOF THE WEAPON.":
     : :
720 GOSUB 1770
730 PRINT " AS A FINAL NOTE, THE       VILLIAN HAS AN A
    VERSION TO  GETTING CAUGHT.  AS TIME    GOES ON, TH
    E VILLIAN WILL"
740 PRINT "TRY TO DO YOU IN, SO TRY TO DISCOVER WHO DID
     THE DEED   BEFORE YOU BECOME THE NEXT  VICTIM.": :
     :
750 PRINT " BY THE WAY, EVERY GOOD     DETECTIVE HAS A
    LITTLE NOTE BOOK HE CARRIES AROUND WITH HIM.  IT MI
    GHT BE A GOOD"
760 PRINT "IDEA IF YOU DID TOO.": : :
770 GOTO 330
780 CALL CLEAR
790 GOSUB 1760 :: C(1)=R
800 GOSUB 1760 :: IF R>6 THEN 800
810 C(2)=R
820 GOSUB 1760 :: IF R>7 THEN 820
830 C(3)=R :: UL=50 :: GOSUB 1760 :: P(1)=R :: P(2)=0 :
    : GOSUB 1760 :: UL=UL-R
840 FOR X=1 TO 10 :: RM(X,0)=0 :: F(X)=0 :: NEXT X
850 FOR X=1 TO 6
860 GOSUB 1760 :: IF R>6 THEN 860
870 IF F(R)=1 THEN 860
880 T=R :: F(R)=1
890 GOSUB 1760 :: RM(R,RM(R,0)+1)=T :: RM(R,0)=RM(R,0)+
    1
900 NEXT X
910 FOR Z=1 TO UL
920 GOSUB 1800
930 PRINT : :: INPUT "COMMAND ? ":I$
```

cont. on next page

```
940 IF I$="*" THEN GOSUB 1060 :: GOTO 1000
950 IF SEG$(I$,1,2)="GO" THEN GOSUB 1380 :: GOTO 1000
960 IF SEG$(I$,1,4)="TAKE" THEN GOSUB 1510 :: GOTO 1000
970 IF SEG$(I$,1,4)="DROP" THEN GOSUB 1660 :: GOTO 1000
980 PRINT : :" SORRY, SIR, BUT I DON'T    UNDERSTAND WH
    AT YOU MEAN    BY ";I$: : :
990 GOSUB 1770
1000 NEXT Z
1010 CALL CLEAR :: PRINT " SORRY TO REPORT, SIR, THAT Y
    OU HAVE NOW JOINED POOR":N$;" IN":"THAT GREAT PRECI
    NCT HOUSE":"IN THE SKY.": : :
1020 PRINT " CHALK UP ANOTHER CRIME TO":S$(C(3));".": :
     :
1030 PRINT " BY THE WAY, THE WEAPON WAS THE ";W$(C(2));
    ".": : :
1040 PRINT " AND IT WAS DONE IN THE":P$(C(1));".": : :
1050 GOTO 1850
1060 IF P(2)=0 THEN PRINT : :" SORRY, BUT YOU ARE NOT
       CARRYING A WEAPON.  PLEASE  REMEMBER THE RULES,
    SIR!": : : :: GOSUB 1770 :: RETURN
1070 CALL CLEAR
1080 Q1=5 :: IF Z>(Z>3)*2 THEN Q1=10
1090 GOSUB 1760 :: IF R>Q1 THEN 1090
1100 PRINT V$(R): :
1110 PRINT " ACCORDING TO YOU, THE CRIMEWAS COMMITED IN
     THE ";P$(P(1))&" ";"WITH ";"THE ";W$(P(2));".": :
1120 PRINT "   NOW THE QUESTION IS":TAB(8);"'WHO DUNNIT
    '?": : :: FOR X=1 TO 7 :: PRINT TAB(7);STR$(X);". "
    ;S$(X):: NEXT X
1130 PRINT :" WHAT IS THE CULPRIT'S      NUMBER? (PRESS
     1-7)"
1140 CALL KEY(3,K,S):: IF K<49 OR K>55 THEN 1140
1150 I=K-48 :: CALL HCHAR(23,23,K)
1160 GOSUB 1760 :: IF R>3 THEN 1160
1170 CALL CLEAR
1180 ON R GOTO 1190,1300,1340
1190 IF C(1)<>P(1)THEN 1270
1200 IF C(2)<>P(2)THEN 1280
1210 IF C(3)<>I THEN 1290
1220 PRINT " CONGRATULATIONS, SIR !!!!  YOU SOLVED THE
    CASE (I      PERSONALLY NEVER DOUBTED    YOUR VERAC
    ITY FOR A MINUTE)."
1230 PRINT : : :" WOULD YOU LIKE ANOTHER     ASSIGNMENT
    , OR DO YOU WANT  TO TAKE A VACATION?": :TAB(7);"(P
    RESS A OR V)": : : :
1240 CALL KEY(3,K,S):: IF K<>65 AND K<>86 THEN 1240
1250 CALL CLEAR :: IF K=86 THEN PRINT " HAVE A GOOD VAC
    ATION, SIR  AND HURRY BACK SOON!": : : : : : : : ::
     GOTO 1850
1260 IF K=65 THEN RUN
1270 GOSUB 1760 :: PRINT :" ";X$(R):"THE ";P$(P(1));" W
    AS NOT THE":"CORRECT ROOM!": : : :: GOSUB 1770 :: R
    ETURN
```

```
1280 GOSUB 1760 :: PRINT :" ";X$(R):"THE ";W$(P(2));" I
    S THE WRONG":"WEAPON, SIR!": : : :: GOSUB 1770 :: R
    ETURN
1290 GOSUB 1760 :: PRINT :" ";X$(R):S$(I);" HAS":"A PER
    FECT ALIBI, SIR!": : : :: GOSUB 1770 :: RETURN
1300 IF C(2)<>P(2)THEN 1280
1310 IF C(3)<>I THEN 1290
1320 IF C(1)<>P(1)THEN 1270
1330 GOTO 1220
1340 IF C(3)<>I THEN 1290
1350 IF C(1)<>P(1)THEN 1270
1360 IF C(2)<>P(2)THEN 1280
1370 GOTO 1220
1380 IF LEN(I$)<4 THEN 1420
1390 FOR X=1 TO 10
1400 IF SEG$(I$,3,LEN(I$))=P$(X)OR SEG$(I$,4,LEN(I$))=P
    $(X)THEN 1440
1410 NEXT X
1420 PRINT : :" SORRY, I CANNOT GO THERE,  SIR.  IT DOE
    S NOT SEEM TO   BE IN SHERLOCK'S HOME": : :
1430 GOSUB 1770 :: RETURN
1440 IF X=P(1)THEN PRINT : :" SIR, WE SEEM TO BE IN THA
    T ROOM ALREADY !": : : :: GOSUB 1770 :: RETURN
1450 IF X<>6 THEN 1490
1460 GOSUB 1760 :: IF R>4 THEN 1490
1470 PRINT : :" SORRY, THE WATERCLOSET IS  IN USE AT TH
    IS TIME.  TRY   AGAIN LATER....": : :
1480 GOSUB 1770 :: RETURN
1490 P(1)=X
1500 RETURN
1510 IF LEN(I$)<6 THEN 1550
1520 FOR X=1 TO 6
1530 IF SEG$(I$,5,LEN(I$))=W$(X)OR SEG$(I$,6,LEN(I$))=W
    $(X)THEN 1560
1540 NEXT X
1550 PRINT : :" SORRY, SIR, I DON'T THINK  THAT WEAPON
    IS IN THIS      CASE !": : : :: GOSUB 1770 :: RETUR
    N
1560 IF RM(P(1),0)=0 THEN PRINT : :" SORRY, SIR, BUT TH
    ERE SEEM TO BE NO WEAPONS IN THIS    ROOM!": : : ::
     GOSUB 1770 :: RETURN
1570 FOR Y=1 TO RM(P(1),0)
1580 IF RM(P(1),Y)=X THEN 1610
1590 NEXT Y
1600 PRINT : :" SORRY, SIR, BUT I DON'T SEETHAT WEAPON
    IN THIS ROOM!": : : :: GOSUB 1770 :: RETURN
1610 T=P(2):: P(2)=RM(P(1),Y):: RM(P(1),Y)=T
1620 IF T>0 THEN RETURN
1630 RM(P(1),Y)=RM(P(1),RM(P(1),0))
1640 RM(P(1),0)=RM(P(1),0)-1
1650 RETURN
1660 IF LEN(I$)<6 THEN 1710
```

cont. on next page

Program Listing—cont.

```
1670 IF P(2)=0 THEN PRINT : :" SORRY, BUT I AM NOT
       CARRYING ANTHING, SIR!": : : :: GOSUB 1770 :: RE
    TURN
1680 FOR X=1 TO 6
1690 IF SEG$(I$,5,LEN(I$))=W$(X)OR SEG$(I$,6,LEN(I$))=W
    $(X)THEN 1720
1700 NEXT X
1710 GOTO 1550
1720 IF X<>P(2)THEN PRINT : :" SORRY, SIR, BUT YOU ARE
    NOTCARRYING THAT WEAPON!": : : :: GOSUB 1770 :: RET
    URN
1730 RM(P(1),0)=RM(P(1),0)+1 :: RM(P(1),RM(P(1),0))=P(2
    ):: P(2)=0
1740 RETURN
1750 GOTO 1750
1760 R=INT(RND*10)+1 :: RETURN
1770 PRINT " PRESS ANY KEY TO CONTINUE": :
1780 CALL KEY(3,K,S):: IF S=0 THEN 1780
1790 CALL CLEAR :: RETURN
1800 CALL CLEAR :: PRINT "ROOM: ";P$(P(1)):"CARRYING: "
    ;W$(P(2)):"VICTIM: ";N$:"VISIBLE: ";
1810 IF RM(P(1),0)=0 THEN 1840
1820 FOR X=1 TO RM(P(1),0)
1830 PRINT W$(RM(P(1),X));" ";:: NEXT X
1840 PRINT :: RETURN
1850 PRINT :"  PRESS <ENTER> TO CONTINUE": :
1860 CALL KEY(3,K,S):: IF K<>13 THEN 1860
1870 CALL CLEAR :: END
```

6. Attack of the Zargons

Due to recently enacted weapons-control laws and your lawless attitudes, you are the only person remaining on earth who has a weapon. As the only defender of the planet in the face of an onslaught of invading alien hordes from the planet Zargon, you are in a unique position. You feel the weight of responsibility and the urgency of the moment as you pull out your patented ACME ALIEN INVADER POWER DRAINER (page 342 in the Whole Planet Catalog) and fearlessly begin firing at the approaching ship.

Such is your power and responsibility in *Attack of the Zargons.*

Game Rules

1. Load and run the Attack of the Zargons program.
2. After the game is loaded there is a short delay before the action begins.
3. Press **S** to move your weapon base left and **D** to move it right. The left and right direction arrows, respectively, are on the front of these keys.
4. Holding down either direction key will make your weapon base move faster.
5. Press the <SPACE BAR> to stop the movement of your weapon base.
6. Fire at the invader by pressing **E.** On the front of this key is an up arrow.
7. You receive 100 energy units for hitting the ship.
8. You expend 5 energy units for each shot taken.
9. You lose half of your accumulated energy units if you are hit.
10. The game ends when the enemy ship reaches the bottom of the screen.

11. To win, you must have at least 1500 points when the game is over.

Programming Notes

This game uses sprites and sounds to create a simple arcade-style game. The Zargonian ship and your weapon base (and other special effects) are created with sprites.

10-30	Welcoming text screen	
40-110	Game initialization	
120-210	Main game loop	
	130	Draw ship and weapon
	140	Check for a keypress to fire at the Zargonian ship
	150	Zargonian ship fires only 10% of time (maximum)
	160-190	Adjust for downward motion of the Zargonian ship
	200	Calculate increment amount
220-230	Game end and check for winning score	
240-250	Low score (less than 3000), print message	
260-320	High score (3000 or greater), print message	
330-360	Another game or end the program	
370-390	Score adjustment (add to or subtract from the score)	
400-460	Zargon fires at weapon	
470	Check for keypress	
480-490	Increase the speed of the weapon base's movement	
500	Stop weapon base	
510	Move weapon base left	
520	Move weapon base right	
530	Fire weapon base	
540	Adjust motion of the weapon base	
550	Return to main game loop	
560-640	Fire at the Zargoniann ship and adjust score	
650-670	Music printing routine	
680	Data statements for states	

Program Listing

```
10 REM ATTACK OF THE ZARGONS
20 CALL CLEAR :: Z$="ATTACK OF THE ZARGONS" :: Y=1 :: G
    OSUB 650 :: Z$="THEY ARE COMING...." :: Y=5 :: X=1
     :: GOSUB 660
30 Z$="....GOOD LUCK!" :: Y=10 :: X=1 :: GOSUB 660
40 DIM ST$(15):: FOR X=1 TO 15 :: READ ST$(X):: NEXT X
50 CALL CLEAR :: T=0 :: R1=2 :: C=15 :: K=8 :: IA=3
60 CALL CHAR(96,"000000183C7EFFFF"):: REM SPACE SHIP
70 CALL CHAR(97,"1818183C2466C3FF")
80 CALL CHAR(98,"C3C366663C3C1818"):: REM HIS SHOTS
90 CALL CHAR(99,"0000183C3C180000"):: REM OUR SHOTS
100 CALL SPRITE(#1,96,5,20,2,0,C)
110 CALL SPRITE(#2,97,9,176,R1,0,SP)
120 FOR Y=20 TO 130 STEP 5
130 CALL POSITION(#1,J,K):: CALL LOCATE(#1,Y,K)
140 GOSUB 470
150 IF RND<.1 THEN GOSUB 400
160 CALL POSITION(#1,J,K)
170 IF K<128 THEN AW=0
180 IF K>128 AND AW=0 THEN AW=1 :: GOTO 200
190 GOTO 140
200 GOSUB 370
210 NEXT Y
220 CALL CLEAR :: CALL DELSPRITE(ALL)
230 IF T>1500 THEN 260
240 DISPLAY AT(1,1):"THIS HAS BEEN A SAD DAY FOR","EART
    H.  ALL HAS BEEN LOST.","THE ZARGONS HAD ENOUGH POW
    ER"
250 DISPLAY AT(4,1):"LEFT TO DESTROY THE EARTH.","YOUR
    SCORE OF "&STR$(T)&" JUST","WASN'T GOOD ENOUGH." ::
     GOTO 330
260 DISPLAY AT(1,1):"CONGRATULATIONS...","YOU HAVE SUCC
    ESSFULLY DE-","FENDED THE EARTH AGAINST THE"
270 DISPLAY AT(4,1):"ZARGONS."
280 DISPLAY AT(6,1):"YOUR SCORE OF "&STR$(T)&" WAS VERY
    ","COMMENDABLE."
290 DISPLAY AT(10,1):"THANKS TO YOUR BRAVERY AND","SKIL
    L IN COMBAT, YOU HAVE","BEEN AWARDED (BY ACT OF"
300 DISPLAY AT(13,1):"CONGRESS), THE STATE OF"
310 X=RND*33 :: IF X<1 OR X>15 THEN 310
320 DISPLAY AT(14,1):ST$(X);" IN DEEP GRATITUDE."
330 DISPLAY AT(20,1):"PLAY AGAIN (Y/N):"
340 ACCEPT AT(20,19)VALIDATE("YN")SIZE(1):CH$ :: IF CH$
    <>"Y" AND CH$<>"N" THEN 340
350 IF CH$="Y" THEN 50
360 CALL CLEAR :: END
370 IF T=0 THEN T=-2
380 M=.5 :: IF T<0 THEN M=1+(1/(ABS(T)/100))
390 RETURN
400 CALL POSITION(#1,J,K):: CALL SPRITE(#3,98,5,J+8,K,6
    4,0)
```

cont. on next page

Program Listing—cont.

```
410 CALL SOUND(80,J*10+110,3)
420 CALL POSITION(#3,J,K):: IF J<160 THEN 420
430 CALL DELSPRITE(#3)
440 CALL POSITION(#2,L,M):: IF K<(M-8)OR K>(M+8)THEN 46
    0
450 CALL SOUND(80,-2,3):: GOSUB 370 :: T=INT(T*M):: DIS
    PLAY AT(23,21)SIZE(6):T
460 RETURN
470 CALL KEY(0,A,S):: IF S=0 THEN 550
480 IF S=-1 AND SP<>0 THEN SP=SP+IA :: IF ABS(SP)>40 TH
    EN SP=40*SGN(IA)
490 IF S=-1 THEN 540
500 IF A=32 THEN SP=0
510 IF A=83 THEN SP=-10 :: IA=-IA
520 IF A=68 THEN SP=10 :: IA=ABS(IA)
530 IF A=69 THEN 560
540 CALL MOTION(#2,0,SP)
550 RETURN
560 CALL POSITION(#2,J,K):: CALL POSITION(#1,L,M):: CAL
    L SPRITE(#4,99,9,J,K)
570 L=J-INT((J-L)/16+1)*16
580 CALL SOUND(80,K*10+110,3):: FOR X=J-8 TO L STEP -16
     :: CALL LOCATE(#4,X,K):: NEXT X
590 CALL COINC(#1,#4,8,K)
600 CALL DELSPRITE(#4)
610 IF K=0 THEN 630
620 T=T+100 :: CALL SOUND(80,-6,2)
630 T=T-5 :: DISPLAY AT(23,21)SIZE(6):T
640 RETURN
650 X=INT(14-LEN(Z$)/2)
660 FOR Z=1 TO LEN(Z$):: DISPLAY AT(Y,X+Z-1):SEG$(Z$,Z,
    1):: IF SEG$(Z$,Z,1)<>" " THEN CALL SOUND(1,RND*400
    0+200,4)
670 NEXT Z :: RETURN
680 DATA ALASKA,UTAH,NEVADA,ARIZONA,NEW MEXICO,WYOMING,
    MONTANA,SOUTH DAKOTA,NORTH DAKOTA,IDAHO,IOWA,KANSAS
    ,NEBRASKA,OKLAHOMA,WISCONSIN
```

7. Phaser Practice

Phaser Practice is a game of luck. However, it does show how to choose random points within set boundaries. This is needed to determine where on the target the phaser "blast" is going to hit.

Game Rules

1. Load and run the Phaser Practice program.
2. The directions will be displayed.
3. Enter the number of players. The game allows one to three players.
4. Enter the names of each player as prompted by the program.
5. The target is drawn and each player, in turn, will be prompted for the shot to use.
6. Press a 1, 2, or 3 and then press <ENTER> for your shot.
7. The game ends at the conclusion of a round in which a player's score is at least 250 points.
8. Upon completion, the players are ranked according to their score. Then you can play again or exit.

Programming Notes

10-80	Game initialization	
90-170	Display instructions	
180-200	Get number of players and check for validity	
210-260	Get player names and call target-drawing routine	
270-280	Game set up	
290-530	Main program loop	
	290	Print round number
	310	Print prompt character beside name
	320-350	Get desired shot
	360-450	Set parameters for each type of shot
	460-500	Show shot printing type and new score

510-530	Check for winner and loop if none
540-570	Sort names and scores
580-590	Display game results
600-630	Play again or exit
640-780	Plot the individual shot and display it
790-800	Generate a random number between 1 and 10
810-910	Draw the target
920-1010	Data for target borders

Program Listing

```
10 REM      PHASER PRACTICE
20 RANDOMIZE
30 DIM X1(4,4),Y1(4,4)
40 FOR J=0 TO 4 :: FOR K=1 TO 4 :: READ X1(J,K):: NEXT
   K :: NEXT J
50 FOR J=0 TO 4 :: FOR K=1 TO 4 :: READ Y1(J,K):: NEXT
   K :: NEXT J
60 T$(1)="OVERARM" :: T$(2)="SIDEARM" :: T$(3)="UNDERAR
   M"
70 H$(0)="MISS !" :: H$(1)="10 POINTER !" :: H$(2)="20
   POINTER !" :: H$(3)="30 POINTER !" :: H$(5)="BULLSE
   YE !"
80 CALL CHAR(96,"E0E0E0000000000000")
90 DISPLAY AT(1,6)ERASE ALL:"PHASER PRACTICE"
100 DISPLAY AT(3,1):"THE OBJECT OF THE GAME IS","TO BE
   THE FIRST PLAYER TO","ACCUMULATE 250 POINTS BY"
110 DISPLAY AT(6,1):"FIRING A PHASER AT A TARGET."
120 DISPLAY AT(8,1):"THE ZONES ON THE TARGET","HAVE VAL
   UES OF 10, 20, 30,","AND 50 POINTS."
130 DISPLAY AT(12,1):"SHOT & TYPE     VALUE","---- - --
   --     -----"
140 DISPLAY AT(14,2):"1) ";T$(1):: DISPLAY AT(14,17):"5
   0 OR MISS"
150 DISPLAY AT(15,2):"2) ";T$(2):: DISPLAY AT(15,17):"1
   0, 20 OR 30"
160 DISPLAY AT(16,2):"3) ";T$(3):: DISPLAY AT(16,17):"A
   NYTHING"
170 DIM W(10):: R=0 :: M=0
180 DISPLAY AT(18,1):"NUMBER OF PLAYERS (1-3):"
190 ACCEPT AT(18,26)VALIDATE(DIGIT):CH$ :: IF CH$<"1" O
   R CH$>"3" THEN 190
200 N=VAL(CH$)
210 CALL CLEAR :: PRINT "PLEASE LIMIT NAMES TO 7" :: PR
   INT "CHARACTERS OR LESS!"
220 FOR X=1 TO N
230 PRINT :: PRINT "NAME OF PLAYER #"&STR$(X);
240 INPUT ": ":A$(X):: IF LEN(A$(X))>7 THEN CALL SOUND(
   100,-5,3):: GOTO 230
250 S(X)=0 :: NEXT X
260 GOSUB 810
```

```
270 DISPLAY AT(20,1):"ROUND:"
280 FOR X=1 TO N :: DISPLAY AT(19+X,14):A$(X);":";:: DI
    SPLAY AT(19+X,24):S(X):: NEXT X
290 R=R+1 :: DISPLAY AT(20,8)SIZE(3):R
300 FOR X=1 TO N
310 DISPLAY AT(19+X,13)SIZE(1):">";
320 DISPLAY AT(22,1)SIZE(12):" ";:: DISPLAY AT(23,1)SIZ
    E(14):"SHOT (1-3):";
330 ACCEPT AT(23,13)VALIDATE("123Q")SIZE(1):CH$ :: IF C
    H$="Q" THEN 600
340 IF CH$<"1" OR CH$>"3" THEN 330
350 T=VAL(CH$)
360 ON T GOTO 370,380,390
370 P1=.6 :: P2=.6 :: P3=.6 :: P4=.6 :: GOTO 400
380 P1=.99 :: P2=.77 :: P3=.43 :: P4=.0001 :: GOTO 400
    390 P1=.95 :: P2=.75 :: P3=.45 :: P4=.05
400 U=RND
410 IF U>P1 THEN B=50 :: GOTO 460
420 IF U>P2 THEN B=30 :: GOTO 460
430 IF U>P3 THEN B=20 :: GOTO 460
440 IF U>P4 THEN B=10 :: GOTO 460
450 B=0
460 S(X)=S(X)+B :: DISPLAY AT(22,1)SIZE(12):T$(T);:: TS
    =B/10
470 GOSUB 640 :: DISPLAY AT(23,1)SIZE(27):H$(TS)
480 DISPLAY AT(19+X,13)SIZE(1):" ";:: DISPLAY AT(19+X,2
    4)SIZE(4):S(X);
490 FOR DE=1 TO 200 :: NEXT DE
500 NEXT X
510 FOR X=1 TO N
520 IF S(X)>=250 THEN 540
530 NEXT X :: GOTO 290
540 FOR X=1 TO N-1 :: FOR Y=X+1 TO N
550 IF S(X)>=S(Y)THEN 570
560 T=S(X):: S(X)=S(Y):: S(Y)=T :: T1$=A$(X):: A$(X)=A$
    (Y):: A$(Y)=T1$
570 NEXT Y :: NEXT X
580 DISPLAY AT(5,1)ERASE ALL:"WE HAVE A WINNER IN ROUND
    ","#"&STR$(R)&" !"
590 DISPLAY AT(8,1):"FINAL SCORES:" :: FOR X=1 TO N ::
    DISPLAY AT(9+X,1):A$(X);" SCORED";S(X);"POINTS" ::
    NEXT X
600 DISPLAY AT(23,1)SIZE(27):"PLAY AGAIN (Y/N):"
610 ACCEPT AT(23,19)VALIDATE("YN"):CH$ :: IF CH$<>"Y" A
    ND CH$<>"N" THEN 610
620 IF CH$="Y" THEN CALL CLEAR :: R=0 :: GOTO 180
630 DISPLAY AT(5,1)ERASE ALL:"THANKS FOR THE GAME!" ::
    END
640 L=B/10 :: IF L=5 THEN L=4
650 X2=0 :: Y2=0
660 A=X1(L,1):: B=X1(L,2):: C=X1(L,3):: D=X1(L,4)
670 X2=INT(RND*(B-A+1))+A :: IF RND>.5 THEN X2=X2+C-A
680 IF X2<A OR X2>D OR(X2>B AND X2<C)THEN 670
```

cont. on next page

Program Listing—cont.

```
690 A=Y1(L,1):: B=Y1(L,2):: C=Y1(L,3):: D=Y1(L,4)
700 Y2=INT(RND*(B-A+1))+A :: IF RND>.5 THEN Y2=Y2+C-A
710 IF Y2<A OR Y2>D OR(Y2>B AND Y2<C)THEN 700
720 C1=119-X2 :: R1=160-Y2 :: C1=ABS(C1/R1)*SGN(C1):: C
    =SGN(R1)*2
730 CALL SPRITE(#1,96,5,160,119)
740 CALL SOUND(80,-1,3):: FOR J=1 TO R1 STEP C :: CALL
    LOCATE(#1,160-J,119+(J*C1)):: NEXT J
750 IF L=0 THEN CALL SOUND(80,-6,3):: GOTO 770
760 CALL SOUND(80,-2,3)
770 CALL DELSPRITE(#1)
780 RETURN
790 RN=INT(RND*11):: IF RN<1 OR RN>10 THEN 790
800 RETURN
810 CALL CLEAR
820 X=6 :: Y=4 :: GOSUB 870
830 X=12 :: Y=8 :: GOSUB 870
840 X=18 :: Y=12 :: GOSUB 870
850 X=24 :: Y=16 :: GOSUB 870
860 RETURN
870 X$="****************************"
880 VT=INT(10-Y/2):: HT=INT(14-X/2)
890 DISPLAY AT(VT,HT):SEG$(X$,1,X)
900 FOR J=1 TO Y-1 :: DISPLAY AT(VT+J,HT):SEG$(X$,1,1);
    :: DISPLAY AT(VT+J,HT+X-1):SEG$(X$,1,1);:: NEXT J
910 DISPLAY AT(VT+Y,HT):SEG$(X$,1,X):: RETURN
920 DATA 8,25,213,233
930 DATA 29,49,189,209
940 DATA 53,73,165,185
950 DATA 77,97,141,161
960 DATA 101,137,101,137
970 DATA 1,10,142,155
980 DATA 14,27,126,139
990 DATA 30,43,110,123
1000 DATA 46,59,94,106
1010 DATA 62,90,62,90
```

8. Acey-Ducey

This is a game for gamblers. Three cards are picked at random, and you bet on whether or not the value of the third card will be between the first two.

The cards are displayed on the screen using subroutines to draw the blank cards and the card's symbols in their correct positions. The card symbols—hearts, diamonds, spades, and clubs—are drawn using custom-designed characters.

One suggestion for improving the game (and testing your programming skills) is to add a real "deck" and deal from it. As the game is programmed, the cards are drawn at random from an "unlimited" deck; a card (for example, the ten of hearts) can occur more than once in a game or even in one hand.

Game Rules

1. Load and run the Acey-Ducey program.
2. The instructions will be displayed.
3. Press <ENTER> to start the game.
4. You begin with a $100.00 stake.
5. Two cards will be shown face up.
6. Enter your bet. You can bet nothing (by entering 0) or up to your full stake.
7. The third card will be dealt.
8. If the third card's value is between the values of the first two cards, you win the amount of your wager. If the third card's value is less than the first card, greater than the second card, or equal to either of the first two cards, you lose the amount of your wager.
9. Card ranking is 2 through 10, Jack, Queen, King, Ace.
10. The game ends when you have no more money.
11. When the game ends, you can play again or exit.

Programming Notes

10-90	Program initialization
100-150	Dollars and cents display routine
160-170	Rounding routine
180-250	Display directions
260-270	Wait for an <ENTER> before proceeding
280-320	Initialize screen and display
330	Check for player with no money
340-360	Pick first two cards
370	Pick suit for first card
380	Show first card
390	Pick suit for second card
400	Show second card
410-460	Get wager; exit or check for validity of wager
470-480	Pick third card and suit
490	Show third card
500	Does the bet win?
510	Yes, so add to your stake
520	No, so decrease your stake
530	Short pause and loop for next hand
540	Out of money, therefore game is over
550-590	Play again or exit
600	Print blank cards on screen (no markings)
610	Check if card is an ace
620-650	Display markings for cards numbered 4 through 10
660-680	Display markings for Ace and cards numbered 2 and 3
690-720	Display markings for Jack, Queen, and King

Program Listing

```
10 REM  ACEY-DUCEY
20 CALL CHAR(96,"1010387C38101000")
30 CALL CHAR(97,"10387C3810103800")
40 CALL CHAR(98,"6C7C383810100000")
50 CALL CHAR(99,"10387C7C38103800")
60 CALL COLOR(9,2,16,10,16,1)
70 CALL CHAR(104,"FFFFFFFFFFFFFFFF")
80 RANDOMIZE
90 GOTO 180
100 Q=INT(Q*100+.5)/100
110 Q$=STR$(Q):: IF Q=0 THEN Q$=""
120 IF INT(Q)=0 THEN Q$="0"&Q$
130 IF Q=INT(Q)THEN Q$=Q$&".00" :: GOTO 150
```

```
140 IF ASC(SEG$(Q$,LEN(Q$)-2,1))<>46 THEN Q$=Q$&"0"
150 RETURN
160 R=INT(RND*17):: IF R<2 OR R>14 THEN 160
170 RETURN
180 CALL CLEAR
190 DISPLAY AT(1,1):"ACEY-DUCEY IS PLAYED IN THE","FOLL
    OWING WAY:"
200 DISPLAY AT(4,1):"THE DEALER (COMPUTER) DEALS","TWO
    CARDS FACE UP.  YOU","HAVE THE OPTION TO BET OR"
210 DISPLAY AT(7,1):"NOT, DEPENDING ON WHETHER","YOU TH
    INK THE NEXT CARD","DEALT WILL HAVE A VALUE BE-"
220 DISPLAY AT(10,1):"TWEEN THE FIRST TWO."
230 DISPLAY AT(12,1):"IF YOU DO NOT WANT TO PLACE","A B
    ET, ENTER'0' AS YOUR BET."
240 DISPLAY AT(14,1):"TIES WITH YOUR UPPER AND","LOWER
    CARDS ARE AUTOMATI-","CALLY WON BY THE HOUSE."
250 DISPLAY AT(18,1):"ENTERING 'Q' AS YOUR BET","WILL A
    LLOW YOU TO END THE","GAME EARLY."
260 DISPLAY AT(23,1):"PRESS <ENTER> TO CONTINUE:"
270 ACCEPT AT(23,28)SIZE(1):CH$ :: IF CH$<>"" THEN 270
280 CALL CLEAR
290 DISPLAY AT(21,1):"STAKE:" :: DISPLAY AT(22,1):"YOUR
     WAGER: $"
300 Q=100
310 GOSUB 100
320 DISPLAY AT(21,8)SIZE(15):Q$
330 IF Q<.02 THEN 540
340 GOSUB 160 :: A=R
350 GOSUB 160 :: B=R
360 IF A>=B-1 THEN 340
370 GOSUB 160 :: IF R>4 THEN 370
380 CV=A :: SU=R :: X=1 :: Y=1 :: GOSUB 600
390 GOSUB 160 :: IF R>4 THEN 390
400 CV=B :: SU=R :: X=1 :: Y=8 :: GOSUB 600
410 ACCEPT AT(22,14)SIZE(10)VALIDATE("1234567890.Q"):CH
    $ :: IF CH$="" THEN 410
420 IF CH$="Q" THEN DISPLAY AT(5,1)ERASE ALL:"THANKS FO
    R PLAYING!" :: END
430 M=VAL(CH$):: IF M<0 THEN CALL SOUND(100,-2,3):: GOT
    O 410
440 IF M=0 THEN DISPLAY AT(23,1)SIZE(20):"CHICKEN !" ::
     GOTO 530
450 IF M<=Q THEN 470
460 GOSUB 100 :: DISPLAY AT(23,1)SIZE(27):"YOU ONLY HAV
    E $";Q$ :: GOTO 330
470 GOSUB 160 :: C=R
480 GOSUB 160 :: IF R>4 THEN 480
490 CV=C :: SU=R :: X=1 :: Y=15 :: GOSUB 600
500 IF C<=A OR C>=B THEN 520
510 DISPLAY AT(23,1)SIZE(27):"YOU WIN !" :: Q=Q+M :: GO
    TO 530
520 DISPLAY AT(23,1)SIZE(27):"SORRY, BUT YOU LOSE !" ::
     Q=Q-M
```

cont. on next page

Program Listing—cont.

```
530 FOR DE=1 TO 500 :: NEXT DE :: DISPLAY AT(23,1)SIZE(
    27):" " :: GOTO 310
540 DISPLAY AT(1,1)ERASE ALL:"SORRY FRIEND, BUT YOU LOS
    T","YOUR ENTIRE STAKE!"
550 DISPLAY AT(5,1):"PLAY AGAIN (Y/N):"
560 ACCEPT AT(5,19)SIZE(1)VALIDATE("YN"):CH$ :: IF CH$<
    >"Y" AND CH$<>"N" THEN 560
570 IF CH$="Y" THEN 280
580 DISPLAY AT(7,1):"HOPE YOU HAD FUN!"
590 END
600 FOR J=0 TO 6 :: DISPLAY AT(X+J,Y):CHR$(104);CHR$(10
    4);CHR$(104);CHR$(104);CHR$(104):: NEXT J
610 IF CV=14 THEN CV=1
620 IF CV<4 OR CV>10 THEN 660
630 Z=INT(CV/2):: Z1=INT(X+3-Z/2+.5):: FOR J=1 TO Z ::
    DISPLAY AT(Z1+J-1,Y+1):CHR$(95+SU);CHR$(104);CHR$(9
    5+SU);:: NEXT J
640 IF Z*2<>CV THEN DISPLAY AT(X+3,Y+2):CHR$(95+SU);
650 GOTO 720
660 IF CV>10 THEN 690
670 Z1=INT(X+3-CV/2+.5):: FOR J=1 TO CV :: DISPLAY AT(Z
    1+J-1,Y+2):CHR$(95+SU);:: NEXT J
680 GOTO 720
690 B$="JACK" :: IF CV=12 THEN B$="QUEEN"
700 IF CV=13 THEN B$="KING"
710 Z1=INT(X+3-LEN(B$)/2+.5):: FOR J=1 TO LEN(B$):: DIS
    PLAY AT(Z1+J-1,Y+2):SEG$(B$,J,1);:: NEXT J
720 RETURN
```

9. Big Government

Ten score and eight years ago, our forefathers had a small government. Now you have a BIIIIIIIIIG (that's *really* big!) government. You are the president of a country and must decide how to run the party and keep your constituents (the unsuspecting voters who elected you) happy. You will be in office for 48 months (unless you are impeached).

You can buy and sell bonds for the treasury, spend money (what some politicians seem to do best) to keep the people happy, and solicit contributions (what some politicians do second best). But watch out! Graft, kickbacks, and unforeseen events always occur when the future looks brightest.

You will be brought up-to-date constantly on the latest events by the news teletype (notice the percussive sound of the hammer hitting the paper) and your ever vigilant advisers who interrupt periodically with news flashes.

If you have hopes and dreams for high office, play *Big Government.* But beware, the public is fickle at best and downright hostile at worst!

Game Rules

1. Load and run the Big Government program.
2. The instructions will be displayed.
3. Enter your name (8 characters maximum) when prompted.
4. Each month until the election will bring new plans, conditions, and perils.
5. You can buy and sell bonds at the current market price. This varies from month to month.
6. You should spend approximately $20.00 per voter to keep them in your political camp. Spending more than that results in more people joining your forces, while spending less alienates some voters.
7. You can alienate a small number of voters, but alienating too

many at one time will result in your impeachment and subsequent removal from office.

8. The amount you spend on soliciting contributions can be up to twice the number of bonds you have in the treasury.
9. You need at least one person in your camp for every $10.00 spent on solicitation. After all, you don't want to overwork your supporters.
10. Soliciting contributions is the primary way to make money.
11. The game ends at election time. This is 48 months (or turns) from the beginning of the game, unless you are impeached.
12. Upon completing the game, the statistics for your term in office will be reviewed and the game will end.

Programming Notes

30-60	Instructions
70-90	Get player's name
100-140	Game initialization and screen setup
150-190	Begin printing monthly status
200-250	Crisis—half of the voters leave camp
260-270	Continue printing monthly status
280	Check for end of game and branch if finished
290	Determine price for bonds
300	Print bond price
310-350	Allow player to buy bonds and adjust figures
360-410	Allow player to sell bonds and adjust figures
420-460	Player specifies amount to spend on voters
470-560	Player specifies amount for solicitation
570-690	Adjust monthly figures by random amounts
700-740	Crisis—alienated too many people
750-760	Deficit-spending error message
770-810	Screen status refresh routine
820	Generate random number between 1 and 8
830-870	End-of-game status report
880-950	Branching logic for how well you played the game
960	Excellent job message
970-990	Poor job message
1000-1020	Fair job message

1030-1060	Prompt to exit or play again
1070-1080	Bad job message
1090-1110	Good money raiser message
1130-1140	Pick random News Flash message
1160-1200	News Flash—dishonest brother
1210-1250	News Flash—cabinet resigns
1260-1310	News Flash—economic policies
1320-1360	News Flash—campaign practices
1370-1420	News Flash—Mideast breakthrough
1430-1470	News Flash—inflation down
1480-1530	News Flash—oil prices up
1540-1570	News Flash—polls down
1580-1620	News Flash—campaign contribution
1630	News Flash initial message routine
1640-1700	Teletype printing routine

Program Listing

```
10 REM BIG GOVERNMENT
20 RANDOMIZE
30 CALL CLEAR :: PRINT TAB(7);"BIG GOVERNMENT" :: PRINT
     : :"YOU ARE THE PRESIDENT (WE   WON'T SAY OF WHICH
     NEARBY   COUNTRY) AND YOU HAVE TO"
40 PRINT "MAKE THE EVERYDAY DECISIONS THAT WILL ENSURE
    YOUR        RE-ELECTION (WHICH HAPPENS"
50 PRINT "TO BE 48 MONTHS FROM NOW.)": : :"YOU CAN MAKE
     MONEY FOR THE  PARTY (WHICH IS ONE OF YOUR MANY JO
    BS) BY BUYING AND"
60 PRINT "SELLING BONDS, AND BE       CAREFUL NOT TO AL
    IENATE TOO MANY VOTERS, BECAUSE THEY DOALL OF YOUR
    WORK AND ARE THE"
70 PRINT "KEY TO YOUR SUCCESS.": : :"YOUR NAME? ";
80 ACCEPT AT(24,12)SIZE(8):N$
90 IF N$="" THEN 80
100 CALL CLEAR :: PRINT "TREASURY FUNDS:" :: PRINT "BON
    DS IN BANK:" :: PRINT "FAVORABLE VOTERS:" :: PRINT
    "MONTHS REMAINING:": :
110 PRINT "<--><--><--><--><--><--><-->": : : : : : : :
     : : : : : : : : : :
120 R=8 :: D1=0 :: P1=0 :: Z=48 :: P=100-INT(RND*20)::
    H=INT(RND*3000):: E=INT(RND*391):: S=3000+H-E :: Y=
    3 :: A=INT(H/Y):: I=5 :: Q=1
130 BR=INT(A/P)
140 GOSUB 770 :: GOTO 200
150 P=P+I :: Z=Z-1 :: GOSUB 770 :: ML=5 :: S$="MR. "&N$
    &", OUR OFFICIAL" :: GOSUB 1640 :: S$="SOURCES REPO
    RT THAT LAST" :: GOSUB 1640
```

cont. on next page

Program Listing—cont.

```
160 S$="MONTH "&STR$(D)&" VOTERS WERE" :: GOSUB 1640
170 S$="ALIENATED, BUT "&STR$(I)&" VOTERS" :: GOSUB 164
    0 :: S$="HAVE JOINED OUR SIDE." :: GOSUB 1640
180 IF RND<.2 THEN GOSUB 1120 :: GOSUB 770
190 IF FL THEN ML=2 :: S$="AN ANONYMOUS CONTRIBUTOR" ::
     GOSUB 1640 :: S$="HAS DONATED $"&STR$(FL)&"." :: G
    OSUB 1640 :: FL=0
200 IF RND>.06 THEN 260
210 FL=1
220 P=INT(P/2)+1 :: GOSUB 770
230 ML=6 :: S$="BAD NEWS, MR. "&N$&". A" :: GOSUB 1640
    :: S$="RECENT NEWS POLL SHOWS THAT" :: GOSUB 1640
240 S$="YOU HAVE LOST HALF OF YOUR" :: GOSUB 1640 :: S$
    ="POTENTIAL VOTERS DUE TO A" :: GOSUB 1640
250 S$="SURPRISE ANNOUNCEMENT FROM" :: GOSUB 1640 :: S$
    ="YOUR OPPOSITION." :: GOSUB 1640
260 ML=2 :: S$="YOU RECEIVED $"&STR$(H)&" IN" :: GOSUB
    1640 :: S$="DONATIONS." :: GOSUB 1640
270 IF E<>0 THEN ML=2 :: S$="KICKBACKS PAID OUT" :: GOS
    UB 1640 :: S$="AMOUNTED TO $"&STR$(E)&"." :: GOSUB
    1640
280 IF Z=0 THEN 830
290 C=INT(RND*11):: Y=C+17
300 ML=5 :: S$="BONDS ARE SELLING AT $"&STR$(Y):: GOSUB
     1640 :: S$="" :: GOSUB 1640
310 S$="# OF BONDS TO BUY?" :: GOSUB 1640 :: S$="" :: G
    OSUB 1640 :: FQ=1
320 ACCEPT AT(R-1,20)SIZE(5)VALIDATE(DIGIT):Q$ :: IF Q$
    ="" THEN 320 ELSE Q=VAL(Q$)
330 IF Y*Q>S THEN GOSUB 750 :: GOTO 300
340 A=A+Q :: S=S-Y*Q :: C=0
350 IF Q>0 THEN GOSUB 770
360 S$="# OF BONDS TO SELL?" :: GOSUB 1640 :: FQ=1
370 ACCEPT AT(R-1,21)SIZE(5)VALIDATE(DIGIT):Q$ :: IF Q$
    ="" THEN 370 ELSE Q=VAL(Q$)
380 IF A<Q THEN ML=3 :: S$="MR. "&N$&", THE PARTY" :: G
    OSUB 1640 :: S$="ONLY HAS "&STR$(A)&" BONDS " :: GO
    SUB 1640
390 IF A<Q THEN S$="IN THE BANK" :: GOSUB 1640 :: ML=1
    :: GOTO 360
400 A=A-Q :: S=S+Y*Q :: C=0
410 IF Q>0 THEN GOSUB 770
420 ML=2 :: S$="HOW MUCH SHOULD BE SPENT TO" :: GOSUB 1
    640 :: S$="KEEP VOTERS HAPPY?" :: GOSUB 1640
430 ACCEPT AT(R-1,20)SIZE(6)VALIDATE(DIGIT):Q$ :: IF Q$
    ="" THEN 430 ELSE Q=VAL(Q$)
440 IF Q>S THEN GOSUB 750 :: GOTO 420
450 S=S-Q :: C=1
460 GOSUB 770
470 ML=3 :: S$="HOW MUCH DO YOU WANT TO" :: GOSUB 1640
    :: S$="SPEND ON SOLICITING" :: GOSUB 1640 :: S$="DO
    NATIONS?" :: GOSUB 1640
```

```
480 ACCEPT AT(R-1,12)SIZE(6)VALIDATE(DIGIT):M$ :: IF M$
   ="" THEN 480 ELSE M=VAL(M$):: IF M=0 THEN 570
490 IF M>(2*A)THEN ML=6 :: S$="MR. "&N$&", YOU KNOW WE"
    :: GOSUB 1640 :: S$="NEED MORE BONDS IN THE BANK"
   :: GOSUB 1640
500 IF M>(2*A)THEN S$="TO BACK THAT TYPE OF" :: GOSUB 1
   640 :: S$="SOLICITATION VENTURE." :: GOSUB 1640
510 IF M>(2*A)THEN S$="PLEASE RECONSIDER THE" :: GOSUB
   1640 :: S$="AMOUNT." :: GOSUB 1640 :: GOTO 470
520 IF M>S THEN GOSUB 750 :: GOTO 470
530 IF M<=(10*P)THEN 560
540 ML=4 :: S$="BUT YOU ONLY HAVE "&STR$(P):: GOSUB 164
   0 :: S$="PEOPLE TO DO THE SOLICITING." :: GOSUB 1640
550 S$="EACH PERSON CAN ONLY SPEND" :: GOSUB 1640 :: S$
   ="$10 FOR THE PROJECT." :: GOSUB 1640 :: GOTO 470
560 S=S-M :: GOSUB 770
570 GOSUB 820
580 H=M*C :: E=0 :: IF C>6 THEN FL=C*P :: H=H+FL
590 GOSUB 820
600 IF C<4 THEN E=INT(S/(C+1))
610 S=S-E+H :: GOSUB 770
620 GOSUB 820
630 IF P=0 THEN P=1
640 I=INT((RND*P)/4)+1
650 C=INT(Q/20)
660 D=P-C :: IF D<=0 THEN I=I+INT(ABS(D)/19*RND*11):: D
   =0 :: GOTO 150
670 IF (D*20)>(9*P)THEN 700
680 P1=INT(D1*110/(P*Z))
690 P=C :: D1=D1+D :: I=INT(I/2):: GOTO 150
700 ML=7 :: S$="YOU ALIENATED "&STR$(D):: GOSUB 1640 ::
    S$="PEOPLE IN ONE MONTH!" :: GOSUB 1640
710 S$="DUE TO EXTREME MISMANAGEMENT" :: GOSUB 1640 ::
   S$="YOU HAVE NOT ONLY BEEN" :: GOSUB 1640
720 S$="IMPEACHED AND THROWN OUT OF" :: GOSUB 1640
730 S$="OFFICE, BUT YOU HAVE BEEN" :: GOSUB 1640 :: S$=
   "DECLARED A NATIONAL FINK." :: GOSUB 1640
740 GOTO 1030
750 ML=4 :: S$="MR. "&N$&", WE ONLY HAVE" :: GOSUB 1640
    :: S$="$"&STR$(S)&" IN THE TREASURY." :: GOSUB 164
   0
760 S$="WE CANNOT CONDONE DEFICIT" :: GOSUB 1640 :: S$=
   "SPENDING." :: GOSUB 1640 :: RETURN
770 DISPLAY AT(1,19):S
780 DISPLAY AT(2,19):A
790 DISPLAY AT(3,19):P
800 DISPLAY AT(4,19):Z
810 RETURN
820 C=INT(RND*8)+1 :: RETURN
830 CALL CLEAR
840 PRINT "IN YOUR 48-MONTH TERM, YOU  HAVE OFFENDED MO
   RE THAN" :: PRINT STR$(D1);" PEOPLE."
850 IF P=0 THEN P=1
```

cont. on next page

Program Listing—cont.

```
860 L=INT(A/P)
870 PRINT :"YOU BEGAN WITH A RATIO OF":STR$(BR);" BANK
    BONDS TO EVERY":"VOTER, AND ENDED WITH A":"RATIO OF
     ";STR$(L);" BONDS PER":"VOTER."
880 IF D1>P THEN PRINT : : : : : : : : : : : :: R=13 ::
     ML=5 :: GOTO 710
890 PRINT
900 IF P>110 AND A>1800 AND S>8000 THEN 1090
910 IF L<BR THEN 1070
920 IF P1>10 THEN 970
930 IF L<10 AND D1>300 THEN 970
940 IF P1>3 THEN 1000
950 IF L<BR THEN 1000
960 PRINT "A TRULY REMARKABLE":"PERFORMANCE !! SURELY
    NONE OF YOUR PREDECESSORS COULD  HAVE DONE A MORE N
    OBLE JOB." :: GOTO 1030
970 PRINT "YOUR UNFEELING, AUTOCRATIC  RULE IS REMINISC
    ENT OF       CHICAGO DURING THE AGE"
980 PRINT "OF BOSS TWEED.  THE VOTERS  (THAT ARE STILL
    LEFT) FIND  YOU A VERY UNLIKABLE GUY."
990 PRINT "YOU WILL BE LUCKY TO GET ANYJOB IN THIS COUN
    TRY IN THE  FUTURE." :: GOTO 1030
1000 PRINT "WELL, YOU DIDN'T MAKE THE  HALL OF FAME, B
    UT YOU DIDN'TMAKE MANY ENEMIES EITHER."
1010 PRINT STR$(INT((P*8)*(RND*13+1)/1000));" PEOPLE WO
    ULD LIKE":"TO SEE YOU SENT TO THE MOON,BUT WE ALL H
    AVE LITTLE"
1020 PRINT "PROBLEMS NOW AND THEN."
1030 PRINT : :"PLAY AGAIN? (Y/N)? ";:: ACCEPT SIZE(1):Q
    $
1040 IF Q$="Y" OR Q$="y" THEN RUN
1050 IF Q$="N" OR Q$="n" THEN CALL CLEAR :: END
1060 GOTO 1030
1070 PRINT "POOR JOB ";N$;".":"YOU WERE SUPPOSED TO WOR
    K ATTHE JOB, NOT JUST SIT BACK  AND TAKE THE PAY."
1080 GOTO 1030
1090 PRINT "YOUR PARTY'S BANKBOOK GREW  QUITE A BIT.  Y
    OU'RE  AN    EXCELLENT PRESIDENT, BUT"
1100 PRINT "YOUR DETRACTORS ARE ASKING  FOR A CONGRESSI
    ONAL REVIEW  OF YOUR FINANCES.":"GOOD LUCK !!"
1110 GOTO 1030
1120 REM CAMPAIGN DISASTERS
1130 CD=INT(RND*9)+1
1140 ON CD GOSUB 1160,1210,1260,1320,1370,1430,1480,154
    0,1580
1150 RETURN
1160 ML=7 :: GOSUB 1630 :: S$="YOUR BROTHER WAS FOUND T
    O BE" :: GOSUB 1640 :: S$="IN CAHOOTS WITH A SPY RI
    NG" :: GOSUB 1640
1170 S$="FROM BOTSWANA.  YOU HAVE" :: GOSUB 1640 :: S$=
    "LOST ONE HALF OF YOUR VOTERS" :: GOSUB 1640
1180 S$="AND ONE THIRD OF YOUR BONDS." :: GOSUB 1640
1190 A=INT(A*.66):: P=INT(P/2):: D1=D1+P
```

```
1200 RETURN
1210 ML=8 :: GOSUB 1630 :: S$="TODAY, THE ENTIRE" :: GO
    SUB 1640 :: S$="PRESIDENTIAL CABINET" :: GOSUB 1640
1220 S$="RESIGNED IN A MASSIVE" :: GOSUB 1640 :: S$="DI
    SAGREEMENT WITH YOUR" :: GOSUB 1640
1230 S$="DOMESTIC POLICY.  YOU LOSE" :: GOSUB 1640 :: S
    $="HALF YOUR VOTERS." :: GOSUB 1640
1240 P=INT(P/2):: D1=D1+P
1250 RETURN
1260 ML=9 :: GOSUB 1630 :: S$="THE NEW ECONOMIC POLICIE
    S" :: GOSUB 1640 :: S$="ANNOUNCED TODAY BY THE" ::
    GOSUB 1640
1270 S$="PRESIDENT HAVE PRECIPITATED" :: GOSUB 1640 ::
    S$="A FALLING-OUT BY THE" :: GOSUB 1640
1280 S$="BANKERS, AND THE PRESIDENT" :: GOSUB 1640 :: S
    $="HAS LOST HALF HIS BONDS AS" :: GOSUB 1640
1290 S$="A RESULT OF THE REACTION." :: GOSUB 1640
1300 A=INT(A/2)
1310 RETURN
1320 ML=8 :: GOSUB 1630 :: S$="THE CAMPAIGN PRACTICES O
    F" :: GOSUB 1640 :: S$="MR. "&N$&" CAME UNDER" :: G
    OSUB 1640
1330 S$="CLOSE SCRUTINY TODAY BY THE" :: GOSUB 1640 ::
    S$="NATIONAL BOARD OF REVIEW" :: GOSUB 1640
1340 S$="AND MR. "&N$&"'S PARTY" :: GOSUB 1640 :: S$="H
    AS BEEN FINED $"&STR$(INT(S/3))&"." :: GOSUB 1640
1350 S=S-INT(S/3)
1360 RETURN
1370 ML=9 :: GOSUB 1630 :: S$="MR. "&N$&" HAS" :: GOSUB
     1640 :: S$="SUCCESSFULLY PREVENTED A" :: GOSUB 164
    0
1380 S$="MID-EAST CRISIS BY HIS" :: GOSUB 1640 :: S$="S
    KILLFUL HANDLING OF FOREIGN" :: GOSUB 1640
1390 S$="AFFAIRS.  THIS IS EXPECTED" :: GOSUB 1640 :: S
    $="TO HAVE GOOD EFFECT ON HIS" :: GOSUB 1640
1400 S$="RE-ELECTION DRIVE." :: GOSUB 1640
1410 P=INT(P*1.4)
1420 RETURN
1430 ML=8 :: GOSUB 1630 :: S$="RECENTLY RELEASED REPORT
    S" :: GOSUB 1640 :: S$="SHOW THAT THE RATE OF" :: G
    OSUB 1640
1440 S$="INFLATION HAS SLOWED" :: GOSUB 1640 :: S$="CON
    SIDERABLY DUE TO THIS" :: GOSUB 1640
1450 S$="ADMINISTRATION'S EFFORTS TO" :: GOSUB 1640 ::
    S$="REDUCE GOVERNMENT SPENDING." :: GOSUB 1640
1460 P=INT(P*1.2)
1470 RETURN
1480 ML=9 :: GOSUB 1630 :: S$="THE ARABS HAVE AGAIN RAI
    SED" :: GOSUB 1640 :: S$="OIL PRICES IN RETALIATION
     TO" :: GOSUB 1640
1490 S$="GOVERNMENT CONCESSIONS IN" :: GOSUB 1640 :: S$
    ="THE MID-EAST.  THIS IS" :: GOSUB 1640
```

cont. on next page

Program Listing—cont.

```
1500 S$="EXPECTED TO REFLECT IN" :: GOSUB 1640 :: S$="P
    RESIDENTIAL SUPPORT AT THE" :: GOSUB 1640
1510 S$="POLLS THIS FALL." :: GOSUB 1640
1520 P=INT(P*.8)
1530 RETURN
1540 ML=5 :: GOSUB 1630 :: S$="RECENT PUBLIC OPINION PO
    LLS" :: GOSUB 1640 :: S$="SHOW MR. "&N$&" HAS HAD"
    :: GOSUB 1640
1550 S$="A 10% DROP IN VOTER SUPPORT." :: GOSUB 1640
1560 P=INT(P*.9)
1570 RETURN
1580 ML=9 :: GOSUB 1630 :: S$="CHARLES T. FARNSWORTH,"
    :: GOSUB 1640 :: S$="THE NOTED TEXAS MILLIONAIRE,"
    :: GOSUB 1640
1590 S$="HAS ANNOUNCED HIS SUPPORT" :: GOSUB 1640 :: S$
    ="FOR MR. "&N$&" AND" :: GOSUB 1640
1600 S$="CONTRIBUTED $4500 TO THE" :: GOSUB 1640 :: S$=
    "PARTY AT A MORNING PRESS" :: GOSUB 1640 :: S$="CON
    FERENCE." :: GOSUB 1640
1610 S=S+4500 :: P=P+1
1620 RETURN
1630 S$="********* NEWS FLASH *********" :: GOSUB 1640 ::
     S$="" :: GOSUB 1640 :: RETURN
1640 R=R+1 :: IF R+ML<24 THEN 1660
1650 R=8 :: FOR L=8 TO 24 :: DISPLAY AT(L,1)SIZE(28)::
    NEXT L
1660 FOR L=1 TO LEN(S$)
1670 DISPLAY AT(R,L):SEG$(S$,L,1):: IF SEG$(S$,L,1)<>"
    " THEN CALL SOUND(-1,-2,0)
1680 NEXT L
1690 ML=ML-1 :: IF ML=0 THEN R=R+1 :: DISPLAY AT(R,1)SI
    ZE(28)
1700 RETURN
```

10. Tic-Tac-Toe

Remember sitting for hours when you were a child, playing Tic-Tac-Toe with whomever would take the time to sit and play with you? Well, now you have a partner who is always ready, willing, and quite able (to beat you). This is as good an excuse as any to start your second or third childhood. With the computer as your opponent, you may have met your match, because you will find that the computer plays quite well. The computer even doubles as scorekeeper, keeping track of who wins which games, including those that go to the proverbial "cat." This game simulates how a person thinks. The computer evaluates the current board conditions and makes a logical choice as to which move to make. This is one reason why children enjoy playing this simple game—it encourages them to think logically.

Game Rules

1. Load and run the Tic-Tac-Toe program.
2. Choose X's or O's by pressing the **X** or **O** key.
3. Specify if you want to move first by answering Yes (Y) or No (N) to the prompt.
4. When it is your turn, signify the position you want to move to by pressing the number associated with that position. Positions are numbered a follows:

```
1     2     3
4     5     6
7     8     9
```

5. The object of the game is to place three of your markers in a row—horizontally, vertically, or diagonally.
6. If neither you nor the computer gets three markers in a row, the "cat" wins the game.
7. At the game's completion, you can play again or exit.

Programming Notes

The computer's choices are based on a mathematical analysis of the possible winning positions. This analysis is performed in lines 190 through 510. This game uses only text, but it can be easily modified to use graphics by changing the display routine in lines 620 through 640. These are the only lines that would have to be changed.

10-30	Initialization and screen preparation
40-50	Prompt for you to choose a marker
60	Assignment of X and O to you and the computer
70-80	Prompt for you to choose who goes first
90	Screen instruction setup
100	Screen display if you move first
110-120	Selection of position if computer moves first
130-160	Get your move and see if it's valid
170	Check for a winner
180	Set arrays to determine the winner
190-510	Select computer's move
520	Set computer move, display results, check for winner, and loop back for next move
530-610	Determine array values for judging the winner
620-640	Display playing board
650-710	Determine the winner
720-730	You win, display message
740-750	Computer wins, display message
760-770	"Cat" wins, display message
780-790	Prompt to play again or exit
800-810	End current game

Program Listing

```
10 REM   TIC-TAC-TOE
20 FOR X=1 TO 9 :: B(X)=0 :: NEXT X :: M$(0)="."
30 DISPLAY AT(1,9)ERASE ALL:"TIC-TAC-TOE"
40 DISPLAY AT(4,1):"WOULD YOU LIKE X'S OR O'S:"
50 ACCEPT AT(4,28)SIZE(1)VALIDATE("XO"):CH$ :: IF CH$<>
    "X" AND CH$<>"O" THEN 50
60 M$(9)=CH$ :: M$(1)="X" :: IF CH$="X" THEN M$(1)="O"
70 DISPLAY AT(6,1):"WOULD YOU LIKE TO MOVE","FIRST (Y/N
    ):"
80 ACCEPT AT(7,14)SIZE(1)VALIDATE("YN"):CH$ :: IF CH$<>
    "Y" AND CH$<>"N" THEN 80
```

```
90 DISPLAY AT(1,1)ERASE ALL:"POSITIONS ARE NUMBERED 1",
   "THROUGH 9 FROM TOP-LEFT TO","BOTTOM-RIGHT CORNER."
100 IF CH$="Y" THEN GOSUB 620 :: GOTO 130
110 X=INT(RND*10):: IF X<1 OR X>9 THEN 110
120 B(X)=1 :: GOSUB 620
130 DISPLAY AT(23,1)SIZE(27):"YOUR MOVE (1-9):"
140 ACCEPT AT(23,18)SIZE(1)VALIDATE(DIGIT):X$ :: IF X$=
    "" THEN 140
150 X=VAL(X$):: IF B(X)<>0 THEN CALL SOUND(120,-1,3)::
    GOTO 130
160 B(X)=9
170 GOSUB 650 :: REM SEE IF WINNER
180 GOSUB 530
190 REM
200 REM  SELECT A MOVE
210 REM
220 CF=2
230 IF R(1)<>CF THEN 260
240 FOR X=1 TO 3 :: IF B(X)=0 THEN 520
250 NEXT X
260 IF R(2)<>CF THEN 290
270 FOR X=4 TO 6 :: IF B(X)=0 THEN 520
280 NEXT X
290 IF R(3)<>CF THEN 320
300 FOR X=7 TO 9 :: IF B(X)=0 THEN 520
310 NEXT X
320 IF R(4)<>CF THEN 350
330 FOR X=1 TO 7 STEP 3 :: IF B(X)=0 THEN 520
340 NEXT X
350 IF R(5)<>CF THEN 380
360 FOR X=2 TO 8 STEP 3 :: IF B(X)=0 THEN 520
370 NEXT X
380 IF R(X)<>CF THEN 410
390 FOR X=3 TO 9 STEP 3 :: IF B(X)=0 THEN 520
400 NEXT X
410 IF R(7)<>CF THEN 440
420 FOR X=1 TO 9 STEP 4 :: IF B(X)=0 THEN 520
430 NEXT X
440 IF R(8)<>CF THEN 470
450 FOR X=3 TO 7 STEP 2 :: IF B(X)=0 THEN 520
460 NEXT X
470 IF CF<>18 THEN CF=18 :: GOTO 230
480 Y=0 :: FOR X=1 TO 9 :: IF B(X)=0 THEN Y=Y+1 :: C(Y)
    =X
490 NEXT X
500 X=INT(10*RND):: IF X<1 OR X>Y THEN 500
510 X=C(X)
520 B(X)=1 :: GOSUB 620 :: GOSUB 650 :: GOTO 130
530 R(1)=B(1)+B(2)+B(3)
540 R(2)=B(4)+B(5)+B(6)
550 R(3)=B(7)+B(8)+B(9)
560 R(4)=B(1)+B(4)+B(7)
570 R(5)=B(2)+B(5)+B(8)
```

cont. on next page

Program Listing—cont.

```
580 R(6)=B(3)+B(6)+B(9)
590 R(7)=B(1)+B(5)+B(9)
600 R(8)=B(3)+B(5)+B(7)
610 RETURN
620 J=8 :: FOR X=1 TO 9 STEP 3 :: J=J+2 :: K=9 :: FOR Y
    =0 TO 2 :: K=K+2
630 DISPLAY AT(J,K)SIZE(1):M$(B(X+Y));
640 NEXT Y :: NEXT X :: RETURN
650 FOR X=1 TO 9 :: IF B(X)=0 THEN 670
660 NEXT X :: GOTO 760
670 GOSUB 530
680 FOR X=1 TO 8
690 IF R(X)=27 THEN 720
700 IF R(X)=3 THEN 740
710 NEXT X :: RETURN
720 GOSUB 620
730 DISPLAY AT(20,1):"YOU WON...GOOD GAME!" :: GOTO 780
740 GOSUB 620
750 DISPLAY AT(20,1):"I WON...JUST CAN'T KEEP UP","CAN
    YOU?" :: GOTO 780
760 GOSUB 620
770 DISPLAY AT(20,1):"WELL, CHALK ONE UP FOR THE","CAT.
    .."
780 DISPLAY AT(23,1)SIZE(27):"PLAY AGAIN (Y/N):"
790 ACCEPT AT(23,19)SIZE(1)VALIDATE("YN"):CH$ :: IF CH$
    <>"Y" AND CH$<>"N" THEN 790
800 CALL CLEAR :: IF CH$="N" THEN END
810 RUN
```

11. Qubic

If you think *Tic-Tac-Toe* is too easy, why not try something with a little more "depth?" Qubic is three-dimensional tic-tac-toe. You win by getting three in a row in any direction on any horizontal, vertical, or diagonal plane.

The computer, as your opponent, plays an aggressive game. If you can beat the computer, you are ready for the US Olympic Qubic Team. Tryouts are in Kenosha Falls, Wisconsin, on the 5th Monday in February each year.

Play begins with the traditional toss of the chip. You have a fifty-fifty chance of winning the toss, but the computer does too. The player who wins the toss moves first. From this point on it's every player for himself.

Remember, you or the computer can win with three markers in a row in any direction! There may be more directions than you are first aware of because of the three-dimensional board, so stay alert.

Game Rules

1. Load and run the Qubic program.

2. Select X or O by pressing the **X** or **O** key.

3. The computer will randomly select who goes first.

4. Enter the level and position where you want to place your marker.

5. Levels are numbered 1 through 3, from left to right.

6. Positions are numbered, on each level, as the positions in *Tic-Tac-Toe:*

```
1   2   3
4   5   6
7   8   9
```

7. The object of the game is to place three of your markers in a row, in any direction.

8. If neither the player nor the computer gets three in a row, the "cat" is the winner.

9. At the completion of each game you can play again or exit.

Programming Notes

This game's logic is similar to that used in *Tic-Tac-Toe,* except pointer arrays are used to determine the analytical course the program will follow. There are 49 possible ways to win in *Qubic,* thus the amount of analysis that is completed before the computer chooses a move can be formidable. The computer, however, does it with one equation.

10-90	Program initialization
100-110	You select type of marker
120	Assignment of X and O to you and the computer
130-140	Choose and print who moves first
150	Delay to read messages
160-170	Setup the screen
180	If you go first, display board and get move
190	Computer takes best position, if available
200-220	Get level for move and check validity
230-250	Get position for move and check validity
260-270	Check if position is taken and make move if it is not taken
280	Display new board
290	Check if winner
300-450	Determine the computer's move
460	Display new board
470-490	Routine to display new board
500-580	Routine to check for winner
590	You win, display notice
600	Computer wins, display notice
610	Cat wins, display notice
620-630	Play again or exit
640	End program
650	Restart game
660-700	Routine to accumulate data arrays
710	Create "winning move" string
720-820	Data arrays for possible win positions

Program Listing

```
10 REM   QUBIC
20 DISPLAY AT(1,12)ERASE ALL:"QUBIC" :: DISPLAY AT(3,1)
   :"ONE MOMENT PLEASE..."
30 RANDOMIZE
40 DIM B(3,9),R(49),P(49,3,2),T(27,2)
50 IW=0 :: CW=0 :: WY=0
60 FOR X=1 TO 49 :: FOR Y=1 TO 3 :: READ A :: P(X,Y,1)=
   INT(A/10):: P(X,Y,2)=A-P(X,Y,1)*10 :: NEXT Y :: NEX
   T X
70 M$(0)="." :: M1$="I'M THINKING" :: M2$="CHECKING FOR
    A WINNER"
80 DISPLAY AT(1,12)ERASE ALL:"QUBIC"
90 FOR L=1 TO 3 :: FOR P1=1 TO 9 :: B(L,P1)=0 :: NEXT P
   1 :: NEXT L
100 DISPLAY AT(4,1):"WOULD YOU LIKE X'S OR O'S:"
110 ACCEPT AT(4,28)SIZE(1)VALIDATE("XO"):CH$ :: IF CH$<
    >"X" AND CH$<>"O" THEN 110
120 M$(9)=CH$ :: M$(1)="X" :: IF CH$="X" THEN M$(1)="O"
130 X=RND :: IF X>.5 THEN DISPLAY AT(6,1):"I WIN THE TO
    SS...I GO FIRST!" :: GOTO 150
140 DISPLAY AT(6,1):"YOU WIN THE TOSS...YOU GO","FIRST!
    "
150 FOR DE=1 TO 500 :: NEXT DE
160 DISPLAY AT(1,1)ERASE ALL:"LEVELS ARE 1, 2, AND 3,",
    "LEFT TO RIGHT."
170 DISPLAY AT(4,1):"POSITIONS ARE NUMBERED (ON","EACH
    LEVEL) 1 THROUGH 9","FROM THE TOP-LEFT TO THE","BOT
    TOM-RIGHT CORNER."
180 IF X<=.5 THEN GOSUB 470 :: GOTO 200
190 B(2,5)=1 :: GOSUB 470
200 DISPLAY AT(20,1)SIZE(27):" " :: DISPLAY AT(23,1)SIZ
    E(27):"LEVEL (1-3):"
210 ACCEPT AT(23,14)SIZE(1)VALIDATE("123"):CH$ :: IF CH
    $<"1" OR CH$>"3" THEN 210
220 L=VAL(CH$)
230 DISPLAY AT(23,1)SIZE(27):"POSITION (1-9):"
240 ACCEPT AT(23,17)SIZE(1)VALIDATE(DIGIT):CH$ :: IF CH
    $<"1" OR CH$>"9" THEN 240
250 P1=VAL(CH$)
260 IF B(L,P1)<>0 THEN CALL SOUND(120,-1,3):: GOTO 200
270 B(L,P1)=9
280 GOSUB 470
290 GOSUB 500 :: REM SEE IF WINNER
300 REM
310 REM
320 REM  SEE IF COMPUTER CAN WIN
330 REM
340 IF B(2,5)=0 THEN B(2,5)=1 :: GOTO 460
350 CF=2
360 FOR X=1 TO 49 :: IF R(X)<>CF THEN 400
370 FOR Y=1 TO 3 :: L=P(X,Y,1):: P1=P(X,Y,2)
380 IF B(L,P1)=0 THEN B(L,P1)=1 :: GOTO 460
390 NEXT Y
400 NEXT X
410 IF CF<>18 THEN CF=18 :: GOTO 360
```

cont. on next page

Program Listing—cont.

```
420 C=0 :: FOR L=1 TO 3 :: FOR P1=1 TO 9 :: IF B(L,P1)=
    0 THEN C=C+1 :: T(C,1)=L :: T(C,2)=P1
430 NEXT P1 :: NEXT L
440 X=INT(RND*27):: IF X<1 OR X>C THEN 440
450 B(T(X,1),T(X,2))=1
460 GOSUB 470 :: GOSUB 500 :: GOTO 200
470 FOR L=1 TO 3 :: FOR P1=1 TO 7 STEP 3 :: FOR Y=P1 TO
     P1+2 :: DISPLAY AT(11+INT(P1/3),4+((L-1)*8)+(Y-P1)
    *2)SIZE(1):M$(B(L,Y));
480 NEXT Y :: NEXT P1 :: NEXT L
490 RETURN
500 DISPLAY AT(20,1)SIZE(27):M1$ :: CH$=M1$ :: M1$=M2$
    :: M2$=CH$
510 FOR L=1 TO 3 :: FOR P1=1 TO 9 :: IF B(L,P1)=0 THEN
    530
520 NEXT P1 :: NEXT L :: GOTO 610
530 GOSUB 660
540 FOR X=1 TO 49
550 IF R(X)=27 THEN 590
560 IF R(X)=3 THEN 600
570 NEXT X
580 RETURN
590 GOSUB 710 :: DISPLAY AT(21,1)SIZE(27):"YOU WON...GO
    OD GAME!" :: GOTO 620
600 GOSUB 710 :: DISPLAY AT(21,1)SIZE(27):"I WON...JUST
     CAN'T KEEP UP," :: DISPLAY AT(22,1)SIZE(27):"CAN Y
    OU?" :: GOTO 620
610 DISPLAY AT(20,1)SIZE(27):"WELL, CHALK ANOTHER ONE U
    P" :: DISPLAY AT(21,1)SIZE(27):"FOR THE CAT..."
620 DISPLAY AT(23,1)SIZE(27):"PLAY AGAIN (Y/N):"
630 ACCEPT AT(23,19)SIZE(1)VALIDATE("YN"):CH$ :: IF CH$
    <>"Y" AND CH$<>"N" THEN 630
640 IF CH$="N" THEN DISPLAY AT(5,1)ERASE ALL:"THANKS FO
    R THE GAMES!" :: END
650 RUN
660 FOR X=1 TO 49 :: R(X)=0
670 FOR Y=1 TO 3
680 R(X)=R(X)+B(P(X,Y,1),P(X,Y,2))
690 NEXT Y :: NEXT X
700 RETURN
710 W$="" :: FOR Y=1 TO 3 :: W$=W$&"("&STR$(P(X,Y,1))&"
    -"&STR$(P(X,Y,2))&") " :: NEXT Y :: DISPLAY
AT(20,1)SIZE(27):W$ :: RETURN
720 DATA 11,12,13,14,15,16,17,18,19,11,14,17,12,15,18,1
    3,16,19,11,15,19,13,15,17
730 DATA 21,22,23,24,25,26,27,28,29,21,24,27,22,25,28,2
    3,26,29,21,25,29,23,25,27
740 DATA 31,32,33,34,35,36,37,38,39,31,34,37,32,35,38,3
    3,36,39,31,35,39,33,35,37
750 DATA 11,21,31,12,22,32,13,23,33
760 DATA 14,24,34,15,25,35,16,26,36
770 DATA 17,27,37,18,28,38,19,29,39
780 DATA 11,22,33,14,25,36,17,28,39
790 DATA 13,22,31,16,25,34,19,28,37
800 DATA 11,24,37,12,25,38,13,26,39
810 DATA 17,24,31,18,25,32,19,26,33
820 DATA 11,25,39,13,25,37,17,25,33,19,25,31
```

12. Depth Charge

In *Depth Charge,* you control *Nemesis,* the fastest ship in the Navy, on a search-and-destroy mission in the North Atlantic. There are always two submarines beneath you. You destroy the submarines by hitting them with depth charges. Timing is of the essence and accuracy is a must.

The final score is based on how many submarines you destroy. The deeper and faster the submarine, the more points you accumulate for destroying it. You only have ten passes across the ocean, so don't hesitate.

Game Rules

1. Load and run the Depth Charge program.
2. Once the game begins, press any key to drop depth charges.
3. Up to 3 depth charges can be on the screen at one time.
4. The object of the game is to destroy as many enemy submarines as possible by hitting them with depth charges.
5. As each submarine is destroyed, it is replaced with another one.
6. Bonus passes are awarded if you get 10,000 points in the first 10 passes.
7. The game is over when you have crossed the screen the allotted number of times.
8. When the game is completed you can play again or exit.

Programming Notes

This program uses moving sprites for each object. Because it is possible to have up to six items on the screen at one time (one ship, two submarines, and three depth charges), there are times when the computer is very "busy" keeping track and checking for all possible hits. When this happens it is possible to have a depth charge that comes very close to hitting a submarine, and even appearing to hit it, but the computer does not record it as a hit. This is an infrequent occurrence, and, although it can be frustrating, it is unavoidable given the limitations of the TI/99-4A.

Play the game and study the listing to learn more about using sprites. All the sprites and special characters in this game (and in other programs in *TI-99/4A Games*) were made using the *Sprite/Character Generator* program (see Program 13).

10-130	Program initialization
140-150	Create game screen
160-170	Draw initial ship and pass counter
180	Draw submarines
190	Check for collisions
200	Check for depth charges off screen
210	Check for keypress
220-230	A keypress, so quit or drop depth charge if conditions are correct
240-270	Check if pass is complete and branch if fewer than 10 passes (0-9)
280	Check for bonus passes
290-330	Should another game be played?
340-400	Drop depth charge
410-510	Ship hit handling routine
520-590	Generate new submarine
600-640	See if depth charges have reached the bottom of the screen

Program Listing

```
10 REM DEPTH CHARGE
20 RANDOMIZE
30 CALL CLEAR :: CALL SCREEN(16)
40 CALL CHAR(104,"FFFFFFFFFFFFFFFF")
50 CALL CHAR(96,"0000000014FFFF7E"):: REM SHIP
60 CALL CHAR(97,"0C0828FFFF7E0000"):: REM SUB
70 CALL CHAR(98,"0000000000001818"):: REM DEPTH CHARGE
80 CALL CHAR(112,"33CC000000000000"):: REM SEA LEVEL
90 CALL CHAR(99,"55AA55AA55AA55AA"):: REM BLOW UP
100 CALL COLOR(9,2,1,10,6,1,11,2,6)
110 CALL CLEAR :: PA=0 :: T=0
120 SU=2 :: DC=3
130 FOR J=1 TO SU :: SA(J)=0 :: NEXT J :: FOR K=1 TO DC
    :: DA(J)=0 :: NEXT K
140 FOR J=6 TO 24 :: CALL HCHAR(J,1,104,32):: NEXT J
150 CALL HCHAR(6,1,112,32)
```

```
160 CALL SPRITE(#1,96,2,33,250,0,-2):: CALL DISTANCE(#1
    ,33,1,J):: J1=J
170 DISPLAY AT(1,1):"SCORE: 0";:: DISPLAY AT(1,18):"PAS
    S:";
180 GOSUB 520
190 CALL COINC(ALL,J):: IF J=-1 THEN GOSUB 420
200 GOSUB 600
210 CALL KEY(0,A,S):: IF S=0 THEN 240
220 IF A=81 THEN CALL DELSPRITE(ALL):: DISPLAY AT(8,1)E
    RASE ALL:"FINAL SCORE: ";T :: END
230 GOSUB 340
240 CALL DISTANCE(#1,33,1,J)
250 IF J>J1 THEN PA=PA+1 :: DISPLAY AT(1,23)SIZE(3):PA+
    1 :: IF PA>9 THEN 280
260 J1=J
270 GOTO 180
280 IF T>5000 AND PA<15 THEN CALL COLOR(10,4,1,11,2,4):
    : GOTO 260
290 CALL DELSPRITE(ALL):: DISPLAY AT(23,1):"PLAY AGAIN
    (Y/N): "
300 ACCEPT AT(23,19)VALIDATE("YN")SIZE(1):CH$ :: IF CH$
    <>"Y" AND CH$<>"N" THEN 300
310 IF CH$="Y" THEN 100
320 CALL CLEAR
330 END
340 REM DROP DEPTH CHARGE
350 FOR J=1 TO DC :: IF DA(J)<>0 THEN 390
360 CALL POSITION(#1,L,M):: CALL SPRITE(#1+J,98,2,L+8,M
    ,2,0)
370 CALL SOUND(100,-1,3)
380 DA(J)=1+J :: GOTO 400
390 NEXT J
400 RETURN
410 REM HIT!
420 FOR J=1 TO DC :: IF DA(J)=0 THEN 500
430 FOR K=1 TO SU :: IF SA(K)=0 THEN 490
440 CALL COINC(#DA(J),#SA(K),4,L):: IF L=0 THEN 490
450 CALL PATTERN(#SA(K),99):: CALL DELSPRITE(#DA(J))::
    CALL SOUND(120,-5,3)
460 CALL POSITION(#SA(K),L,M):: T=T+L*5 :: DISPLAY AT(1
    ,7)SIZE(7):T
470 CALL DELSPRITE(#SA(K))
480 DA(J)=0 :: SA(K)=0 :: GOTO 500
490 NEXT K
500 NEXT J
510 RETURN
520 REM PUT OUT NEW SUB?
530 FOR J=1 TO SU :: IF SA(J)<>0 THEN 580
540 SY=RND*200 :: IF SY<50 OR SY>150 THEN 540
550 SX=1
560 CALL SPRITE(#9+J,97,2,SY,SX,0,1)
570 SA(J)=9+J
580 NEXT J
```

cont. on next page

Program Listing—cont.

```
590 RETURN
600 FOR J=1 TO DC
610 IF DA(J)=0 THEN 630
620 CALL POSITION(#DA(J),L,M):: IF L>150 THEN CALL DELS
    PRITE(#DA(J)):: DA(J)=0
630 NEXT J
640 RETURN
```

Section 3

MISCELLANEOUS PROGRAMS

13. Sprite/Character Generator

This utility allows you to quickly and easily create "hex patterns" that can be used for special characters or sprites.

To create a special character or sprite without using a standard ASCII character, you must use the CALL CHAR statement. For example, CALL CHAR("FFFFFFFFFFFFFFFF") creates a solid character on the screen. For more information on creating sprites and characters, you can refer to the TI Extended BASIC manual.

Many programs in *TI-99/4A Games* use special characters and sprites. All of these were created using this program. With a little practice, you should be able to easily create similar characters and sprites.

Program Instructions

1. Load and run the Sprite/Character Generator program.
2. The drawing screen will be displayed. Note the two arrows near the upper-left corner of the screen and the command and code lines near the bottom.
3. The two arrows indicate the current row and column position of the cursor. The intersection to which the arrows are pointing is where the pixel that can be turned on or off is located.
4. When the program is first run, all of the pixels in the character are turned off. Note that the codes at the bottom of the screen are all set to "00."
5. The commands are as follows:

E	Move cursor up. Indicated by the "row arrow" moving up one row. Moving off the top of the pattern causes the cursor to reappear on the bottom of the pattern.
X	Move cursor down. Indicated by the "row arrow" moving down one row. Moving off the bottom of the pattern causes the cursor to reappear on the top of the pattern.

S Move cursor left. Indicated by the "column arrow" moving left one column. Moving off the left side of the pattern causes the cursor to reappear to the right of the pattern.

D Move cursor right. Indicated by the "column arrow" moving right one column. Moving off the right side of the pattern causes the cursor to reappear to the left of the pattern.

Q Quit the program and return to TI Extended BASIC.

B Change the background color for the sprite display field on the right side of the screen. Any color, 1-16, can be selected. The colors are:

1. Transparent
2. Black
3. Medium Green
4. Light Green
5. Dark Blue
6. Light Blue
7. Dark Red
8. Cyan
9. Medium Red
10. Light Red
11. Dark Yellow
12. Light Yellow
13. Dark Green
14. Magenta
15. Gray
16. White

C Change the color of the sprite or character. Any color, 1-16, can be specified. Color designations are the same as those listed under the B command.

W "Wipes" the screen so you can start with a cleared screen.

/ Turn on or off the pixel at the current row and column position.

Programming Notes

10-170	Initialization
180	Check for keypress
190	Check if **Q** was pressed
200	Check if **E** was pressed
210	Check if **X** was pressed
220	Check if **S** was pressed
230	Check if **D** was pressed
240	Adjust row arrow

250	Adjust column arrow
260	Check if **B** was pressed
270	Check if **C** was pressed
280	Check if **W** was pressed
290	Check if **/** was pressed
300	Branch for another keypress
310-350	Change background color
360-400	Change sprite color
410	Adjust column arrow position
420-430	Adjust row arrow position
440	Display entire drawing field
450-460	Display individual pixel location
470-490	Display sprite and code line
500-560	Adjust hex code for character that is being created

Program Listing

```
10 REM SPRITE GENERATOR
20 DISPLAY AT(1,2)ERASE ALL:"CHARACTER/SPRITE GENERATOR
   "
30 DISPLAY AT(3,4):"ONE MOMENT PLEASE..."
40 CALL CHAR(96,"FFFFFFFFFFFFFFFF")
50 CALL CHAR(104,"FFFFFFFFFFFFFFFF")
60 C$="0000000000000000"
70 CALL CHAR(97,C$)
80 CALL CHAR(98,"1010101054381000"):: CALL CHAR(99,"000
   00804FE040800")
90 DIM A(8,8):: FOR X=1 TO 8 :: FOR Y=1 TO 8 :: A(X,Y)=
   0 :: NEXT Y :: NEXT X
100 CX=1 :: CY=1
110 BG=2 :: SC=5
120 CALL COLOR(10,BG,1):: CALL COLOR(9,SC,1):: CALL COL
    OR(#1,SC)
130 GOSUB 440 :: GOSUB 410 :: GOSUB 420
140 DISPLAY AT(3,1)SIZE(27):" "
150 GOSUB 470
160 DISPLAY AT(23,1):"CODES: ";C$
170 DISPLAY AT(18,1)SIZE(27):"COMMANDS: ESDXQBCW/"
180 CALL KEY(0,CH,S):: IF S=0 THEN 180
190 IF CH=81 THEN DISPLAY AT(5,1)ERASE ALL:"BYE!" :: EN
    D
200 IF CH=69 THEN CY=CY-1 :: IF CY<1 THEN CY=8
210 IF CH=88 THEN CY=CY+1 :: IF CY>8 THEN CY=1
220 IF CH=83 THEN CX=CX-1 :: IF CX<1 THEN CX=8
230 IF CH=68 THEN CX=CX+1 :: IF CX>8 THEN CX=1
240 IF CH=69 OR CH=88 THEN GOSUB 420
250 IF CH=83 OR CH=68 THEN GOSUB 410
260 IF CH=66 THEN GOSUB 310:GOTO 300
270 IF CH=67 THEN GOSUB 360 :: GOTO 300
280 IF CH=87 THEN 20
```

```
290 IF CH=47 THEN A(CY,CX)=1-A(CY,CX):: GOSUB 500 :: Y=
    CY :: X=CX :: GOSUB 450
300 GOTO 150
310 DISPLAY AT(18,1)SIZE(27):"BACKGROUND COLOR (1-16):"
320 ACCEPT AT(18,26)VALIDATE(DIGIT)SIZE(2):CH$ :: IF CH
    $="" THEN 320
330 IF VAL(CH$)<1 OR VAL(CH$)>16 THEN 320
340 BG=VAL(CH$):: CALL COLOR(10,BG,1)
350 RETURN
360 DISPLAY AT(18,1)SIZE(27):"SPRITE COLOR (1-16):"
370 ACCEPT AT(18,22)VALIDATE(DIGIT)SIZE(2):CH$ :: IF CH
    $="" THEN 370
380 IF VAL(CH$)<1 OR VAL(CH$)>16 THEN 370
390 SC=VAL(CH$):: CALL COLOR(#1,SC):: CALL COLOR(9,SC,1
    )
400 RETURN
410 DISPLAY AT(7,5)SIZE(10):" ";:: DISPLAY AT(7,5+CX)SI
    ZE(1):CHR$(98);:: RETURN
420 DISPLAY AT(7+OY,5)SIZE(1):" ";
430 DISPLAY AT(7+CY,5)SIZE(1):CHR$(99);:: OY=CY :: RETU
    RN
440 FOR Y=1 TO 8 :: FOR X=1 TO 8 :: GOSUB 450 :: NEXT X
     :: NEXT Y :: RETURN
450 IF A(Y,X)=0 THEN B$=" " ELSE B$=CHR$(96)
460 DISPLAY AT(7+Y,5+X)SIZE(1):B$;:: RETURN
470 DISPLAY AT(10,20):CHR$(104);CHR$(104);:: DISPLAY AT
    (11,20):CHR$(104);CHR$(104);
480 CALL CHAR(97,C$):: CALL SPRITE(#1,97,SC,77,174)
490 RETURN
500 Z=0 :: FOR X=1 TO 8 :: Z=Z+2^(8-X)*A(CY,X):: NEXT X
510 Z1=INT(Z/16):: Z2=Z-(Z1*16)
520 D$=STR$(Z1):: IF Z1>9 THEN D$=CHR$(Z1+55)
530 D1$=D$
540 D$=STR$(Z2):: IF Z2>9 THEN D$=CHR$(Z2+55)
550 D1$=D1$&D$ :: C$=SEG$(C$,1,(CY-1)*2)&D1$&SEG$(C$,CY
    *2+1,16)
560 RETURN
```

14. Master Catalog

This is a program that you may find useful. If you are like many computer-enthusiasts, you have a large collection of tapes. On those tapes are assorted programs and who knows what else. Precisely the point of *Master Catalog.* It will allow you to organize and keep track of program names so you can find them easily. It is a short data-base management program that acts as a filing system for your tapes and programs.

You can search by program name and type, language, record number, date entered (into the system), and tape location (which tape the program is on). You can even sort all the records in your file.

Program Instructions

1. Load and run the Master Catalog program.
2. The first requirement is to choose between loading an old file or creating a new one.
3. The second requirement is entering the date. The format for entering the date is displayed on the screen. Press <ENTER> after each response. The computer checks for a valid date.
4. After you have entered the date, the following menu choices will be displayed:

   1) ENTER
   2) CHANGE
   3) DELETE
   4) SEARCH
   5) LIST
   6) SORT
   7) SAVE & EXIT

5. To do any of the functions listed on the menu, press the number of that function.
6. Each function is self-prompting; follow the directions and answer the questions as they appear on the screen.

7. ENTER will add a record to the system. A record consists of the following information:

   PROGRAM NAME The file's name (35 characters maximum).

   LANGUAGE TYPE Enter the language type (e.g. BASIC, Extended BASIC) for the file, if applicable. 10 characters maximum.

   PROGRAM TYPE Enter the program type (e.g. game, utility) for the file, if applicable. 10 characters maximum.

   DATE ENTERED This is the system date you entered when you first started the program. You do not have to reenter the date for each new entry.

   TAPE LOCATION This is any location that helps you locate the tape at a later time. This field can have a maximum of five characters.

8. CHANGE allows you to change any information that was entered through the ENTER function.

9. DELETE will cause individual records to be deleted from the data base. You will need to enter the file number of the record to be deleted. Then the record is displayed, and you are asked whether you want to delete the record shown.

10. SEARCH will search the file for a specific record or for all records that match a "keyword" that you enter. To search for a match, specify the record to recall or the field to use. The longer the specified keyword, the faster the program will search.

11. LIST displays all records. There is a pause after each screen of information. Press any key to proceed to the next screen.

12. SORT will sort the records in the data base by any specific field. Sorting is done in ascending order.

13. SAVE & EXIT will save your modified data base to cassette and then prompt you to continue with the same file or exit the program.

Programming Notes

As you use the program, note that many of the routines do special functions, such as accepting string input that includes virtually any character. There is also a routine to analyze and reformat entered dates. The sorting routine used is QuickSort. The detailed explanation of these routines is beyond the scope of this book, but information on how they work is in *BASIC Tricks for the TI-99/4A* (Howard W. Sams #22384 for the book only and #26217 for the Combo Pack).

10	REMark statement
20	Define underline character
30-80	Prompt for choice of files
90-160	Date input and analysis routine
170	Program initialization
180	Branch around program lines for loading an old file if choice is to create a new file
190-260	Load a cassette file
270	Error message for loading error
300-340	Print program menu
350-370	Prompt for choice and validate response
380	Branch to appropriate program section
390-510	Routine for inputting a record
520-670	Routine for changing a record
680-750	Routine for deleting a record
760-910	Routine for searching records
920-970	Routine for listing records
980-1150	Routine for sorting records
1160-1220	Save file
1230-1240	Prompt to continue or exit program
1250-1270	File save error routine
1280	Program exit

Program Listing

```
10 REM MASTER CATALOG
20 CALL CHAR(128,"FF00000000000000")
30 CALL CLEAR
40 ON ERROR 30
50 PRINT TAB(10);"WELCOME TO":TAB(8);"MASTER CATALOG":
    : :TAB(7);"1. LOAD OLD FILE":TAB(7);"2. START NEW F
    ILE":TAB(7);"3. EXIT PROGRAM"
    : :
```

```
60 PRINT :TAB(7);"PRESS 1, 2, OR 3": : : : : : :
70 CALL KEY(3,K,S):: IF (K<49)+(K>51)THEN 70
80 IF K=51 THEN 1280
90 CALL CLEAR
100 DISPLAY AT(10,10):"TODAY'S DATE" :: DISPLAY AT(12,1
    4):"/  /  " :: CALL HCHAR(13,14,128,2):: CALL HCHAR
    (13,17,128,2):: CALL HCHAR(13,20,128,2)
110 ACCEPT AT(12,12)SIZE(2)VALIDATE(DIGIT):A$ :: IF A$=
    "" THEN 160 ELSE MM=VAL(A$):: IF MM<1 OR MM>12 THEN
     110
120 ACCEPT AT(12,15)SIZE(2)VALIDATE(DIGIT):A$ :: IF A$=
    "" THEN 120 ELSE DD=VAL(A$)
130 IF DD>31 THEN 120
140 IF (MM=2 AND DD>29)OR((MM=4 OR MM=6 OR MM=9 OR MM=1
    1)AND DD>30)THEN 120
150 ACCEPT AT(12,18)SIZE(2)VALIDATE(DIGIT):A$ :: IF A$=
    "" THEN 150 ELSE YY=VAL(A$):: IF YY<0 OR YY>99 THEN
     150
160 DT$=SEG$("00"&STR$(MM),LEN(STR$(MM))+1,2)&"/"&SEG$(
    "00"&STR$(DD),LEN(STR$(DD))+1,2)&"/"&SEG$("00"&STR$
    (YY),LEN(STR$(YY))+1,2)
170 DIM T$(51,5),ST(50):: N=0
180 IF K=50 THEN 300
190 CALL CLEAR :: PRINT TAB(10);"FILE LOAD": :: ON ERRO
    R 270
200 OPEN #1:"CS1",INTERNAL,INPUT ,FIXED
210 INPUT #1:N
220 FOR L=1 TO N
230 INPUT #1:T$(L,1),T$(L,2),T$(L,3),T$(L,4),T$(L,5)
240 NEXT L
250 CLOSE #1
260 GOTO 300
270 CLOSE #1
280 PRINT : :TAB(7);"FILE LOAD ERROR": :"  PRESS ANY KE
    Y TO CONTINUE": :
290 CALL KEY(3,K,S):: IF S=0 THEN 290 ELSE 30
300 CALL CLEAR
310 ON ERROR 300
320 PRINT "   MASTER PROGRAM CATALOG": : :
330 PRINT TAB(10);"1. ENTER":TAB(10);"2. CHANGE":TAB(10
    );"3. DELETE"
340 PRINT TAB(10);"4. SEARCH":TAB(10);"5. LIST":TAB(10)
    ;"6. SORT":TAB(10);"7. SAVE & EXIT": : :
350 PRINT TAB(8);"PRESS 1 TO 7": : : : :
360 CALL KEY(3,K,S):: IF K<49 OR K>55 THEN 360
370 CALL CLEAR
380 ON K-48 GOTO 390,520,680,760,920,980,1160
390 IF N=50 THEN DISPLAY AT(12,4):"SORRY, THE FILE IS F
    ULL" :: DISPLAY AT(15,9):"PRESS ANY KEY"
400 IF N=50 THEN CALL KEY(3,K,S):: IF S<>1 THEN 400 ELS
    E 300
410 DISPLAY AT(2,11):"PROGRAM #";STR$(N+1):: DISPLAY AT
    (6,1):"NAME:" :: DISPLAY AT(10,1):"LANGUAGE TYPE:"
```

cont. on next page

Program Listing—cont.

```
420 DISPLAY AT(14,1):"PROGRAM TYPE:" :: DISPLAY AT(18,1
    ):"LOCATION:"
430 CALL HCHAR(7,9,128,15):: CALL HCHAR(11,18,128,10)::
     CALL HCHAR(15,17,128,10):: CALL HCHAR(19,13,128,5)
440 ACCEPT AT(6,7)SIZE(15):I$ :: T$(N+1,1)=I$
450 ACCEPT AT(10,16)SIZE(10):I$ :: T$(N+1,2)=I$
460 ACCEPT AT(14,15)SIZE(10):I$ :: T$(N+1,3)=I$
470 T$(N+1,4)=DT$
480 ACCEPT AT(18,11)SIZE(5):I$ :: T$(N+1,5)=I$
490 N=N+1
500 DISPLAY AT(22,6):"ANY MORE ENTRIES?" :: DISPLAY AT(
    23,8):"PRESS Y OR N"
510 CALL KEY(3,K,S):: IF K=89 THEN CALL CLEAR :: GOTO 3
    90 ELSE IF K=78 THEN 300 ELSE 510
520 DISPLAY AT(2,9):"FILE CHANGES" :: DISPLAY AT(5,5):"
    FILE NUMBER (1-";STR$(N);")?"
530 ACCEPT AT(5,25)SIZE(2)VALIDATE(DIGIT):A$ :: IF A$="
    " THEN 530 ELSE CH=VAL(A$):: IF CH<1 OR CH>N THEN 6
    60
540 DISPLAY AT(9,1):"1. NAME: ";T$(CH,1):: DISPLAY AT(1
    1,1):"2. LANGUAGE: ";T$(CH,2):: DISPLAY AT(13,1):"3
    . PGM TYPE: ";T$(CH,3)
550 DISPLAY AT(15,1):"4. DATE ENTERED: ";T$(CH,4):: DIS
    PLAY AT(17,1):"5. LOCATION: ";T$(CH,5)
560 DISPLAY AT(20,7):"NUMBER TO CHANGE?" :: DISPLAY AT(
    21,9):"PRESS 1 TO 5"
570 CALL KEY(3,K,S):: IF K<49 OR K>53 THEN 570 ELSE CH2
    =K-48
580 ON CH2 GOSUB 610,620,630,640,650
590 DISPLAY AT(20,7):"ANY MORE CHANGES?" :: DISPLAY AT(
    21,9):"PRESS Y OR N"
600 CALL KEY(3,K,S):: IF K=89 THEN 560 ELSE IF K=78 THE
    N 300 ELSE 600
610 CALL HCHAR(10,12,128,15):: ACCEPT AT(9,10)SIZE(-15)
    :I$ :: T$(CH,CH2)=I$ :: CALL HCHAR(10,12,32,15):: R
    ETURN
620 CALL HCHAR(12,16,128,10):: ACCEPT AT(11,14)SIZE(-10
    ):I$ :: T$(CH,CH2)=I$ :: CALL HCHAR(12,16,32,10)::
    RETURN
630 CALL HCHAR(14,16,128,10):: ACCEPT AT(13,14)SIZE(-10
    ):I$ :: T$(CH,CH2)=I$ :: CALL HCHAR(14,16,32,10)::
    RETURN
640 CALL HCHAR(16,20,128,8):: ACCEPT AT(15,18)SIZE(-10)
    :I$ :: T$(CH,CH2)=I$ :: CALL HCHAR(16,20,32,8):: RE
    TURN
650 CALL HCHAR(18,16,128,5):: ACCEPT AT(17,14)SIZE(-5):
    I$ :: T$(CH,CH2)=I$ :: CALL HCHAR(18,16,32,5):: RET
    URN
660 DISPLAY AT(12,6):"INVALID FILE NUMBER" :: DISPLAY A
    T(14,9):"PRESS ANY KEY"
670 CALL KEY(3,K,S):: IF S<>1 THEN 670 ELSE 300
680 DISPLAY AT(2,9):"DELETE FILES" :: DISPLAY AT(5,5):"
    FILE NUMBER (1-";STR$(N);")?"
```

```
690 ACCEPT AT(5,25)SIZE(2)VALIDATE(DIGIT):A$ :: IF A$="
    " THEN 690 ELSE CH=VAL(A$):: IF CH<1 OR CH>N THEN 6
    60
700 DISPLAY AT(9,1):"NAME: ";T$(CH,1):: DISPLAY AT(11,1
    ):"LANGUAGE: ";T$(CH,2):: DISPLAY AT(13,1):"PGM TYP
    E: ";T$(CH,3)
710 DISPLAY AT(15,1):"DATE ENTERED: ";T$(CH,4):: DISPLA
    Y AT(17,1):"LOCATION: ";T$(CH,5)
720 DISPLAY AT(20,8):"O.K. TO DELETE" :: DISPLAY AT(21,
    9):"PRESS Y OR N"
730 CALL KEY(3,K,S):: IF K=78 THEN 300 ELSE IF K=89 THE
    N 740 ELSE 730
740 CALL CLEAR :: DISPLAY AT(10,3):"PLEASE WAIT -- DELE
    TING"
750 FOR L=CH TO N :: FOR L2=1 TO 5 :: T$(L,L2)=T$(L+1,L
    2):: NEXT L2 :: NEXT L :: N=N-1 :: GOTO 300
760 DISPLAY AT(2,9):"FILE SEARCH" :: DISPLAY AT(5,2):"S
    EARCH FOR?" :: CALL HCHAR(6,16,128,15)
770 ACCEPT AT(5,14)SIZE(15):S$
780 FOR L=1 TO N
790 CALL CLEAR :: DISPLAY AT(15,2):"SEARCHING FOR: ";S$
800 FOR L2=1 TO 5
810 IF LEN(S$)>LEN(T$(L,L2))THEN 880
820 FOR L3=1 TO (LEN(T$(L,L2))-LEN(S$)+1)
830 IF SEG$(T$(L,L2),L3,LEN(S$))<>S$ THEN 870
840 PRINT "FILE #";STR$(L):"NAME: ";T$(L,1):"LANGUAGE:
    ";T$(L,2):"PGM TYPE: ";T$(L,3):"DATE ENTERED: ";T$(
    L,4):"LOCATION: ";T$(L,5): : :
850 PRINT TAB(6);"CONTINUE SEARCHING?":TAB(9);"PRESS Y
    OR N": :
860 CALL KEY(3,K,S):: IF K=89 THEN 890 ELSE IF K=78 THE
    N 300 ELSE 860
870 NEXT L3
880 NEXT L2
890 NEXT L
900 CALL CLEAR :: DISPLAY AT(20,7):"SEARCHING COMPLETE"
     :: DISPLAY AT(21,9):"PRESS ANY KEY"
910 CALL KEY(3,K,S):: IF S<>1 THEN 910 ELSE 300
920 FOR L=1 TO N
930 PRINT "FILE #";STR$(L):"NAME: ";T$(L,1):"LANGUAGE:
    ";T$(L,2):"PGM TYPE: ";T$(L,3):"DATE ENTERED: ";T$(
    L,4):"LOCATION: ";T$(L,5): :
940 IF (INT(L/3)*3)<>L AND L<>N THEN 970
950 PRINT TAB(9);"PRESS ANY KEY"
960 CALL KEY(3,K,S):: IF S=0 THEN 960 ELSE CALL CLEAR
970 NEXT L :: GOTO 300
980 DISPLAY AT(2,10):"FILE SORT"
990 DISPLAY AT(5,8):"1. NAME":TAB(8);"2. LANGUAGE":TAB(
    8);"3. PGM TYPE":TAB(8);"4. DATE ENTERED":TAB(8);"5
    . LOCATION"
1000 DISPLAY AT(12,8):"SORT BY WHAT #?":TAB(9);"PRESS 1
     TO 5"
```

cont. on next page

Program Listing—cont.

```
1010 CALL KEY(3,K,S):: IF K<49 OR K>53 THEN 1010 ELSE C
    H=K-48
1020 CALL CLEAR :: DISPLAY AT(10,3):"PLEASE WAIT -- SOR
    TING"
1030 P=1 :: Q=N :: T=0
1040 IF P>=Q THEN 1140
1050 V$=T$(P,CH):: I=P :: J=Q+1
1060 J=J-1 :: IF T$(J,CH)>V$ THEN 1060
1070 I=I+1 :: IF T$(I,CH)<V$ AND I<N THEN 1070
1080 IF J<=I THEN 1100
1090 FOR X=1 TO 5 :: W$=T$(I,X):: T$(I,X)=T$(J,X):: T$(
    J,X)=W$ :: NEXT X :: GOTO 1060
1100 FOR X=1 TO 5 :: W$=T$(P,X):: T$(P,X)=T$(J,X):: T$(
    J,X)=W$ :: NEXT X
1110 IF (J-P)<(Q-J)THEN ST(T+1)=J+1 :: ST(T+2)=Q :: Q=J
    -1 :: GOTO 1130
1120 ST(T+1)=P :: ST(T+2)=J-1 :: P=J+1
1130 T=T+2 :: GOTO 1040
1140 IF T<>0 THEN Q=ST(T):: P=ST(T-1):: T=T-2 :: GOTO 1
    040
1150 GOTO 300
1160 PRINT TAB(10);"FILE SAVE": :: ON ERROR 1250
1170 OPEN #1:"CS1",INTERNAL,OUTPUT,FIXED
1180 PRINT #1:N
1190 FOR L=1 TO N
1200 PRINT #1:T$(L,1),T$(L,2),T$(L,3),T$(L,4),T$(L,5)
1210 NEXT L
1220 CLOSE #1
1230 PRINT : :TAB(9);"EXIT PROGRAM":TAB(9);"PRESS Y OR
    N"
1240 CALL KEY(3,K,S):: IF K=89 THEN 1280 ELSE IF K=78 T
    HEN 300 ELSE 1240
1250 CLOSE #1
1260 PRINT : :TAB(7);"FILE SAVE ERROR": :"  PRESS ANY K
    EY TO CONTINUE": :
1270 CALL KEY(3,K,S):: IF S<>1 THEN 1270 ELSE 300
1280 CALL CLEAR :: END
```

SAMS

TI-99/4A™
Games

- Gives TI-99/4A users 11 fun-packed games
- Includes 2 other miscellaneous programs
- Provides helpful loading instructions
- Teaches how to use text, graphics, and sounds in developing your own computer games
- Complete program listings provided for every program

TI-99/4A™ Games is not only fun, it's educational. See for yourself how these programs work and try your hand at adapting them to produce different special effects.

NOTE: Requires TI Extended BASIC Cartridge

Machine Requirements:
- **TI-99/4A Personal Computer**
- **TI Extended BASIC**
- **Monitor or Television Set**

Howard W. Sams & Co., Inc.
4300 West 62nd Street, Indianapolis, Indiana 46268 U.S.A.

$8.95/22398

ISBN: 0-672-22398-8