SAMS

22204

Entertainment Games in TI BASIC and Extended BASIC

Khoa Ton
and Quyen Ton

BLACKSBURG CONTINUING EDUCATION SERIES™
edited by Titus, Titus & Larsen

ENTERTAINMENT GAMES IN TI BASIC AND EXTENDED BASIC

by

Khoa Ton and Quyen Ton

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

FIRST EDITION
SIXTH PRINTING — 1984

International Standard Book Number: 0-672-22204-3
Library of Congress Catalog Card Number: 83-50493

Edited by: *Frank N. Speights*

Printed in the United States of America.

Preface

The Texas Instruments home computer was first introduced to the market as a 16K-byte TI 99/4 console unit. This moderately powerful home computer met some constraints in marketing due to its high price and its keyboard design. The popularity of the TI home computer reached its height only recently when new modifications of the earlier console unit were incorporated into the TI 99/4A, resulting in an improved keyboard that meets the standard typewriter format. With a drastic reduction in price, a great number of TI home computers has been sold; the new console has been warmly received by middle-income American families. However, this boom has created a lack of both documentation and entertainment games for the TI languages.

This book aims to provide TI home computer owners with a cheap way of obtaining arcade-type games and allows hours of enjoyment playing games. Learning TI BASIC and Extended BASIC while keying in the games, enjoying the variety of features and the rich imagination of games for only a fraction of a dollar, increasing your knowledge while playing with the games, and making use of profitable scholastic and business programs (such as *Address Inventory, Typing Skill, Word Search*) are the main purposes of this book.

Some of the programs in this book are written for use with or without the use of a joystick and a speech synthesizer so that you can use a variety of the hardware that is available for the TI 99/4(A) console; other programs need only a console and either a monitor or a tv set in order to run them.

The narrative for each game explains briefly the idea of the game performance, its entertainment and meaning, and is written in

sections such as "Introduction," "Features," and "How To Play, Step-by-Step." The games are self-explanatory; when you run them, instructions will appear first in most games to tell you how to play. When you get used to the procedure, you will not have to come back to the book looking for further explanations.

Now and then among the games, a personal-use program is given to attract the interest of those serious people who want something that is more realistic than imaginative. This type of program (such as *Spelling Test* or *Biorhythm*) is extremely useful for educational purposes but the entertainment value of the program has not been lost.

As a learning process, each program is also explained in plain English, from line to line. This explains how the programmer comes up with the specific features that form the program as a whole while avoiding the use of a flowchart, a concept that requires a higher programming skill.

For the majority of the games, you need only a TI 99/4 or TI 99/4A microcomputer to play with; some other programs will require joysticks, a speech synthesizer, and a disk drive or a cassette tape player. All the programs are pretested and are ready to play. Take care that you don't make any mistakes in typing them in. Please feel free to enhance any program to your own taste or to exterminate any bug you might find, if one exists, but we find it hard to believe that you will discover a serious one. Have fun.

KHOA TON
QUYEN TON

This book is dedicated to our parents, sister, and brother.

Khoa Ton is an eleventh grader at Lowell High School in San Francisco, California. He has used TI BASIC and Extended BASIC since 1978. During the last five years, he has compiled numerous programs in various aspects of software: education, spelling, inventory, and games.

Khoa is also familiar with the languages of other home computers, such as Atari and Hewlett-Packard. He is presently working at writing games in TI Assembly Language in order to improve their speed and animation and, thus, provide games with more excitement.

Quyen Ton is a ninth grader at Lowell High School in San Francisco, California. Besides an award for best academic achievement for a straight A average during a three-year term at Aptos Middle School, he was awarded recognition of being first in school art during his middle school graduation ceremonies. Quyen received the first prize award for obtaining the highest score in MUNCH MAN during a TI festival held in San Francisco during 1983.

Quyen has an extensive knowledge of TI BASIC and Extended BASIC. He has written many game programs since 1979.

Contents

CHAPTER 8

CHAPTER 9

CHAPTER 10

CHAPTER 11

CHAPTER 12

CHAPTER 13

CHAPTER 14

CHAPTER 15

CHAPTER 16

CHAPTER 17

CHAPTER 18

CHAPTER 19

CHAPTER 20

CHAPTER 21

CHAPTER 1

Programming Notes

The games in this book are programmed in TI BASIC and Extended BASIC so they do not need any translation to utilize the powerful features of the computer. All programs are usable with a cassette tape recorder unless otherwise stated. In some programs, accessories are required while, in others, they merely enhance the game and are not a necessity.

TI Extended BASIC Module

In order to program in Extended BASIC, you must have the Extended BASIC language module. Without the module, you can only program games that are written in TI BASIC.

Cassette

To save the games in this book, you'll have to have some type of storage device. Although you can save all programs in this book on tape, some will require a disk system to operate. When using a cassette recorder with a program to store files, be sure to use CS1 and make certain the remote plug is functional.

Disk System

Only one program in this book will absolutely require a disk drive. The *Address Inventory* program will not work with a cassette recorder. You'll only need one disk drive to use the program.

Speech Synthesizer

The Speech Synthesizer is not required in any game but, in a few, they enhance the performance. If you do not have this acces-

sory, you can still type in a game using speech by including the speech sections as they are. You just won't hear any talking. However, this technique will slow program execution tremendously. A better alternative is to leave out all CALL SAY(". . . ") entries. For example, change:

```
2120 CALL SCREEN(12):: CALL SAY("YELLOW"):: GOTO 10
```

to

```
2120 CALL SCREEN(12):: GOTO 10
```

Joysticks

Some programs in this book will require joysticks for maneuvering where keyboard controls are undesirable. Programs that use joysticks will specify their use in FEATURES. When using the TI 99/4A, be sure that the ALPHA LOCK key is in the up (off) position to allow proper operation of the controller unit.

Keyboard

Keyboard input is very common in the programs. When asked a question, type in your answer and press ENTER if you still see the cursor. Keys that are used for controlling movements are E=UP, X=DOWN, S=LEFT, and D=RIGHT. In most games, you will not need to press ENTER after using one of the above keys. Specific usage of the keyboard will be listed with the game.

Memory Expansion

All games will work without use of the memory expansion. For those persons who have the expansion unit, we coded CALL INIT and CALL LOAD(−31878,X) to speed up execution time of Extended BASIC programs. If you don't have the expansion unit, or you have the new Extended BASIC module Version 110, leave out all CALL INIT and CALL LOAD(−31878,X) entries. (X is the number of sprites allowed to move.) For example, change

```
2120 CALL INIT::CALL LOAD(−31878,0)
```

to

```
2120 REM
```

or

```
2120 CALL CLEAR::CALL INIT::CALL LOAD(-31878,20)
     ::CALL SCREEN(10)
```

to

```
2120 CALL CLEAR::CALL SCREEN(10)
```

If you miss any, don't worry, the computer will find them for you at execution time by listing errors.

If you discover that the program is not running properly, delete the REM entries that you have just made. This should correct the problem.

TI 99/4 Impact Printer

You will not need this printer to run any program. Only three programs, *Auto Sprite Definition, Biorhythm,* and *Address Inventory* make use of the printer. In these programs, you can refuse to use the printer if you do not have it by simply not letting the program print to the printer. If you are certain that you'll never get the TI 99/4 Impact Printer, or any other RS-232 compatible printer, you can completely leave out the printer output sections. However, this requires some programming skills. Our printer characteristics may be different from your printer's, so remember to change all OPENs to RS-232 to match your printer's characteristics. If you have the TI 99/4 Impact Printer and have not changed any factory-set switches, you should delete BA=4800 and DA=8 from all OPENs to the printer.

If you do not own a printer that has special graphic functions, you will not be able to print outputs from *Biorhythm* and *Auto Sprite Definition,* as programmed.

Inputting Lines

In most Extended BASIC and some TI BASIC programs, we coded very long lines in order to speed up the execution time and save memory. To input a line longer than the allowed line length, just

type in the line until you hear an "end-of-line" tone (in TI BASIC, close any unclosed quotes), press ENTER, press REDO using SHIFT R on the TI 99/4 or FCTN 8 in the TI 99/4A console (in TI BASIC, type the line number and then SHIFT E or FCTN E), and then continue with the line. You should now have enough space to finish the line.

CHAPTER 2

S * A * M

(BASIC)

Introduction

You are in command of a surface-to-air missile base.

Suddenly out of nowhere, an alien attack force appears in the sky, shooting at your base. The safety of your missile base depends upon your skill and your fast reflexes in shooting down the aliens before they shoot you.

The game uses graphic figures and is written either for TI BASIC or for Extended BASIC without sprites. (If you have Extended BASIC and Memory Expansion, insert line

```
1 CALL INIT::CALL LOAD(-31878,0)
```

to speed up the game.)

Guided missiles are introduced (labeled with the letter "G") and can be used to delay speed in case of early firing. The scoring of shot-down aliens is displayed on the screen and is summed up when all nineteen missiles have been launched.

Triumphant music is played as a reward for those winners who have shot down all nine different kinds of aliens.

A high score requires good coordination as well as good strategy. HAVE FUN!

Features

1. This is a fast action BASIC program (as fast as TI BASIC can go).
2. Sound effects and music are used to enhance game play.
3. The figures are composed of nine different kinds of typical outer-space aliens.

4. "On the spot" scoring is shown on a small screen at the bottom right-hand side of the missile base when an alien is shot down. The final score will appear at the left-hand side when a game is over.
5. For every three missiles, a guided one is available. When using this missile, its speed can be reduced to compensate for early launchings.
6. Aliens are programmed to fire down at the leftmost missile randomly, but only the ones coming out after one alien has escaped can shoot.
7. The higher the flying alien is when shot down, the greater the score will be.
8. To vary the scoring system, an additional feature is added. The scoring value for the second and succeeding alien, of each kind shot down, is doubled.
9. Destroyed aliens will be shown on the screen for the purpose of counting and visualizing the number and kind of aliens downed by the player.

How To Play, Step-By-Step

1. Run the program, type RUN, then press ENTER. The title S.A.M. will appear on the screen along with characters which will be defined as alien figures when you press a key for the next step.
2. Press "N" if you do not need instructions; pressing "Y" will provide basic instructions in self-explanatory notes.
3. A missile base will be shown on the screen with an arrangement of nineteen missiles ready to fire. One of nine different kinds of aliens will randomly appear, flying either from left to right or vice versa into the range of your rightmost missile. When in range, depress the "SPACE BAR" to launch a missile.
4. A regular missile will move twice as fast as all the aliens, while guided missiles will slow down to the same speed as the aliens when the "SPACE BAR" is held down.
5. When all missiles have been fired, the computer will arrange aliens that are on the ground in columns of each kind.
6. If you have shot down all nine different kinds of aliens, a song is played to congratulate you. (This is very hard to achieve.)
7. Press "ENTER" to play again, or press any other key to stop.
8. Helpful hints.
   A. Try to shoot all nine kinds of aliens. 200 points are awarded for this feat.

B. Go all out to get "THE KING." He seldom comes out (he is the topmost alien).
C. USE YOUR GUIDED MISSILES! Shoot them before you would a regular one and then slow them down.
D. Get your guided missiles ready by shooting off unguided missiles at the low-flying aliens. Then wait for the high-flying high-valued aliens.
E. While your missile is in the air, the aliens will not shoot. Use this to your advantage.
F. The alien following right after one that has been shot down will never shoot. If this is a low one, or one that is not important, let it go!

Program Explanation

0010-0040	Program name.
0050-0100	Data for aliens and song.
0100-0210	Opening screen; want instructions?
0220-0320	Describe colors, characters.
0330-0420	Create aliens (up and down facing from data).
0430-0440	Clear screen. If don't want instruction, GOTO 570.
0450-0560	Write instructions.
0570-0590	Create yellow color for screen; clear screen.
0600-0650	Set variables.
0660-0750	Draw missile base; start with alien moving right.
0760-0780	Generate new alien. Out of missiles? If no, GOTO 810.
0790-0800	Erase old alien; jump to end section (1670).
0810-0820	Check to see if next alien may shoot or not. Yes, if AL (number of aliens passed after the last one shot) is larger than two. If not, GOTO 840.
0830	Set number for alien to shoot back.
0840	Determine direction of alien.
0850-0880	Set for alien to move left.
0890-0900	Set for alien to move right.
0910-0950	Determine alien characteristics.
0960-0970	LOOP. Is alien out of screen? If yes, then generate new alien.
0980-0990	Check if SPACE BAR key is pressed; if no, then LOOP (960).
1000-1020	Yes, missile shot. If no more missiles, GOTO 790.
1030	Start missile on its way (FOR—NEXT loop).
1040	Erase old missile.

1050	Missile hits alien? If yes, GOTO BANG! (1420).
1070	If missile is guided (multiple of 3), then next line, else 1100.
1080-1090	Check for guiding (depressed SPACE BAR); if yes, then GOTO 1340.
1100-1110	If missile moved twice, then GOTO MOVE ALIEN (1340).
1120	While missile is in the air, does alien go out of screen? If no, keep moving missile (NEXT ML).
1130-1140	If yes, then erase missile; exit FOR—NEXT loop legally.
1170	Erase missile, then 960 loop.
1180	ALIENS part of program.
1190	Make "alieny" sound.
1200-1210	Check for player firing. If yes, then GOTO 1000.
1220	Is this alien allowed to shoot back, and will it? If no, then GOTO MOVE ALIEN (1340).
1250-1320	Yes! Shoot!
1330-1380	Move alien.
1390	If missile is not in the air, then GOTO 980.
1400	If alien is in the air, then check the coincidence. If alien runs into missile, then GOTO BANG! (1420). If no, return to 1120.
1410-1660	BANG! routine; sound, display score, alien flips and falls, GOTO 1140 (exit missile loop legally).
1670	GAME END section.
1680-1980	Count and display downed aliens. Sounds and graphics.
1990-2000	Did the player get all nine different aliens? If no, GOTO 2210.
2010-2200	Yes! Congratulations.
2210-2340	End message; restart?
2350-2390	Subroutine to print on screen without scrolling.

S * A * M

[BASIC]

```
10 REM ///////
20 REM / SAM /
30 REM ///////
40 REM  BY KHOA TON
50 REM //DATA FOR ALIENS//
60 DATA 1092BA547CC6FE92,423C2418187E1818,7CD67CFEC6AAF
   E00,00183C667E5A42C3,008199FFDBFFBD18,BAFED67C7C4428
   00
70 DATA 60983C243C7E42E7,003C66FF3C182442,00001866FF180
   000
80 REM //DATA FOR SONG//
90 DATA 20,890,21,110,18,220,14,330,15,440,16,550,17,66
   0,31,220,34,330,20,440,22,550
100 DATA 10,660,10,550,60,440,20,330,100,110,1,40000
110 REM //OPENING SCREEN//
120 CALL CLEAR
130 CALL SCREEN(15)
140 PRINT TAB(8);"S * URFACE TO"::TAB(8);"A * IR"::TAB(
    8);"M * ISSILES"::::::::::
150 PRINT "DO YOU REQUIRE INSTRUCTIONS?"
160 FOR S=99 TO 125 STEP 3
170 CALL VCHAR(7,S-94,S)
180 NEXT S
190 CALL SOUND(100,660,9)
200 CALL KEY(0,K,S)
210 IF (K=78)+(K=89)THEN 230 ELSE 200
220 REM //INITIALIZE//
230 CALL COLOR(10,5,1)
240 CALL COLOR(11,13,1)
250 CALL COLOR(12,2,1)
260 CALL COLOR(2,13,1)
270 CALL COLOR(9,7,1)
280 CALL CHAR(42,"FFFFFFFFFFFFFFFF")
290 CALL CHAR(96,"1010102838387CEE")
300 CALL CHAR(125,"163F36488CDBE729")
310 CALL CHAR(101,"00000080CD75FE7E")
320 CALL CHAR(113,"3C24183C24183C24")
330 REM  //CREATE CHARACTERS//
340 FOR I=99 TO 123 STEP 3
350 READ A$
360 CALL CHAR(I,A$)
370 FOR J=15 TO 1 STEP -2
380 C$=C$&SEG$(A$,J,2)
390 NEXT J
400 CALL CHAR(I+1,C$)
410 C$=""
420 NEXT I
430 CALL CLEAR
440 IF K=78 THEN 570
450 REM //INSTRUCTIONS//
460 PRINT TAB(10);"**********":TAB(10);"* S.A.M *":"****
    ******        ***********" ::"  THE PURPOSE OF THIS
    GAME"
```

```
470 PRINT "IS TO UTILIZE YOUR NINETEEN SURFACE-TO-AIR M
    ISSILES TO  SHOOT DOWN ALIEN SABOTEURS.":
480 PRINT "  TO LAUNCH A MISSILE, PRESS""SPACE"". YOUR
    MISSILES WILL":"BE LAUNCHED FROM RIGHT TO":"LEFT."
490 PRINT "  TO COMPLETELY DESTROY THE ALIEN FORCE, YOU
    MUST HIT   EVERY DIFFERENT KIND OF THE INVADERS."
500 PRINT " ALIENS ARE SCORED ACCORDINGTO HEIGHT AND TH
    E SECOND    ALIEN OF EACH KIND HIT WILL COUNT DOUBLE
    IN SCORE."
510 PRINT "USE ""SPACE"" TO GUIDE MISSILE"::" GOOD LUCK
    ...":" PRESS ANY KEY."
520 CALL VCHAR(3,2,42,22)
530 CALL VCHAR(3,31,42,22)
540 CALL HCHAR(24,3,42,29)
550 CALL KEY(0,K,S)
560 IF S=0 THEN 550
570 A$="****                          ***"
580 CALL SCREEN(12)
590 CALL CLEAR
600 REM ///START///
610 K=27
620 T=8
630 FI=0
640 SC=0
650 LA=1
660 REM //BASE//
670 PRINT A$:A$:A$:A$:A$
680 CALL VCHAR(19,31,42,6)
690 PRINT "*****G**G**G**G**G**G**G****"
700 CALL HCHAR(24,3,42,29)
710 FOR I=8 TO 26
720 CALL SOUND(1,110,0,-3,0)
730 CALL VCHAR(22,I,96)
740 NEXT I
750 GOTO 890
760 REM //MAIN CORE//
770 RANDOMIZE
780 IF K>T THEN 810
790 CALL VCHAR(IX,IY,32)
800 GOTO 1690
810 LA=LA+1
820 IF LA<2 THEN 840
830 FI=1
840 IY=INT(RND*2)
850 IF IY THEN 890
860 RY=-1
870 IY=31
880 GOTO 910
890 RY=1
900 IY=2
910 OX=1
920 OY=1
930 IX=INT(RND*17+1)
940 CH=INT(IX/2)*3+99
950 CALL VCHAR(1,32,32,48)
```

```
960 IF (IY>31)+(IY<2)THEN 770
970 GOTO 1190
980 CALL KEY(0,X,S)
990 IF X<>32 THEN 960
1000 CALL SOUND(-200,-5,27)
1010 IF K=T THEN 790
1020 K=K-1
1030 FOR ML=21 TO 1 STEP -1
1040 CALL VCHAR(ML+1,K,32)
1050 IF (IY=K)*(IX=ML)THEN 1420
1060 CALL VCHAR(ML,K,96)
1070 IF INT((K-8)/3)<>(K-8)/3 THEN 1100
1080 CALL KEY(0,X,S)
1090 IF X=32 THEN 1340
1100 IF ML/2<>INT(ML/2)THEN 1150
1110 GOTO 1340
1120 IF (IY>31)+(IY<2)=0 THEN 1150
1130 CALL VCHAR(ML,K,32)
1140 ML=1
1150 NEXT ML
1160 CALL VCHAR(ML+1,K,32)
1170 GOTO 960
1180 REM  ///ALIENS///
1190 CALL SOUND(-900,IX*150,20,-5,25)
1200 CALL KEY(0,X,S)
1210 IF X=32 THEN 1000
1220 IF (FI)*(IY=T)=0 THEN 1340
1230 REM //SHOOT?//
1240 IF INT(RND*2)THEN 1340
1250 REM //YES!!//
1260 CALL SOUND(-100,790,0,-6,15)
1270 CALL VCHAR(IX+1,T,113,21-IX)
1280 CALL VCHAR(22,T,101)
1290 CALL SOUND(800,110,0,-5,0)
1300 CALL VCHAR(IX+1,T,32,21-IX)
1310 FI=0
1320 T=T+1
1330 REM //MOVE ALIEN//
1340 IY=IY+RY
1350 CALL VCHAR(OX,OY,32)
1360 CALL VCHAR(IX,IY,CH)
1370 OX=IX
1380 OY=IY
1390 IF ML=0 THEN 980
1400 IF (IY=K)*(IX=ML)THEN 1420 ELSE 1120
1410 REM //BANG!//
1420 CALL SOUND(300,110,8,-5,0)
1430 CALL VCHAR(ML,K,125)
1440 A$=STR$(((99-CH)/3+10)*5)
1450 X1=20
1460 X=28
1470 GOSUB 2360
1480 CALL SOUND(50,-4,1)
1490 CALL VCHAR(ML,K,CH+1)
1500 CALL SOUND(50,-5,5)
```

```
1510 CALL VCHAR(ML,K,CH)
1520 CALL SOUND(50,-7,12)
1530 CALL VCHAR(ML,K,CH+1)
1540 REM //ALIEN FALL//
1550 FOR I=ML+1 TO 23
1560 CALL SOUND(-150,1000-I*10,30-I)
1570 CALL VCHAR(I-1,K,32)
1580 CALL VCHAR(I,K,CH+1)
1590 NEXT I
1600 FOR I=1 TO 29 STEP 2
1610 CALL SOUND(I*10,110,I+1,-5,I)
1620 NEXT I
1630 IY=1
1640 LA=0
1650 FI=0
1660 GOTO 1140
1670 REM ///GAME END///
1680 REM //COMPUTE SCORE//
1690 FOR I=8 TO 26
1700 CALL GCHAR(23,I,CH)
1710 IF (CH=42)+(CH=71)THEN 1980
1720 CH=CH-1
1730 SC=SC+((99-CH)/3+10)*5
1740 FOR J=9 TO 25 STEP 2
1750 J1=8
1760 CALL GCHAR(J1,J,CH1)
1770 IF CH1=32 THEN 1890
1780 IF CH1<>CH THEN 1970
1790 REM //COUNT ALIENS//
1800 J1=J1+1
1810 CALL GCHAR(J1,J,CH1)
1820 IF CH1=CH THEN 1800
1830 SC=SC+((99-CH)/3+10)*5
1840 CALL SOUND(100,321,0)
1850 CALL SOUND(200,562,0)
1860 CALL SOUND(183,456,0)
1870 CALL SOUND(177,331,0)
1880 CALL SOUND(323,721,0)
1890 CALL VCHAR(J1,J,CH)
1900 CALL SOUND(-200,110,4,-3,0)
1910 CALL VCHAR(23,I,42)
1920 A$=STR$(SC)
1930 X1=20
1940 X=3
1950 GOSUB 2360
1960 J=25
1970 NEXT J
1980 NEXT I
1990 CALL GCHAR(8,25,CH)
2000 IF CH=32 THEN 2220
2010 REM //GOT THEM ALL!//
2020 X1=3
2030 X=9
2040 A$="YOU DID IT!!"
```

```
2050 GOSUB 2360
2060 RESTORE 90
2070 FOR I=1 TO 16
2080 READ X,K
2090 CALL SOUND(90+10*X,K*5,0,110,4,K,10)
2100 CALL SCREEN(I)
2110 NEXT I
2120 X1=X1+1
2130 X=6
2140 A$="YOU HAVE WIPED OUT"
2150 GOSUB 2360
2160 X1=X1+1
2170 X=2
2180 A$="THE WHOLE ALIEN ATTACK FORCE"
2190 GOSUB 2360
2200 SC=SC+200
2210 REM //END MESSAGE//
2220 A$="TO TAKE ON ANOTHER WAVE"
2230 X=5
2240 X1=17
2250 GOSUB 2360
2260 A$="PRESS 'ENTER'"
2270 X=10
2280 X1=18
2290 GOSUB 2360
2300 CALL KEY(0,X,S)
2310 IF X=13 THEN 570
2320 IF S=0 THEN 2300
2330 CALL CLEAR
2340 END
2350 REM //PRINTS SCORE//
2360 FOR A=1 TO LEN(A$)
2370 CALL VCHAR(X1,A+X,ASC(SEG$(A$,A,1)))
2380 NEXT A
2390 RETURN
```

CHAPTER 3

Gold Bag

(BASIC)

Introduction

A gold bag is shown at the end of a line of gold coins. Choosing the right steps to approach the treasure requires an intelligent calculation; this requires a quickness of thought and solid judgment. Each player has a chance to take from 1 to 3 coins, each time it is his turn, in order to reach the gold bag which stores more gold pieces.

You should have "the key" to win the game. Theoretically, if you play with a person who knows the key, you will always lose. Fortunately, the game is programmed to randomly generate more coins so that any situation may be reversed. After a number of plays, you should figure out the key yourself. This is a test of the player's intelligence, so establish a method to approach the key to the treasure—good luck!

Features

1. This is a game programmed for two players. Each player will have a choice of taking from 1 to 3 coins each time it is his turn.
2. A display will show the players' names along with the number of the current round. A flashing indicator will signal each player's turn.
3. The program is a graphic display that exhibits colors and sound for every movement of the player.
4. The scoring system gives different values for each gold bag in order to change the "luck" pattern and, thus, stimulate the player's attention. Both players will have their scores displayed at the top of the screen at all times.
5. The game is intentionally programmed to generate from 1 to 4 coins randomly to change the pattern of winning so that the

first player who discovers the "key" pattern will not win too easily.

6. Winning and losing messages are shown on the screen when the game is over.

How To Play, Step-By-Step

1. Run the program. The computer will ask whether you need instructions or not, with the letter N for no and Y for yes.
2. Input the names of two players (one at a time); the names can have 7 characters maximum. After two names have been typed in, the computer will ask whether you want to change your name. If you type Y, it will return to the beginning of the sequence and ask for the names again.
3. After two names have been input, the game starts by displaying a formation of coins arranged in a line with a gold bag at the end. Initially, 20 coins will appear. On the screen, you will also see the names of the two players along with the amount of gold pieces that they have.
4. One player will be randomly chosen to go first and the indicator will indicate his turn. The first player will then press either key 1, 2, or 3, according to his or her strategy. The second player will do the same when it's his turn.
5. A round is over when one of the players gets the gold bag. At this time, 5 to 15 gold pieces will be added to that player's current amount.
6. The next round will be automatically generated after the preceding one is over. The game is finished after all the initial rounds have been played. The player who gets the most gold pieces is the winner.

Program Explanation

0010-0040	Title.
0060-0070	Clear screen and set to dark blue.
0080-0110	Color setting for characters.
0120-0240	Defining characters.
0270-0300	Print out opening title on screen.
0310-0350	Opening sounds from data on line 1720.
0360	Reset screen color to dark green.
0370-0410	Ask for instructions with a key-press and check

	that "Y" is pressed; if other keys are pressed, then the game will start from line 520.
0420-0500	Instructions.
0510	Clear screen.
0520-0530	Ask for name of player 1 and check to see if the name exceeds 7 characters.
0540-0550	Ask for name of player 2 and check for name length.
0560-0600	Ask if name is to be changed and set the player's score variable.
0610-0620	Ask for number of rounds to be played and check the value if less than 1 or more than 8.
0630-0710	Set up playing screen.
0740	Rounds' count.
0750-0870	Set up gold bag and coins for each round.
0880	Randomly select the first player to move.
0910-0920	Set S1 (turn indicator) to the first player to move.
0930-0950	Check if key "1", "2", or "3" is pressed.
0970-0990	If W=1, then gold bag is taken and add gold bag amount to score on line 980; otherwise, add the amount of coins that have been taken only on line 990.
1000-1020	Place score of player 1 on screen.
1030-1050	Delay; check to see if gold bag has been taken and start new round. Otherwise, it is the next player's turn.
1060-1220	Turn of player 2 is set identical to that of player 1 in lines 910 to 1050.
1230	End of a round.
1240-1380	When all rounds have been played, message will reveal that player 1 wins (1290-1300), player 2 wins (1310-1320), or it is a tie (1330). Lines 1340-1350 ask for a key-press to start a replay.
1390-1450	Subroutine for turn indicator.
1460-1720	Subroutine for: Erase Coin(s) from screen when taken (1470-1550), Random Addition of coins to change pattern (1560-1610), Sound when gold bag is taken (1620-1680), and adding the gold bag random value (1690).
1730	Data for sounds.

GOLD BAG

[BASIC]

```
10 REM ////////////
20 REM / GOLD BAG /
30 REM ////////////
40 REM  BY QUYEN TON
50 REM ///INITIALIZE///
60 CALL CLEAR
70 CALL SCREEN(5)
80 CALL COLOR(9,5,1)
90 CALL COLOR(10,11,5)
100 CALL COLOR(11,13,1)
110 CALL COLOR(12,13,1)
120 A$="FFFFFFFFFFFFFFFF"
130 CALL CHAR(96,A$)
140 CALL CHAR(112,A$)
150 CALL CHAR(120,A$)
160 A$="FEFEFE7C7C381010"
170 CALL CHAR(112,A$)
180 CALL CHAR(120,A$)
190 CALL CHAR(36,"0000E78585B694F5")
200 FOR I=104 TO 108
210 READ A$
220 CALL CHAR(I,A$)
230 NEXT I
240 DATA 3C4299A1A199423C,0000381F0F040719,13C5858381C7
    413F,00E0E4FCF810F80C,C4040282428408F0
250 REM / TITLE /
260 REM
270 PRINT TAB(7);CHR$(105);CHR$(107);" GOLD      ";CHR$(
    105);CHR$(107)
280 PRINT TAB(7);CHR$(106);CHR$(108);"       BAG ";CHR$(
    106);CHR$(108)::::::::::::
290 CALL HCHAR(10,9,104,14)
300 CALL HCHAR(13,9,104,14)
310 RESTORE 1720
320 FOR J=1 TO 4
330 READ D,N
340 CALL SOUND(D,N,0,550,3)
350 NEXT J
360 CALL SCREEN(13)
370 PRINT "DO YOU REQUIRE INSTRUCTIONS?"
380 CALL KEY(0,K,S)
390 IF S=0 THEN 380
400 CALL SOUND(100,440,3)
410 IF K<>89 THEN 520
420 REM //INSTRUCTIONS//
430 CALL CLEAR
440 PRINT TAB(9);CHR$(104);" GOLD BAG ";CHR$(104):::
450 PRINT " THIS IS A TWO PLAYER GAME.":::" THE OBJECT O
    F THIS GAME IS TO GATHER MORE MONEY THAN"
460 PRINT "YOUR OPPONENT.":::" THERE IS A LINE OF GOLD
    COINS LEADING TO A BAG OF   GOLD.":::
```

```
470 PRINT " EACH PERSON IS ALLOWED TO  TAKE ONE TO THRE
    E COINS ON  HIS TURN. THE PLAYER WHO"
480 PRINT "GETS THE GOLD BAG MAY GET UPTO FIFTEEN GOLD
    PIECES!":::::"PRESS ANY KEY...";
490 CALL KEY(0,K,S)
500 IF S=0 THEN 490
510 CALL CLEAR
520 INPUT "PLAYER #1 ?":PL1$
530 IF LEN(PL1$)>7 THEN 520
540 INPUT "PLAYER #2 ?":PL2$
550 IF LEN(PL2$)>7 THEN 540
560 INPUT "CHANGE ANY NAMES ? (Y/N) ":N$
570 PM1=0
580 PM2=0
590 IF N$="Y" THEN 520
600 PRINT ::::
610 INPUT "HOW MANY ROUNDS WOULD YOU   LIKE TO PLAY ?":
    R
620 IF (R>9)+(R<1)THEN 610
630 REM // SCREEN SET-UP //
640 REM
650 PRINT TAB(9);CHR$(104);" GOLD BAG ";CHR$(104):
660 PRINT TAB(5);CHR$(112);TAB(24);CHR$(120)
670 PRINT TAB(2);PL1$;TAB(22);PL2$
680 PRINT TAB(12);"ROUND":::::::::::::::::TAB(5);"PRESS
    : (1)-(2)-(3)":::
690 CALL HCHAR(8,1,96,320)
700 CALL HCHAR(5,4,36)
710 CALL HCHAR(5,23,36)
720 REM // START //
730 REM
740 FOR I=1 TO R
750 REM /DRAW GOLD BAG/
760 CALL VCHAR(12,26,105)
770 CALL VCHAR(12,27,107)
780 CALL VCHAR(13,26,106)
790 CALL VCHAR(13,27,108)
800 W=0
810 X=4
820 REM /DRAW COINS/
830 FOR J=1 TO 20
840 CALL VCHAR(13,4+J,104)
850 CALL SOUND(-99,J*10+400,0)
860 NEXT J
870 CALL HCHAR(6,16,48+I)
880 IF INT(RND*2+1)=2 THEN 1080
890 REM /PLAYER 1'S TURN/
900 REM  / AND SCORING  /
910 S1=11
920 GOSUB 1400
930 CALL KEY(0,K,S)
940 IF S=0 THEN 930
950 IF (K<49)+(K>51)THEN 930
960 GOSUB 1470
970 IF W<>1 THEN 990
```

```
980 PM1=PM1+GB
990 PM1=PM1+(K-48)
1000 FOR K=1 TO LEN(STR$(PM1))
1010 CALL HCHAR(5,5+K,ASC(SEG$(STR$(PM1),K,1)))
1020 NEXT K
1030 FOR D=1 TO 200
1040 NEXT D
1050 IF W=1 THEN 1230
1060 REM /PLAYER 2'S TURN/
1070 REM  / AND SCORING  /
1080 S1=12
1090 GOSUB 1400
1100 CALL KEY(0,K,S)
1110 IF S=0 THEN 1100
1120 IF (K<49)+(K>51)THEN 1100
1130 GOSUB 1470
1140 IF W<>1 THEN 1160
1150 PM2=PM2+GB
1160 PM2=PM2+(K-48)
1170 FOR K=1 TO LEN(STR$(PM2))
1180 CALL HCHAR(5,24+K,ASC(SEG$(STR$(PM2),K,1)))
1190 NEXT K
1200 FOR D=1 TO 200
1210 NEXT D
1220 IF W=1 THEN 1230 ELSE 910
1230 NEXT I
1240 REM /END MESSAGES/
1250 REM
1260 PRINT ::::::::::::::::::::::::
1270 IF PM1<PM2 THEN 1310
1280 IF PM1=PM2 THEN 1330
1290 PRINT PL1$;", YOU'VE WON."::"YOU HAVE";PM1;"GOLD P
     IECES,"::"WHILE ";PL2$;"ONLY HAS";PM2::"GOLD PIECE
     S."
1300 GOTO 1340
1310 PRINT PL2$;", YOU'VE WON."::"YOU HAVE";PM2;"GOLD P
     IECES,"::"WHILE ";PL1$;"ONLY HAS";PM1::"GOLD PIECE
     S."
1320 GOTO 1340
1330 PRINT "BOTH OF YOU ARE TIED AT";PM1;"GOLD PIECES."
1340 PRINT :::"TO PLAY AGAIN PRESS (Y)"
1350 CALL KEY(0,K,S)
1360 IF S=0 THEN 1350
1370 IF K=89 THEN 560
1380 STOP
1390 REM /INDICATING WHO'S  TURN/
1400 FOR J=1 TO 10
1410 CALL COLOR(S1,S1-4,1)
1420 CALL SOUND(-100,S1*54,J)
1430 CALL COLOR(S1,13,1)
1440 NEXT J
1450 RETURN
1460 REM  /TAKE AWAY COINS/
1470 RANDOMIZE
1480 FOR J=1 TO K-48
```

```
1490 X=X+1
1500 CALL GCHAR(13,X,CH)
1510 IF CH<>104 THEN 1620
1520 CALL SOUND(100,400,0)
1530 CALL HCHAR(13,X,96)
1540 NEXT J
1550 IF X>23 THEN 1720
1560 IF INT(RND*5)>0 THEN 1720
1570 Z=INT(RND*4+1)
1580 X=X-Z
1590 CALL HCHAR(13,X+1,104,Z)
1600 CALL SOUND(-100,600,5,-7,0)
1610 GOTO 1720
1620 RESTORE 1730
1630 FOR J=1 TO 4
1640 READ D,N
1650 CALL COLOR(S1,16,1)
1660 CALL SOUND(D-50,N,0,480,3)
1670 CALL COLOR(S1,1,1)
1680 NEXT J
1690 GB=INT(RND*10+5)
1700 CALL HCHAR(12,16,96,48)
1710 W=1
1720 RETURN
1730 DATA 300,392,300,440,300,392,1000,466
```

CHAPTER 4

Arrow Zap

(BASIC)

Introduction

Arrow Zap is a pinball-type game that has different mazes arranged with block walls, diamonds, and figures. The principle of the game is based on a hit, which bounces back at equal incident and reflecting angles. An arrow, in place of a ball, traveling in a predetermined maze will hit and reflect from the objects that it meets on the way. The score will accumulate until the arrow hits the zero number, by chance, in one of the diamonds and is zapped to destruction. A number from 0 to 9 is generated randomly by the computer each time there is a hit by the arrow; this is a process that requires only luck to obtain a high score.

Playing *Arrow Zap* does not require any skill but does need a little thought in choosing the initial path to be followed so that you can outscore a novice.

This is an interesting game designed with sound and music to animate the display screen and test how lucky you are at any moment of the day. The game can be played with either one or two players and a summation of the scores is displayed on the screen. Colors are vivid, sound is abundant, and the game is enjoyable for any member of your family.

Features

1. Abundant colors and graphics are the main attractions of this game. Sound is introduced in every action to animate an exciting mood. The game is composed of 3 different mazes which the player can choose between.
2. Each square block, when hit, will advance one step. Each diamond figure, when hit, will change to an X figure, and vice versa. The diamonds contain a number from 0 to 9 that is ran-

domly generated and the 0's have the power to zap the arrow at any time that it is hit. After playing for a while, the existing maze will change its pattern, thus creating a different variation.

3. A preset winning score appears at bottom of the board at the beginning of each play; winning score mark, game number, and score display are other features of the game.
4. On each turn, the program sets 20 seconds as the time limit for deciding on the path to choose to start. If this time limit runs out, the turn will automatically go to the other player. Finally, a flashing arrow signal indicates the winner at the end of each game.

How To Play, Step-By-Step

1. Run the program. Input the number of players for this game.
2. Next, input the game number (maze) you want to play; choose 1, 2, or 3.
3. When the playing screen is set up, press a number from 1 to 9 for scoring (by hundredths); then, press ENTER. Pressing ENTER causes the bouncing arrow to appear.
4. After the first player is chosen, he uses the LEFT and RIGHT arrow keys (D,S) to move the arrow to the place from which he desires to fire. Then, he presses the UP arrow key (E) to fire. After the arrow has been fired, just sit tight and watch the actions. Player number 2 will use keys J and K to move the arrow left and right and the I key to fire the arrow.
5. The arrow simulates the ball in a pinball machine which bounces around and collects scores for a player. The arrow will bounce and deflect according to the object it hits.
   A. It will make a U-turn when it hits the X's, and that X will change into a diamond with a random number on it.
   B. The arrow will make a 90-degree turn when it hits a diamond. The diamond will then change into an X and the number that was on the diamond will be added to the current score of the player.
   C. The arrow will reverse direction when it hits a block. That block, with any room left, will be pushed one space in that direction.
   D. When the arrow hits any diamond containing a "0," the arrow will be zapped and the turn goes to the next player.
   E. If he desires, each player may invert the playing field at the

start of each of his turns by using the Q/Y keys. This changes the X's to diamonds and vice versa.

6. Game ends when one player reaches the winning score that is set at the beginning of the game. If the player that went first reaches the winning score, the second player has one more chance to fire his arrow and try to accumulate a greater score to win.

Program Explanation

0010-0040	Program name.
0050-0150	Print opening screen.
0160-0710	Redefine colors and characters.
0720-0780	Need instruction?
0790-0850	Instructions.
0860-0940	Ask for number of players and game number.
0950-1100	Draw outline of playing field.
1110-1210	Playfield #1.
1220-1300	Playfield #2.
1310-1350	Playfield #3.
1360-1550	Put in diamonds and X's.
1560-1630	Input winning score.
1640-1680	Set initial values.
1690-1780	Get arrow ready.
1790-2290	Player control portion: Move left and right; invert. Exit portion when UP is pressed.
2300	Main loop.
2310-2410	This loop moves arrow. When hit, branch out and, then, back to here.
2420-2590	Next player up (branch to here every time arrow hits bottom).
2600-2720	Arrow's position after Hit Subroutine.
2730-4180	Hit Subroutines for arrow.
2740-3000	Hit X. Change to random diamond and bounce.
3010-3160	Hit box and bounce.
3170-3370	Hit "0"; arrow destructs.
3380-4120	Hit diamond, bounce, and change to X's.
4130-4180	Subroutines to print score.
4190-4420	Check for winner. If no winner, then go to 1720; else, show winner. If "E" is pressed, go to 640 (restart); if any other key is pressed, then end.

ARROW ZAP
[BASIC]

```
10 REM //////////////
20 REM / ARROW ZAP /
30 REM //////////////
40 REM
50 REM //FIRST SCREEN//
60 CALL CLEAR
70 CALL SCREEN(16)
80 PRINT "       ***ARROW ZAP***        ":::::TAB(9);"012
   3456789":::::::::"I AM MAKING SHAPES":"PLEASE WAIT..
   ."
90 FOR I=3 TO 26 STEP 2
100 K=K+1
110 CALL VCHAR(15,I+3,K*4+92)
120 CALL VCHAR(16,I+3,K*4+93)
130 CALL VCHAR(15,I+4,K*4+94)
140 CALL VCHAR(16,I+4,K*4+95)
150 NEXT I
160 REM ///DEFINE CHAR///
170 REM //NUMBERS//
180 DATA E3C1C9C9C9C9C1E3,E7C7E7E7E7E7E7C3,C3C9F9F1E3CF
    C1C1,C3C9F9C1C1F9C9C3,C9C9C9C1E1F9F9F9,C1C1CFC3E1F9
    C9C3
190 DATA E3C1C9CFC3C9C1E3,C1C1C9F9E3E3C7C7,E3C1C9E3C1C9
    C1E3,E3C9C9C9E1F9F9F9
200 REM //ARROWS//
210 DATA 183C7EE73C3C3C3C,080CFEF7F7FE0C08,3C3C3C3CE77E
    3C18,10307FEFEF7F3010
220 REM //X'S//
230 DATA 60B0F86C3E1B0F06,060F1B3C6CF8B060,060D1F367CD8
    F060,60F0D87C361F0D06
240 FOR I=48 TO 57
250 READ C$
260 CALL CHAR(I,C$)
270 NEXT I
280 RESTORE 180
290 DIM A$(3),PLA(1)
300 A$(0)="03070F1F3F7FFFFF"
310 A$(1)="FFFF7F3F1F0F0703"
320 A$(2)="C0E0F0F8FCFEFFFF"
330 A$(3)="FFFFFEFCF8F0E0C0"
340 REM //DIAMONDS//
350 FOR K=96 TO 132 STEP 4
360 READ C$
370 FOR I=0 TO 3
380 ON I+1 GOTO 390,420,450,480
390 X=10
400 Y=1
410 GOTO 500
420 X=2
430 Y=9
440 GOTO 500
450 X=9
```

```
460 Y=2
470 GOTO 500
480 X=1
490 Y=10
500 B$=A$(I)
510 FOR J=X TO X+6 STEP 2
520 B$=SEG$(B$,1,J-1)&SEG$(C$,Y,1)&SEG$(B$,J+1,16)
530 Y=Y+2
540 NEXT J
550 CALL CHAR(K+I,B$)
560 NEXT I
570 NEXT K
580 FOR I=136 TO 143
590 READ C$
600 CALL CHAR(I,C$)
610 NEXT I
620 CALL CHAR(33,"FF818181818181FF")
630 CALL CHAR(42,"FFFFFFFFFFFFFFFF")
640 CALL SCREEN(15)
650 CALL COLOR(1,7,1)
660 CALL COLOR(9,13,12)
670 CALL COLOR(10,7,12)
680 CALL COLOR(11,5,12)
690 CALL COLOR(12,14,12)
700 CALL COLOR(13,9,12)
710 CALL COLOR(14,6,12)
720 REM //START//
730 CALL SOUND(200,330,5,220,6,330,3,-1,3)
740 CALL SOUND(600,330,3,440,3,330,3,-1,3)
750 CALL SOUND(1,110,0)
760 CALL CLEAR
770 INPUT "NEED INSTRUCTIONS?(Y/N)":B$
780 IF B$<>"Y" THEN 860
790 PRINT "   ***** ARROW ZAP *****":::"  THIS IS A PIN
    BALL-ACTION"
800 PRINT "GAME. THE OBJECT IS TO REACHTHE WINNING SCOR
    E BEFORE    YOUR OPPONENT OR WITH THE"
810 PRINT "LEAST AMOUNT OF ARROWS.":::"KEY STROKES:":::"S
    /J= MOVES ARROW LEFT":"D/K= MOVES ARROW RIGHT":
820 PRINT "E/I= STARTS ARROW MOVING UP":"Q/Y= INVERTS P
    LAYING FIELD"::"AFTER ""E""/""I"" IS PRESSED, NO":"
    KEY IS FUNCTIONAL."
830 PRINT :"PRESSING ""E"" AT THE END":"OF THE GAME WIL
    L END GAME.":::"GOOD LUCK...PRESS ANY KEY."
840 CALL KEY(0,K,S)
850 IF S=0 THEN 840
860 CALL CLEAR
870 CALL CHAR(46,"0000001")
880 CALL COLOR(2,5,12)
890 B$=""
900 INPUT "HOW MANY PLAYERS? (1,2)":NP
910 IF NP>1 THEN 930
920 B$="1 PLAYER: SHOT NUMBER :1"
930 PRINT "WHICH GAME DO YOU WISH TO   PLAY? (1,2,3)";
940 INPUT GA
```

```
950 REM //PLAY FIELD//
960 IF (GA>3)+(GA<1)THEN 760
970 PRINT "     *****ARROW ZAP*****"::::::::::::::::::
    ::"WINNING SCORE: 200 GAME":B$;
980 CALL VCHAR(23,30,GA+48)
990 WSC=200
1000 CALL HCHAR(6,4,46,516)
1010 CALL VCHAR(1,31,32,94)
1020 CALL VCHAR(5,3,33,18)
1030 CALL VCHAR(5,30,33,18)
1040 CALL HCHAR(5,4,33,26)
1050 CALL HCHAR(22,4,42,26)
1060 CALL HCHAR(7,4,33,4)
1070 CALL HCHAR(7,26,33,4)
1080 CALL VCHAR(6,7,33)
1090 CALL VCHAR(6,26,33)
1100 ON GA GOTO 1110,1220,1310
1110 RESTORE 1370
1120 J=0
1130 FOR I=0 TO 4
1140 CALL HCHAR(19,4,33,8)
1150 CALL HCHAR(19,22,33,8)
1160 J=J+1
1170 CALL VCHAR(J+10,I+10,33)
1180 CALL VCHAR(J+10,23-I,33)
1190 NEXT I
1200 CALL VCHAR(16,16,33)
1210 GOTO 1430
1220 RESTORE 1400
1230 CALL HCHAR(19,4,33,7)
1240 CALL HCHAR(19,23,33,7)
1250 CALL HCHAR(13,4,33,5)
1260 CALL HCHAR(14,24,33,6)
1270 CALL HCHAR(12,13,33)
1280 CALL HCHAR(12,21,33)
1290 CALL HCHAR(15,16,33)
1300 GOTO 1430
1310 RESTORE 1420
1320 CALL HCHAR(17,11,33,12)
1330 CALL VCHAR(12,11,33,5)
1340 CALL VCHAR(12,22,33,5)
1350 GOTO 1430
1360 REM //FIELD #1//
1370 DATA 10,6,15,5,13,7,16,11,7,12,10,13,17,14,8,16,10
     ,19
1380 DATA 12,16,17,18,7,20,10,26,16,21,13,25,15,27,-1,0
1390 REM //FIELD #2//
1400 DATA 7,13,7,20,8,16,10,6,10,27,11,10,11,23,12,17,1
     5,9,16,5,16,23,15,27,17,12,17,19,-1,0
1410 REM //FIELD #3//
1420 DATA 7,12,7,20,8,17,10,8,10,13,10,25,11,5,11,19,13
     ,14,14,18,14,26,17,6,18,24,17,27,-1,0
1430 READ X,Y
1440 IF X=-1 THEN 1540
```

```
1450 RANDOMIZE
1460 K=(INT(RND*19+1)*4)+92
1470 IF K<135 THEN 1490
1480 K=140
1490 CALL VCHAR(X,Y,K)
1500 CALL VCHAR(X+1,Y,K+1)
1510 CALL VCHAR(X,Y+1,K+2)
1520 CALL VCHAR(X+1,Y+1,K+3)
1530 GOTO 1430
1540 CALL HCHAR(6,4,48,3)
1550 CALL HCHAR(6,27,48,3)
1560 CALL KEY(0,K,S)
1570 IF K=13 THEN 1620
1580 IF (K<48)+(K>57)THEN 1540
1590 WSC=(K-48)*100
1600 CALL VCHAR(23,18,K)
1610 GOTO 1540
1620 CALL SOUND(100,110,0)
1630 CALL VCHAR(23,30,ASC(STR$(GA)))
1640 REM   //START//
1650 PLA(0)=0
1660 PLA(1)=0
1670 SH=1
1680 PL=6
1690 CALL SOUND(999,800,9)
1700 CALL HCHAR(4,1,32,32)
1710 IF PL=6 THEN 4200
1720 CALL VCHAR(4,PL-1,138)
1730 T=21
1740 PX=5
1750 PY=17
1760 X=21
1770 Y=16
1780 XDIR=0
1790 REM //PLAYER'S MOVE//
1800 CALL VCHAR(X,Y,136)
1810 CALL SOUND(100,330,6,490,7,330,7,-8,5)
1820 CALL KEY(SGN(PL-6)+1,K,ST)
1830 IF K+1 THEN 1850
1840 CALL JOYST(SGN(PL-6)+1,K,ST)
1850 T=T-.25
1860 IF T>INT(T)THEN 1890
1870 SC$="!!"&STR$(T)
1880 GOSUB 4140
1890 IF T<0 THEN 2430
1900 IF S=0 THEN 1820
1910 IF (K-5)*(ST-4)THEN 2000
1920 FOR I=500 TO 1000 STEP 100
1930 CALL SOUND(-100,I,I/100*2)
1940 NEXT I
1950 CALL HCHAR(5,15,33,3)
1960 XDIR=-1
1970 YDIR=0
1980 AC=136
```

```
1990 GOTO 2310
2000 IF (K-2)*(K+4)THEN 2050
2010 CALL VCHAR(X,Y,46)
2020 IF Y<5 THEN 1800
2030 Y=Y-1
2040 GOTO 1800
2050 IF (K-3)*(K-4)THEN 2110
2060 CALL VCHAR(X,Y,46)
2070 IF Y>28 THEN 1800
2080 Y=Y+1
2090 GOTO 1800
2100 REM  //INVERT//
2110 IF (K-18)+(XDIR=9)THEN 1820
2120 CALL SOUND(99,990,0)
2130 XDIR=9
2140 ON GA GOTO 2150,2170,2190
2150 RESTORE 1370
2160 GOTO 2200
2170 RESTORE 1400
2180 GOTO 2200
2190 RESTORE 1420
2200 READ SX,SY
2210 IF SX=-1 THEN 1820
2220 CALL GCHAR(SX,SY,K)
2230 IF K=140 THEN 2270
2240 CALL SOUND(99,770,4)
2250 GOSUB 4040
2260 GOTO 2200
2270 CALL SOUND(99,440,4)
2280 GOSUB 2950
2290 GOTO 2200
2300 REM  //MAIN LOOP//
2310 CALL GCHAR(X+XDIR,Y+YDIR,CH)
2320 CALL VCHAR(X,Y,46)
2330 IF CH=46 THEN 2380
2340 IF (CH>95)*(CH<100)THEN 3180
2350 IF CH=42 THEN 3340
2360 GOSUB 2740
2370 GOTO 2310
2380 X=X+XDIR
2390 Y=Y+YDIR
2400 CALL VCHAR(X,Y,AC)
2410 GOTO 2310
2420 REM //WHO'S UP?//
2430 CALL VCHAR(X,Y,46)
2440 CALL VCHAR(21,16,136)
2450 IF NP=1 THEN 2470
2460 IF PL=6 THEN 2550 ELSE 2520
2470 PX=24
2480 PY=26
2490 SH=SH+1
2500 SC$=" :"&STR$(SH)
2510 GOSUB 4140
2520 PL=6
```

```
2530 K=5
2540 GOTO 2570
2550 PL=29
2560 K=13
2570 CALL VCHAR(21,16,136)
2580 CALL COLOR(2,K,12)
2590 GOTO 1690
2600 REM   //DIRECTION SUBROUTINES//
2610 SX=X+XDIR
2620 SY=Y+YDIR
2630 RETURN
2640 SX=X+XDIR-1
2650 SY=Y+YDIR
2660 RETURN
2670 SX=X+XDIR
2680 SY=Y+YDIR-1
2690 RETURN
2700 SX=X+XDIR-1
2710 SY=Y+YDIR-1
2720 RETURN
2730 REM //HITS//
2740 IF CH=33 THEN 3020
2750 IF CH<136 THEN 3390
2760 REM //HIT X//
2770 CALL SOUND(300,440,15,330,15,550,15)
2780 ON CH-139 GOSUB 2610,2640,2670,2700
2790 IF XDIR THEN 2870
2800 YDIR=-YDIR
2810 AC=138-YDIR
2820 IF CH/2=INT(CH/2)THEN 2850
2830 X=X-1
2840 GOTO 2930
2850 X=X+1
2860 GOTO 2930
2870 XDIR=-XDIR
2880 AC=137+XDIR
2890 IF CH>141 THEN 2920
2900 Y=Y+1
2910 GOTO 2930
2920 Y=Y-1
2930 RANDOMIZE
2940 CALL VCHAR(X,Y,AC)
2950 K=INT(RND*10)*4+96
2960 CALL VCHAR(SX,SY,K)
2970 CALL VCHAR(SX+1,SY,K+1)
2980 CALL VCHAR(SX,SY+1,K+2)
2990 CALL VCHAR(SX+1,SY+1,K+3)
3000 RETURN
3010 REM //HIT BOX//
3020 K=2*XDIR+X
3030 S=2*YDIR+Y
3040 CALL SOUND(100,220,11,110,10,130,19,-3,10)
3050 CALL GCHAR(K,S,CH)
3060 IF CH<>46 THEN 3090
3070 CALL VCHAR(X+XDIR,Y+YDIR,46)
```

```
3080 CALL VCHAR(K,S,33)
3090 IF XDIR=0 THEN 3130
3100 XDIR=-XDIR
3110 AC=137+XDIR
3120 GOTO 3150
3130 YDIR=-YDIR
3140 AC=138-YDIR
3150 CALL VCHAR(X,Y,AC)
3160 RETURN
3170 REM //HIT 0!//
3180 CALL VCHAR(X,Y,46)
3190 ON CH-95 GOSUB 2610,2640,2670,2700
3200 CALL SOUND(300,110,15,110,15,220,13,-8,0)
3210 CALL VCHAR(SX,SY,140)
3220 CALL VCHAR(SX+1,SY,141)
3230 CALL VCHAR(SX,SY+1,142)
3240 CALL VCHAR(SX+1,SY+1,143)
3250 CALL SOUND(600,110,9,110,8,440,6,-8,0)
3260 CALL COLOR(14,2,12)
3270 FOR I=X TO 21 STEP 1
3280 CALL GCHAR(I,Y,CH)
3290 CALL VCHAR(I,Y,INT(RND*4+136))
3300 CALL SOUND(300,500-I*10,I)
3310 CALL VCHAR(I,Y,CH)
3320 NEXT I
3330 CALL COLOR(14,6,12)
3340 FOR I=450 TO 150 STEP -50
3350 CALL SOUND(-100,I,4,220,0)
3360 NEXT I
3370 GOTO 2430
3380 REM //HIT DIAMONDS//
3390 P=INT(CH/4)*4
3400 CALL SOUND(200,220,5,330,3)
3410 ON CH-P+1 GOTO 3420,3570,3720,3870
3420 SX=X+XDIR
3430 SY=Y+YDIR
3440 IF XDIR<>1 THEN 3510
3450 XDIR=0
3460 YDIR=-1
3470 X=X+1
3480 Y=Y-1
3490 AC=139
3500 GOTO 4010
3510 XDIR=-1
3520 YDIR=0
3530 X=X-1
3540 Y=Y+1
3550 AC=136
3560 GOTO 4010
3570 SX=X+XDIR-1
3580 SY=Y+YDIR
3590 IF XDIR<>-1 THEN 3660
3600 XDIR=0
3610 YDIR=-1
3620 X=X-1
```

```
3630 Y=Y-1
3640 AC=139
3650 GOTO 4010
3660 YDIR=0
3670 XDIR=1
3680 X=X+1
3690 Y=Y+1
3700 AC=138
3710 GOTO 4010
3720 SX=X+XDIR
3730 SY=Y+YDIR-1
3740 IF XDIR<>1 THEN 3810
3750 XDIR=0
3760 YDIR=1
3770 X=X+1
3780 Y=Y+1
3790 AC=137
3800 GOTO 4010
3810 XDIR=-1
3820 YDIR=0
3830 X=X-1
3840 Y=Y-1
3850 AC=136
3860 GOTO 4010
3870 SX=X+XDIR-1
3880 SY=Y+YDIR-1
3890 IF XDIR<>-1 THEN 3960
3900 XDIR=0
3910 YDIR=1
3920 X=X-1
3930 Y=Y+1
3940 AC=137
3950 GOTO 4010
3960 XDIR=1
3970 YDIR=0
3980 X=X+1
3990 Y=Y-1
4000 AC=138
4010 PLA(SGN(PL-6))=PLA(SGN(PL-6))+P/4-24
4020 CALL SOUND(300,330,4,330,3)
4030 CALL VCHAR(X,Y,AC)
4040 CALL VCHAR(SX,SY,140)
4050 CALL VCHAR(SX+1,SY,141)
4060 CALL VCHAR(SX,SY+1,142)
4070 CALL VCHAR(SX+1,SY+1,143)
4080 IF X=21 THEN 4180
4090 CALL SOUND(-200,550,4,220,3)
4100 SC$="00"&STR$(PLA(SGN(PL-6)))
4110 PX=6
4120 PY=PL
4130 REM //PRINTS DIGITS//
4140 S=LEN(SC$)
4150 CALL VCHAR(PX,PY,ASC(SEG$(SC$,S,1)))
4160 CALL VCHAR(PX,PY-1,ASC(SEG$(SC$,S-1,1)))
4170 CALL VCHAR(PX,PY-2,ASC(SEG$(SC$,S-2,1)))
```

```
4180 RETURN
4190 REM  //WINNER?//
4200 IF (PLA(0)<WSC)*(PLA(1)<WSC)THEN 1720
4210 IF PLA(0)=PLA(1)THEN 1720
4220 IF PLA(1)>PLA(0)THEN 4290
4230 IF NP>1 THEN 4270
4240 SH=SH-1
4250 SC$=" :"&STR$(SH)
4260 GOSUB 4140
4270 PL=6
4280 GOTO 4300
4290 PL=29
4300 CALL VCHAR(6,PL-3,137)
4310 CALL VCHAR(6,PL+1,139)
4320 CALL HCHAR(5,PL-2,138,3)
4330 CALL HCHAR(7,PL-2,136,3)
4340 CALL COLOR(14,7,12)
4350 CALL KEY(0,K,S)
4360 IF K=69 THEN 4420
4370 IF K<>-1 THEN 640
4380 CALL SOUND(200,550,0,-8,0)
4390 CALL COLOR(14,16,12)
4400 CALL SOUND(200,220,0,-8,0)
4410 GOTO 4340
4420 CALL CLEAR
```

CHAPTER 5

Cosmic Guns

(BASIC)

Introduction

The program has three separate stages. The first stage consists of a space base armed with five powerful Cosmic Guns that fire randomly to form a curtain of deadly rays to prevent enemy penetration of their base. The second stage is a horizontal shaft equipped with fast moving laser beams that are the last defense to a fully computerized headquarters (the third stage) in which is hidden a secret code for a self-destruction procedure.

You are flying a spacecraft sent from earth and your mission is to penetrate the Cosmic Gunners' tight defense of both the first and second stage in order to reach their headquarters, discover, and figure out the self-destruction secret code, so that you can key it in to ruin completely their base.

Your craft has a built-in warping effect that allows you to move two spaces instead of one. This can be used to avoid the firing of the Cosmic Gunners whenever you feel it necessary, but be careful, as this move will diminish your fuel reserved for the final purpose.

It's exciting! Try it.

Features

1. This is a full graphic, sound, music, and vivid color program that is written in BASIC.
2. The program is designed to have three stages of play: crossing the Cosmic Gunners' region, penetrating the deadly horizontal

laser-defense shaft, and decoding the secret self-destruction code.

3. There are interesting starting and winning animation graphics.
4. Fuel for the spacecraft will be decreased or increased according to conditions met with fuel status checking device.
5. A tunnel that penetrates into enemy headquarters will appear randomly above or below the horizontal shaft to allow for a little luck besides your skill.
6. The program is devised to use the directional keys (E, D, X, and S) to move.
7. A warping effect is a useful tool designed in your craft that allows you to avoid the firing. It also allows the craft to hop two spaces when necessary.

How To Play, Step-By-Step

1. Run the program. A group of letters will appear which are then followed with an animation sequence that reveals the title. If you do not want to see the animation sequence, depress any key before the letters appear.
2. Press Y for instructions or N for none.
3. Use keys S and D to move your craft left or right for the first stage. Your craft needs to cross the screen three times without being destroyed in order to reach the horizontal shaft, the second stage. As you reach the left side of the screen, your fuel will be regenerated with an amount that is enough to go across the second time. If you are hit by the cosmic guns, your fuel will be decreased by 100 units each time you are hit.
4. By pressing the A key, for warp, your craft will hop two spaces. This is used to avoid the firing guns, but it will cost you 30 fuel units.
5. By pressing the space bar at any time, you can check your fuel status and move forward one step.
6. When you reach the tunnel entrance, you are about to enter stage 2. Stage 2 consists of three horizontal laser guns that are aiming and firing constantly at your craft. Fast reflexes are required. Using the four arrow keys (E, D, X, and S) allows you to move up, down, and to advance your craft so as to reach the base of the guns, which leads to their headquarters. Oh yes, don't try to back out of this tunnel.

HINT: These guns may seem very fast and difficult to elude, but they will shoot a bolt only half way to your ship at first. At this time, you have a chance to press E or X and move your ship up or down and out of the way. After this split second, that same gun will complete firing its laser bolt the length of the tunnel.

A trick to use in this stage is to keep holding down the S key which moves you forward and also slows down the gun's reaction time. Thus it will be easier for you to see the warning bolt and react to it.

7. Upon reaching the gun base, a tunnel will appear either at the top or the bottom of the shaft (by random). You have to enter the tunnel leading to the Cosmic Gunners' headquarters to play the third stage.
8. Playing the third stage requires a good memory. By pressing the number keys and remembering exactly where their positions are, you can key in the secret self-destruction code correctly from left to right. You can only make 8 errors in putting the code in order. After you have keyed in the correct code, just wait and you will see an animated destruction of the headquarters.
9. Game ends when you have either succeeded or failed.

Program Explanation

0010-0040	Program name.
0050-0330	Print out the title screen on a black background and define the colors and characters.
0340-0370	If any key is pressed during this time, skip the title animation.
0380-0490	Shoot down the letters (title animation).
0500-0540	GOSUB to play Cosmic Guns theme.
0550-0650	Instructions.
0660-0770	Set up stage I.
0790-1020	Loop for stage I.
1030-1110	Music subroutine.
1120-1200	Stars subroutine.
1210-1270	Subroutine to print guns for stage I.
1280-1510	Subroutine to shoot guns for stage I, check for hits, adjust variables, and branch accordingly.

1520-1770	Keyboard input subroutine.
1570	Moves left.
1600	Moves right.
1610	Warps.
1710	Fuel check.
1780-1860	Explosion subroutine.
1870-2000	Out of fuel.
2010-2150	End message.
2160-2210	Subroutine to print without scrolling.
2220-2520	Stage II; set variables and draw stage II play screen.
2530-2690	Movements for stage II subroutine.
2700-2760	Hit wall, go to end messages.
2770-2810	Did not hit wall, so draw ship.
2820-2940	Loop for stage II, randomly fire guns, check for hit on ship.
2950-3000	Display upper secret passage.
3010-3050	Display lower secret passage.
3060-3150	Stage III and instructions.
3160-3240	Get secret code.
3250-3260	Display stage III screen.
3270-3350	Player input.
3360-3520	Wrong code; if "tries" expires, go to end message.
3530	MADE IT!
3540-4080	Animation of escaping Cosmic Gunners' base.
4090-4110	Congratulations message.

COSMIC GUNS
[BASIC]

```
10 REM ///////////////
20 REM / COSMIC GUNS /
30 REM ///////////////
40 REM  BY KHOA TON
50 REM ///INITIALIZE///
60 CALL CLEAR
70 CALL SCREEN(2)
80 PRINT TAB(7);"ASPACEGAMEYOU'";TAB(7);"LACOSMICEGUNSU
   ";TAB(7);"LLLOVETOPLAY!!";::::::::::;
90 FOR J=1 TO 8
100 CALL COLOR(J,16,1)
110 NEXT J
120 CALL COLOR(9,11,1)
130 CALL COLOR(14,9,1)
140 CALL COLOR(15,8,1)
```

```
150 CALL COLOR(10,16,1)
160 CALL COLOR(11,16,1)
170 CALL COLOR(12,10,1)
180 CALL COLOR(13,15,1)
190 CALL CHAR(130,"FFFFFFFFFFFFFFFF")
200 CALL CHAR(98,"000054387C3854")
210 CALL CHAR(97,"00925400C6005492")
220 CALL CHAR(104,"0000001")
230 CALL CHAR(112,"000000000C0C0")
240 CALL CHAR(121,"000000FF")
250 CALL CHAR(128,"FFFF3C3C3C18183C")
260 CALL CHAR(96,"0003315EFF5E3103")
270 CALL CHAR(120,"1818181818181818")
280 CALL CHAR(129,"C0C0F9FFFFF9C0C0")
290 REM ///1ST SCREEN///
300 CALL HCHAR(1,9,128,2)
310 CALL HCHAR(1,17,128)
320 CALL HCHAR(1,22,128)
330 CALL VCHAR(12,2,129,3)
340 D=D+1
350 CALL KEY(0,K,ST)
360 IF ST THEN 550
370 IF D<20 THEN 340
380 FOR J=1 TO 11
390 READ R,C,K,L
400 CALL SOUND(-100,880,0)
410 IF K<>0 THEN 450
420 CALL VCHAR(R,C,120,L)
430 CALL VCHAR(R,C,32,L)
440 GOTO 470
450 CALL HCHAR(R,C,121,L)
460 CALL HCHAR(R,C,32,L)
470 NEXT J
480 DATA 2,9,0,11,12,3,11,14,2,17,0,11,2,22,0,22,2,10,0
    ,22
490 DATA 13,3,1,7,2,9,0,22,12,3,1,29,14,3,1,14,2,17,0,2
    2,14,3,1,29
500 GOSUB 1040
510 FOR D=1 TO 200
520 NEXT D
530 RESTORE 1770
540 GOSUB 1060
550 PRINT "NEED INSTRUCTIONS?(Y/N)";
560 CALL KEY(0,K,S)
570 IF (K=89)+(K=78)=0 THEN 560
580 IF K=78 THEN 670
590 CALL CLEAR
600 PRINT " EARTH IS BEING ATTACKED BY ALIENS WHO CALL
    THEMSELVES  COSMIC GUNNERS. THESE ALIENS"
610 PRINT "ATTACK FROM INSIDE THEIR    INVINCIBLE BASE
    AND ARE"
620 PRINT "BEATING EARTH FORCES!! THEY":"CAN ONLY BE DE
    STROYED BY    ACTIVATING THE COSMIC BASE'S"
630 PRINT "SELF-DESTRUCT MECHANISM.":::"HERE ARE THE KEY
    STROKES:":::"ARROW KEYS=MOVE SHIP"::TAB(10);"A=WARP
```

```
    ":::
640 PRINT TAB(6);"SPACE=FUEL STATUS":::"UP,DOWN NOT ACT
    IVE OUTSIDE":"A,SPACE NOT ACTIVE INSIDE"::
650 INPUT "PRESS ENTER":A$
660 REM ///START///
670 CALL CLEAR
680 M1=31
690 V=0
700 A=0
710 F=1200
720 M=31
730 L=22
740 RANDOMIZE
750 GOSUB 1130
760 GOSUB 1210
770 GOSUB 1030
780 CALL VCHAR(L,M1,96)
790 GOSUB 1290
800 CALL KEY(0,K,ST)
810 IF (ST=0)+(K=69)+(K=88)THEN 790
820 GOSUB 1530
830 T=V
840 CALL GCHAR(L,M1,V)
850 IF M1>2 THEN 1000
860 REM //UP A LEVEL//
870 M1=32
880 CALL HCHAR(L,M,T)
890 V=32
900 CALL SOUND(100,540,5)
910 CALL SOUND(300,999,0)
920 F=F+100
930 R$="FUEL CELL REGENERATED"
940 X=4
950 GOSUB 2170
960 IF L=12 THEN 2220
970 L=L-5
980 GOTO 1010
990 REM //GO FORWARD//
1000 CALL VCHAR(L,M,T)
1010 CALL VCHAR(L,M1,96)
1020 GOTO 790
1030 REM //MUSIC//
1040 DU=1
1050 RESTORE 1110
1060 FOR I=1 TO 8
1070 READ S,R
1080 CALL SOUND(R*DU,S,0,S/2,17)
1090 NEXT I
1100 RETURN
1110 DATA 294,350,330,100,349,200,294,190,347,190,440,3
     00,415,800,44733,1
1120 REM //STARS//
1130 S=112
1140 FOR J=1 TO 2
1150 FOR I=I TO 50
```

```
1160 CALL VCHAR(RND*22+2,RND*28+4,S)
1170 NEXT I
1180 S=104
1190 NEXT J
1200 RETURN
1210 REM //PRINT GUNS//
1220 FOR I=5 TO 29 STEP 6
1230 CALL VCHAR(1,I,128)
1240 NEXT I
1250 CALL VCHAR(1,1,130,33)
1260 CALL VCHAR(15,2,130,10)
1270 RETURN
1280 REM //SHOOT GUNS//
1290 RANDOMIZE
1300 R=INT(RND*5+1)*6-1
1310 CALL SOUND(-100,880,0)
1320 CALL VCHAR(2,R,120,22)
1330 CALL VCHAR(2,R,32,22)
1340 IF R<>M1 THEN 1510
1350 N1=L
1360 F=F-100
1370 IF F>0 THEN 1460
1380 R$="    YOU HAVE FAILED     "
1390 X=3
1400 GOSUB 1780
1410 GOSUB 2170
1420 FOR D=1 TO 500
1430 NEXT D
1440 CALL CLEAR
1450 GOTO 2060
1460 CALL SOUND(300,2000,0,-5,5)
1470 CALL VCHAR(N1,M1,96)
1480 R$="HIT DEFLECTED: "&STR$(F)&" F. "
1490 X=4
1500 GOSUB 2170
1510 RETURN
1520 REM //INPUT//
1530 M=M1
1540 N=N1
1550 F=F-5
1560 IF F<0 THEN 1870
1570 IF K<>83 THEN 1600
1580 M1=M1-1
1590 RETURN
1600 IF K<>65 THEN 1710
1610 F=F-30
1620 M1=M1-2
1630 FOR I=16 TO 0 STEP -4
1640 CALL SOUND(200,(I+3*42),I)
1650 NEXT I
1660 CALL SOUND(90,-5,5)
1670 R$="    **** WARPED ****     "
1680 X=3
1690 GOSUB 2170
1700 RETURN
```

```
1710 IF K<>32 THEN 1750
1720 R$="    FUEL STATUS: "&STR$(F)&"    "
1730 X=4
1740 GOSUB 2170
1750 M1=M1-1
1760 RETURN
1770 DATA 294,350,330,100,349,200,415,190,440,200,349,3
     00,294,800,44733,1
1780 REM //EXPLOSION//
1790 V=0
1800 FOR I=145 TO 131 STEP -1
1810 CALL SOUND(-200,110,V,-5,V)
1820 CALL VCHAR(N1,M1,I)
1830 V=V+2
1840 NEXT I
1850 CALL HCHAR(N1,M1,32)
1860 RETURN
1870 IF R$="2" THEN 1890
1880 N1=L
1890 FOR I=500 TO 2000 STEP 50
1900 CALL SOUND(-80,I,0)
1910 NEXT I
1920 CALL SOUND(100,999,0)
1930 CALL VCHAR(N1,M1,98)
1940 FOR D=1 TO 50
1950 NEXT D
1960 CALL VCHAR(N1,M1,97)
1970 CALL SOUND(10,1000,1)
1980 CALL VCHAR(N1,M1,32)
1990 FOR D=1 TO 800
2000 NEXT D
2010 REM //END MESSAGE//
2020 CALL CLEAR
2030 PRINT "AFTER YOUR SELF-DESTRUCTION, THE COSMIC GUN
     NERS MOVED IN AND ATTACKED THE EARTH AND  WERE ";
2040 PRINT "SUCCESSFUL IN INVADING":"HER.":::::
2050 GOTO 2070
2060 PRINT "THE COSMIC GUNNERS HAVE  SUCCEEDED IN DESTR
     OYING THE EARTH'S LAST HOPE."::"THE EARTH IS THEIR
     S."
2070 PRINT :::::::"YOU WILL BE REMEMBERED FOR  TRYING."
2080 DU=2
2090 GOSUB 1050
2100 FOR D=1 TO 1000
2110 NEXT D
2120 CALL CLEAR
2130 PRINT TAB(8);"T H E  E N D":::::::::::
2140 GOSUB 1060
2150 STOP
2160 REM //PRINT MESSAGES//
2170 FOR I=1 TO LEN(R$)
2180 J=ASC(SEG$(R$,I,1))
2190 CALL VCHAR(24,X+I,J)
2200 NEXT I
2210 RETURN
```

```
2220 REM ///STAGE II///
2230 CALL HCHAR(L,M,32)
2240 CALL CHAR(42,"")
2250 RF=3
2260 M=31
2270 N=12
2280 M1=31
2290 N1=12
2300 CALL CLEAR
2310 CALL VCHAR(L,M,96)
2320 CALL HCHAR(1,1,130,321)
2330 CALL HCHAR(14,1,130,321)
2340 CALL HCHAR(10,4,32,2)
2350 CALL HCHAR(14,4,32,2)
2360 FOR J=10 TO 14 STEP 4
2370 FOR I=4 TO 31 STEP 4
2380 CALL VCHAR(J,I,32)
2390 NEXT I
2400 NEXT J
2410 RANDOMIZE
2420 IF INT(RND*2)THEN 2450
2430 CALL HCHAR(10,4,42,2)
2440 GOTO 2460
2450 CALL HCHAR(14,4,42,2)
2460 FOR I=1 TO 5
2470 CALL HCHAR(9+I,1,130,2)
2480 NEXT I
2490 FOR I=1 TO 3
2500 CALL VCHAR(10+I,3,129)
2510 NEXT I
2520 GOTO 2830
2530 CALL KEY(1,K,ST)
2540 IF ST=0 THEN 2810
2550 F=F-5
2560 IF F<0 THEN 1890
2570 IF K<>5 THEN 2600
2580 N1=N1-1
2590 GOTO 2690
2600 IF K+1<>1 THEN 2630
2610 N1=N1+1
2620 GOTO 2690
2630 IF K<>2 THEN 2670
2640 M1=M1-1
2650 IF M1<1 THEN 3070
2660 GOTO 2690
2670 IF M1>29 THEN 2690
2680 M1=M1+1
2690 CALL GCHAR(N1,M1,CH)
2700 IF CH=32 THEN 2770
2710 CALL VCHAR(N1,M1,96)
2720 CALL VCHAR(N,M,32)
2730 IF CH=42 THEN 2950
2740 R$="2"
2750 GOSUB 1790
2760 GOTO 2020
```

```
2770 CALL VCHAR(N,M,32)
2780 N=N1
2790 M=M1
2800 CALL VCHAR(N1,M1,96)
2810 RETURN
2820 REM //FIRE GUNS II//
2830 RANDOMIZE
2840 R=INT(RND*3+1)
2850 IF M<5 THEN 2870
2860 CALL HCHAR(R+10,4,121,INT((M1-4)/2))
2870 GOSUB 2530
2880 CALL SOUND(-200,-4,0,880,4)
2890 CALL HCHAR(RF+10,4,121,29)
2900 CALL HCHAR(RF+10,4,32,29)
2910 IF RF+10<>N1 THEN 2930
2920 GOSUB 1360
2930 RF=R
2940 GOTO 2830
2950 IF N1=14 THEN 3010
2960 CALL HCHAR(7,1,32,3)
2970 CALL VCHAR(5,3,32,2)
2980 CALL HCHAR(5,3,32,2)
2990 CALL VCHAR(5,5,32,5)
3000 GOTO 2770
3010 CALL HCHAR(17,1,32,3)
3020 CALL VCHAR(17,3,32,2)
3030 CALL HCHAR(19,3,32,2)
3040 CALL VCHAR(15,5,32,5)
3050 GOTO 2770
3060 REM //STAGE III//
3070 CALL CLEAR
3080 PRINT TAB(5);"YOU ARE ALMOST THERE":::::"NOW, YOU M
     UST DETONATE THE"::"COSMIC BASE'S SELF-DESTRUCT"::
3090 PRINT "MECHANISM BY ENTERING A"::"SECRET CODE BY U
     SING THE"::"KEYS MARKED 1 TO 9. YOU CAN"::
3100 PRINT "MAKE ONLY 8 ERRORS.":::::"MAY THE FORCE BE
     WITH YOU"
3110 GOSUB 1040
3120 FOR D=1 TO 700
3130 NEXT D
3140 T=48
3150 Y=13
3160 REM //GET CODE//
3170 A$="123456789"
3180 B$=""
3190 FOR I=1 TO 9
3200 RANDOMIZE
3210 X=INT(RND*LEN(A$)+1)
3220 B$=B$&SEG$(A$,X,1)
3230 A$=SEG$(A$,1,X-1)&SEG$(A$,X+1,9)
3240 NEXT I
3250 CALL CLEAR
3260 PRINT "   SELF-DESTRUCT CONTROL":::::::::TAB(13);"E
     RRORS":::::::::">>>>>>>>>>_________<<<<<<<<<":::::
     :
```

```
3270 CALL KEY(0,K,S)
3280 IF (S=0)+(K<49)+(K>57)THEN 3270
3290 S=POS(B$,CHR$(K),1)+12
3300 CALL VCHAR(18,S,K)
3310 IF S<>Y THEN 3370
3320 CALL SOUND(200,660,8)
3330 CALL SOUND(100,550,8)
3340 Y=Y+1
3350 IF Y=22 THEN 3540 ELSE 3270
3360 REM //WRONG//
3370 FOR D=1 TO 30
3380 CALL SOUND(-99,7777,5)
3390 NEXT D
3400 CALL VCHAR(18,S,95)
3410 T=T+1
3420 CALL VCHAR(10,17,T)
3430 IF T<57 THEN 3270
3440 REM //TOO BAD//
3450 FOR I=16 TO 2 STEP -1
3460 CALL SOUND(-999,1760/I,I,335,I,297,I,-5,0)
3470 CALL SCREEN(I)
3480 NEXT I
3490 CALL SOUND(-4250,-6,0)
3500 CALL SOUND(4250,-7,9)
3510 CALL SOUND(4250,-7,24)
3520 GOTO 2020
3530 REM ///MADE IT!///
3540 CALL CLEAR
3550 PRINT TAB(9);"YOU MADE IT!!":::::::::::"NOW, GET
     OUT!!!"
3560 FOR D=110 TO 400 STEP 3
3570 CALL SOUND(-99,D,5)
3580 NEXT D
3590 CALL CLEAR
3600 CALL COLOR(13,2,2)
3610 CALL COLOR(9,10,1)
3620 CALL HCHAR(9,1,130,448)
3630 FOR I=1 TO 32 STEP 3
3640 CALL VCHAR(9,I,32)
3650 NEXT I
3660 CALL VCHAR(1,1,32,240)
3670 CALL HCHAR(12,1,32,64)
3680 FOR I=1 TO 5
3690 CALL HCHAR(I+8,I+10,32,3)
3700 NEXT I
3710 CALL VCHAR(12,16,96)
3720 CALL VCHAR(10,11,130,5)
3730 CALL COLOR(13,15,1)
3740 FOR I=32 TO 16 STEP -.5
3750 CALL VCHAR(13,I,104)
3760 D=D+6
3770 CALL SOUND(-999,D,3)
3780 CALL VCHAR(13,I,32)
3790 NEXT I
3800 CALL SOUND(200,-7,6)
```

```
3810 X=12
3820 FOR I=16 TO 1 STEP -1
3830 X=X-.5
3840 CALL VCHAR(X,I,96)
3850 D=D+6
3860 CALL SOUND(-999,D,3,-7,4)
3870 CALL VCHAR(X,I,32)
3880 NEXT I
3890 FOR I=700 TO 1400 STEP 10
3900 CALL SOUND(-99,I,0)
3910 NEXT I
3920 FOR I=3 TO 15
3930 CALL COLOR(13,I+1,1)
3940 CALL VCHAR(RND*13+9,RND*20+12,32,5)
3950 CALL SOUND(-999,-5,0)
3960 NEXT I
3970 FOR I=3 TO 16
3980 CALL COLOR(13,I,1)
3990 CALL HCHAR(RND*13+9,RND*20+12,32,5)
4000 CALL SOUND(-999,-6,I)
4010 NEXT I
4020 CALL CHAR(130,"F00FF00FF00FF00F")
4030 CALL SOUND(300,-7,9)
4040 CALL SOUND(100,-6,9)
4050 CALL CHAR(130,"213")
4060 CALL SOUND(200,-7,9)
4070 CALL SOUND(100,-6,9)
4080 CALL CLEAR
4090 PRINT "  YOU HAVE SINGLE HANDEDLY  DESTROYED THE I
     NVINCIBLE    COSMIC GUNNERS AND SAVED THEWORLD."::
4100 PRINT " YOU ARE A HERO !!!!":::::::::::
4110 GOTO 2080
4120 GOTO 4120
```

CHAPTER 6

Typing Skill

(BASIC)

Introduction

Typing Skill is one of the most interesting games of this book. Written in TI BASIC, it is intended to test your skill in using an ordinary typewriter, stimulate your mind, check your reaction, and exercise your fingers. It is a real family game because everyone in the family can play no matter how young he or she is. The eagerness of the players for getting high scores makes the game exciting and competitive.

A randomly generated letter, or symbol, will appear one at a time traveling across a small rectangular field, moving from right to left at a specific preset speed. You have to search for the same letter or symbol on your computer console keyboard and press the correct key in no more than a second. This checks your attentiveness and memory and creates an animated atmosphere.

Your typing skill will increase as you play and enjoy the game without having to resort to the hard practice methods used to learn typing with an ordinary typewriter. Your reaction time will also be faster and sharper after playing *Typing Skill* for a certain period of time.

Enjoy the game and learn while playing.

Features

1. The game is programmed to have 12 speeds ranging from 10 (slowest) to −1 (fastest). Each number sets one specific speed for the play.

2. A total of 30 letters and symbols, including SPACE (an invisible character) will appear randomly, one at a time, and will travel across a small rectangular area, moving from right to left at a preset speed.
3. A scoring system is devised that counts each correct letter pressed; the sooner you press a key, the higher the score will be. Also, the score will add up and be finalized on the screen after finishing each play.
4. Play ends if any letter or symbol reaches the left side of the rectangular field.

How To Play, Step-By-Step

1. Run the program. Input Y for instructions, N for none.
2. The computer will ask you to set the game speed by inputting a number between −1 and 10. Type in the desired speed and press ENTER.
3. A letter or symbol will appear from the right side of the small rectangular field drawn for the play. This letter/symbol will move to the left at a preset speed. You are required to press the key on your computer keyboard that corresponds to the letter/symbol as fast as you can before the letter/symbol reaches the left side of the screen; failure to be fast enough means the end of the game.
4. Step 3 will be repeated until the game is over. This occurs when either a letter or symbol has reached the left side of the playing area or when you have successfully zapped them 30 times. Your score will appear on the screen to mark your skill.

Program Explanation

0010-0040	Program name.
0050-0080	Define screen color and redefine characters.
0090-0140	Display title and ask for instructions.
0150-0240	Instructions.
0250-0310	Ask for speed and set speed variables.
0320-0400	Draw screen.
0420-0440	Set variables; (SC) score, (SH) number of times hit, (I) score counter.
0450	Randomize.
0460-0640	Loop for 30 letters.
0470-0530	Randomly get letter/symbol.

0540-0610	Move letter/symbol and GOSUB for key check (0730) and time delay (1030).
0620-0630	Exit loop.
0650-0720	Game ends; display score and number of times hit. Ask if replay wanted.
0730-0770	Subroutine for key check.
0780-0860	Hit—sound, add and print score.
0870-0900	Miss—sound.
0910-0940	Subroutine for calculating and adding score.
0950-1020	Subroutine for printing score.
1030-1070	Subroutine for time delay.

TYPING SKILL

[BASIC]

```
10 REM  ////////////////
20 REM  / TYPING SKILL /
30 REM  ////////////////
40 REM
50 CALL SCREEN(16)
60 CALL CHAR(42,"FFFFFFFFFFFFFFFF")
70 CALL COLOR(2,7,1)
80 CALL CLEAR
90 REM   ///TITLE///
100 REM
110 PRINT TAB(6);" ***************":TAB(6);"*
      *":TAB(6);"* TYPING SKILL *"
120 PRINT TAB(6);"*               *";TAB(6);"***********
    ****":::::::::::
130 INPUT "DO YOU NEED INSTRUCTIONS?":A$
140 IF POS(A$,"Y",1)=0 THEN 250
150 CALL CLEAR
160 REM  ///INSTRUCTIONS///
170 REM
180 PRINT " THIS IS A FAST ACTION GAME.THE PURPOSE OF T
    HIS GAME IS TO TEST YOUR TYPING SKILL."
190 PRINT " LETTERS WILL BE PASSING    THROUGH A SMALL
    SCREEN FROM LEFT TO RIGHT. THE FASTER"
195 PRINT "YOU ELIMINATE THEM, THE     BETTER YOUR SCOR
    E!"
200 PRINT " TO DESTROY THE LETTERS, YOUMUST PRESS THE C
    ORRESPONDING KEY."
210 PRINT " THE SPEED OF THE LETTERS ISSET BY YOU."
220 PRINT :" 0 IS THE FASTEST SPEED AND 10 IS THE SLOWE
    ST.":" A NEGATIVE VALUE WILL MOVE"
230 PRINT "THE LETTERS TWO SPACES AT A TIME.":" GOOD LU
    CK AND BEWARE OF THE":
240 PRINT """INVISIBLE CHARACTER""(SPACE)":
250 INPUT "SPEED?":KT
260 IF KT>10 THEN 250
```

```
270 IF KT>=0 THEN 310
280 Q1=-2
290 KT=ABS(KT)
300 GOTO 320
310 Q1=-1
320 REM   //DRAW SCREEN//
330 REM
340 CALL CLEAR
350 PRINT TAB(10);"0123456789"::::::::
360 PRINT TAB(12);"SCORE":TAB(13);"0":::TAB(12);"SPEED"
    :TAB(13);KT:::::::::
370 CALL HCHAR(4,12,61,10)
380 CALL HCHAR(6,12,61,10)
390 CALL VCHAR(4,11,91,3)
400 CALL VCHAR(4,22,93,3)
410 REM  //START//
420 SC=0
430 SH=0
440 I=1
450 RANDOMIZE
460 REM   /LOOP FOR 30 LETTERS/
470 FOR LI=1 TO 30
480 CALL HCHAR(5,I,32)
490 WE=7
500 C=INT(RND*43+48)
510 IF C=61 THEN 500
520 IF C<>81 THEN 550
530 C=32
540 REM   /LOOP TO MOVE LETTERS/
550 FOR I=21 TO 12 STEP Q1
560 CALL HCHAR(5,I,C)
570 GOSUB 750
580 GOSUB 1050
590 CALL HCHAR(5,I,32)
600 IF WE<>7 THEN 640
610 NEXT I
620 CALL SOUND(400,-5,0)
630 LI=31
640 NEXT LI
650 REM  //GAME DONE//
660 REM
670 PRINT "YOU'VE GOT";SH;"LETTERS."
680 REM  /TRY AGAIN?/
690 REM
700 INPUT "WOULD YOU LIKE TO TRY AGAIN?":B$
710 IF POS(B$,"Y",1)THEN 250
720 STOP
730 REM  /PRESS RIGHT KEY?/
740 REM
750 CALL KEY(0,K,S)
760 IF S=0 THEN 900
770 IF K<>C THEN 890
780 REM   //HIT !//
790 CALL SOUND(50,2999,0)
800 CALL SOUND(100,3999,0)
```

```
810 CALL HCHAR(5,I,42)
820 GOSUB 930
830 GOSUB 970
840 SH=SH+1
850 WE=0
860 RETURN
870 REM  //MISS!//
880 REM
890 CALL SOUND(90,444,0)
900 RETURN
910 REM  //ADD TO SCORE//
920 REM
930 SC=SC+(I-12)
940 RETURN
950 REM  //PRINT SCORE//
960 REM
970 M$=STR$(SC)
980 FOR M=1 TO LEN(M$)
990 N=ASC(SEG$(M$,M,1))
1000 CALL HCHAR(11,14+M,N)
1010 NEXT M
1020 RETURN
1030 REM  //DELAY//
1040 REM
1050 FOR Q=1 TO KT*10
1060 NEXT Q
1070 RETURN
```

CHAPTER 7

Spelling Test

(BASIC)

Introduction

This program is for the TI 99/4A only. It uses the Auto Sprite Definition method to design symbols of the Webster phonetic alphabet. The Auto Sprite Definition program is a powerful tool that makes use of TI 99/4(A) BASIC to draw graphics, figures, and lines. The Auto Sprite Definition program is presented later in the book.

A set of vocabulary words and their equivalent phonetic spellings is input to the computer for the spelling test. Running the program will initiate phonetic symbols on the screen; you are asked to write the equivalent word in English. The correct or incorrect words of your answer are registered as such in the computer memory and are displayed for comparison when the test has been completed. You can input 30 words for the current program but it can be expanded to store as many words as your computer memory can handle. All you have to do is modify the program by changing the dimension of the arrays, W$, W1$, and W2$, to the desired amount and set array W to the maximum allowable errors.

Spelling Test is ideal for grade school and high school English education. It illustrates the old maxim: "Learning while playing."

Features

1. The program is designed to use both lowercase and capital alphabet letters and an imitation of Webster phonetic symbols.
2. Your list of spelling words can be saved on diskettes or cassette tape for use exclusively with the program.

3. The spelling test could be taken from a list of words available from three sources:
   A. Data previously input to the program.
   B. A separate memory input.
   C. A spelling list previously saved on diskette or cassette tape.
4. For the purpose of a spelling exercise, a display of both the correct and incorrect words is programmed to appear at the end of the test so that you can see your misspelled words (for the sake of correct spelling).
5. This program is written for a maximum of 30 words input, data, or saved list but it is expandable to the computer capacity.

How To Play, Step-By-Step

1. Run the program. A display of options on the screen will give you a chance to choose one of the following options:
   A. *TO TAKE TEST.* Key in number 1 to take this option.
   B. *TO INPUT WORDS.* This will give you a chance to renew your spelling list whenever the previous list is no longer interesting to you. Type in number 2 for this option.
   C. *TO END PROGRAM.* This will stop the program.
2. When you choose to take the test, these displays will appear:
   A. *IN DATA.* This option will work only if you already have a list of words in your program (by way of DATA statements); then you can continue to take your test. If your program has no data and you do not know how to key in your data, stop the program. Then rerun it so that you can choose option 2 (TO INPUT WORDS). This will give you an example of the format needed to input your words in data.
   B. *IN MEMORY.* This option can only be used when you have previously input a list into the memory.
   C. *ON DISK.* Choosing this option will allow you to use a list recorded on diskette.
   D. *ON CASSETTE.* This option will allow you to use a list recorded on cassette tape.
3. When you input option 2 (TO INPUT WORDS), the screen will show you an example how to input your spelling words in the correct format. The phonetic symbols are accessible by control characters from the keyboard. A list of these control characters is given at the end of this section.

4. If you make a mistake and have pressed ENTER, just go ahead and finish typing the rest of the list. Then, type 9,9 when all the words have been input. The screen will display instructions to allow correction.
5. When the test begins, phonetic symbols for each word will appear in the center of the screen. You are to pronounce and spell the word correctly by keying in its spelling. The process will continue until all the words contained in the list have been spelled.
6. After you finish the last word, the computer will correct your spelling and will list your misspelled words along with the correct spelling for your comparison.

Phonetic Symbols

Control Characters	A B C D E F G H I J K L M N O P Q R
Phonetic Letters	ă ā â ä ĕ ē ə ĭ ī î ŏ ō ô ŭ û ü o͝o o͞o

Program Explanation

0010-0040	Program name.
0050-0060	Dimension W$ for words, W1$ for words in phonetic spelling, and W2$ for incorrectly spelled words.
0070-0130	Data to define phonetic characters.
0140-0210	Data to define lowercase alphabet.
0220-0320	Defining characters.
0330	Set keyboard control to lowercase mode and control characters.
0350-0380	Clear screen and display title.
0390-0420	Clear screen and display main options.
0430-0480	Ask for input; then branch to the selections.
0490-0560	Inputting words; I = number of words to input.
0590-0660	Word editing.
0680-0720	Ask for word storage of the previously input list, by displaying the diskette or cassette options.
0730-0760	Initialize file processing for diskette.
0770-0820	Record list; then return to main options.
0830-0880	Initialize file processing for cassette and go to line 770 to record list.
0890-0950	Display testing selection and ask for option.
0960-1040	Process word list from disk.
1050-1100	Process word list from cassette.

1110-1160	Initialize word list from data in program.
1170-1250	Taking the test; WR set for 0 wrong; line 1120 prints out the phonetic spelling and line 1230 asks for the right spelling.
1260-1350	Correcting the spelled words.
1360-1400	Print the misspelled words, along with correctly spelled ones.
1410-1430	Ask for replay.
1440	YOUR data. OPTIONAL!

Notes

1. When inputting your vocabulary list into data or memory, you should look up the words in a Webster dictionary to find the correct phonetic spelling.
2. When keying in the program, disregard line 510 until later (do not type this line yet). Run the program for the phonetic symbols to be defined and then stop the program (Function 4). Now type in line 510 as it is shown. For phonetic symbols, use control characters. With the CTRL (Control) key depressed, type the control character that corresponds to the phonetic letter you want.

SPELLING TEST
[BASIC]

```
10 REM /////////////////
20 REM / SPELLING TEST /
30 REM /////////////////
40 REM
50 OPTION BASE 1
60 DIM W$(30),W1$(30),W2$(30),WR1(30)
70 REM  /PRONUNCIATION CHA./
80 DATA 423C0038043C443A,007E0038043C443A,10280038043C4
   43A,00240038043C443A
90 DATA 443800384478403C,007C00384478403C,00000078043C4
   438
100 DATA 4438001000101010,007C001000101010,102800100010
    1010
110 DATA 4438003844444438,007C003844444438,102800384444
    4438
120 DATA 4438004848484834,1028004848484834,004800484848
    4834
130 DATA 423C0044AAAAAA44,007C0044AAAAAA44
140 REM  /LOWERCASE/
150 DATA 00000038043C443E,0040404078444478,000000384440
```

```
    403C,000404043C44443A
160 DATA 00000038447C403C,00304840F0404040,000038443C04
    4438,004040407048484C
170 DATA 0020002020202458,0004000404044438,004040485060
    7048,0060202020202070
180 DATA 000000A854545454,0000005824242424,000000384444
    4438,0000384444784040
190 DATA 000038444C340404,0000005824202020,0000003C4038
    0478,0020207020202418
200 DATA 0000004848484834,0000004444282810,000000444454
    5428,0000004428102844
210 DATA 00002424241C0438,0000007C0418207C
220 REM /INITIALIZE/
230 RESTORE 150
240 FOR I=97 TO 122
250 READ A$
260 CALL CHAR(I,A$)
270 NEXT I
280 RESTORE 80
290 FOR I=129 TO 146
300 READ A$
310 CALL CHAR(I,A$)
320 NEXT I
330 CALL KEY(5,II,II)
340 REM /1ST SCREEN/
350 CALL CLEAR
360 PRINT TAB(8);"SPELLING TEST"::TAB(8)::::::::::::::"
    Press any key...";
370 CALL KEY(0,K,S)
380 IF S=0 THEN 370
390 REM /START/
400 CALL CLEAR
410 WR=0
420 PRINT "PRESS"::"(1) TO TAKE TEST"::"(2) TO INPUT WO
    RDS"::"(3) TO END PROGRAM"::
430 INPUT I
440 IF I<>3 THEN 470
450 PRINT ::::::"GOODBYE..."
460 STOP
470 IF I=1 THEN 890
480 IF I<>2 THEN 430
490 REM /INPUT WORDS/
500 CALL CLEAR
510 PRINT "* WORD INPUT *":::"Enter ""9,9"" to stop."::
    :"EXAMPLE: spelling,spl'ng"::
520 I=0
530 I=I+1
540 IF I=31 THEN 580
550 INPUT W$(I),W1$(I)
560 IF W$(I)<>"9" THEN 530
570 REM /EDIT WORD/
580 CALL CLEAR
590 PRINT "* WORD EDITING *"::"Press ENTER if correct."
    :"If wrong, enter any letter. Then, retype words: a
    s,as":::
```

```
600 W=I-1
610 FOR I=1 TO W
620 PRINT W$(I),W1$(I);
630 INPUT A$
640 IF A$="" THEN 660
650 INPUT W$(I),W1$(I)
660 NEXT I
670 REM /OUTPUT TO STORAGE/
680 INPUT "DO YOU WISH TO SAVE THE      WORDS?(Y/N):":A$
690 IF (A$<>"Y")*(A$<>"y")THEN 400
700 CALL CLEAR
710 PRINT "PRESS:"::"(1) FOR DISK"::"(2) CASSETTE"::
720 INPUT I
730 REM //DISK OUTPUT//
740 IF I<>1 THEN 840
750 INPUT "FILENAME? EXAMPLE:DSK1.WORDS":A$
760 OPEN #1:A$,INTERNAL,VARIABLE,OUTPUT,SEQUENTIAL
770 PRINT #1:W
780 FOR I=1 TO W
790 PRINT #1:W$(I),W1$(I)
800 NEXT I
810 CLOSE #1
820 GOTO 400
830 REM //CASSETTE OUTPUT//
840 IF I<>2 THEN 720
850 PRINT ::"* INSERT CASSETTE TAPE    CS1   THEN PRESS E
    NTER"::
860 INPUT A$
870 OPEN #1:"CS1",INTERNAL,OUTPUT,SEQUENTIAL,FIXED
880 GOTO 770
890 REM ///TEST///
900 CALL CLEAR
910 PRINT "YOUR TEST IS:"::"(1) IN DATA"::"(2) IN MEMOR
    Y"::"(3) ON DISK"::"(4) ON CASSETTE"::
920 INPUT I
930 IF I=1 THEN 1120
940 IF I=2 THEN 1180
950 IF I<>3 THEN 1050
960 REM //DISK INPUT//
970 INPUT "FILENAME? EXAMPLE:DSK1.FOOT ":A$
980 OPEN #1:A$,INTERNAL,SEQUENTIAL,VARIABLE,INPUT
990 INPUT #1:W
1000 FOR I=1 TO W
1010 INPUT #1:W$(I),W1$(I)
1020 NEXT I
1030 CLOSE #1
1040 GOTO 1180
1050 IF I<>4 THEN 920
1060 REM //CASSETTE INPUT//
1070 PRINT :::"* INSERT CASSETTE TAPE    CS1   THEN PRESS
     ENTER"::
1080 INPUT A$
1090 OPEN #1:"CS1",INTERNAL,SEQUENTIAL,FIXED,INPUT
1100 GOTO 990
```

```
1110 REM/DATA WORDS/
1120 RESTORE 1450
1130 W=30
1140 FOR I=1 TO W
1150 READ W$(I),W1$(I)
1160 NEXT I
1170 REM/SPELL!/
1180 PRINT : : : : : : : : : : : : : :"LET'S GET STARTED!
     ": : : : : : : : : : : : : : : : : : : : : : : : : :
1190 PRINT TAB(12);"SPELL..": : : : : : : : : : : : : : :
     : : : : : : : : : :
1200 WR=0
1210 FOR I=1 TO W
1220 PRINT TAB(INT(28-LEN(W1$(I)))/2-2);"(";W1$(I);")":
     : : : : : :
1230 INPUT W2$(I)
1240 PRINT : : : :
1250 NEXT I
1260 PRINT : : : : : : : : : : : : : : : : : : :"I WILL N
     OW CHECK YOUR         SPELLING."
1270 PRINT : : : : : : : : : : : : : :
1280 FOR I=1 TO W
1290 IF W$(I)<>W2$(I)THEN 1330
1300 C=C+1
1310 NEXT I
1320 GOTO 1360
1330 WR=WR+1
1340 WR1(WR)=I
1350 GOTO 1310
1360 PRINT "YOU MISSPELLED ";WR;"WORD(S):": :
1370 PRINT "INCORRECT       CORRECT": : :
1380 FOR I=1 TO WR
1390 PRINT W$(WR1(I)),W2$(WR1(I)): :
1400 NEXT I
1410 PRINT "DO YOU WANT SAME LISTING?   (Y/N)"
1420 INPUT A$
1430 IF (A$="Y")+(A$="y")THEN 1190 ELSE 400
1440 REM/DATA FOR WORDS/
1450 DATA available,a vā'lə bəl,basically,bā'sĭk lē,beg
     inning,bĭ gĭn'ĭng,believe,bĭ lĭv'
1460 DATA benefit,bĕn'ə fĭt,benefited,bĕn ə fĭt'ĭd,boun
     dary,boun'də rē,Britain,Brĭt'ən
1470 DATA calendar,kăl'ən dər,campaign,kăm pān',capital
     ,kăp'ə təl,category,kăt'ə gôr ē
1480 DATA certificate,sər tĭf'ĭ kĭt,characteristic,kăr
     ĭk tə rĭs' tĭk,chief,chēf
1490 DATA circuit,sur'kĭt,circumstance,sur'kəm stăns,ci
     vilization,sĭv ə lə zā'shən
1500 DATA column,kŏl'əm,commissioner,kə mĭsh' ən ər,com
     mittees,kə mĭt'ēs,comparison,kəm păr'ə sən
1510 DATA competent,kŏm'pə tənt,competition,kŏm pə tĭsh
     'ən,conceivable,kən sēv'ə bŏ
1520 DATA conception,kən sĕp'shən,confidential,kŏn fə d
     ĕn'shəl,conscience,kŏn'shəns
1530 DATA conscious,kŏn'shəs,consistency,kən sĭs'tən sē
```

CHAPTER 8

Address Inventory

(BASIC)

Introduction

A record of the addresses and telephone numbers of relatives and friends is needed in every home. Now that we are entering the computer age, why hassle with paper and pencil when you have a computer? With *Address Inventory*, your TI 99/4(A) is a faithful filing clerk. It will store the name, address, and telephone number of up to 200 people on a diskette and it will retrieve any name upon your command. Editing address files can easily be done using *Address Inventory*. A listing of the names only (to check on name availability) can be done alphabetically or nonalphabetically. Also, provisions for printing addresses out using a TI 99/4 Impact Printer are available.

Features

1. This program is designed to store up to 200 names along with an address and a telephone number for each name.
2. A disk system with at least one disk drive is required to run the program.
3. A TI 99/4 Impact Printer or other RS-232-compatible printer is optional.
4. Information is inputted in three main categories:
   A. **Name:** Names can be twenty characters in length.
   B. **Address:** Addresses are input as STREET/CITY/STATE ZIP
   C. **Telephone:** Phone numbers that are inputted will include the area code.
5. Searches available:
   A. **Name:** Names may be searched using any letter or letters in a name.
   B. **Address:** Addresses may be searched in the same way as names.

C. **Telephone:** Phone numbers may be searched by any number combination.

6. Instruction screens (menus) require only a press of a key to proceed.

How To Use, Step-By-Step

1. Key in and save both programs.
2. Run ADDRESS INVENTORY FILE INITIALIZATION program with diskette in drive 1. This will initialize ADDFILE1 and ADDFILE2 files on the diskette for use with *Address Inventory*.
3. Run ADDRESS INVENTORY program.
4. STARTING SCREEN: To Input Data, press 2.
   A. Enter information as asked:

```
LAST NAME?_

MIDDLE INITIALS & FIRST NAME

<EXAMPLE> I.M. SMART
?_

ADDRESS <EXAMPLE>
298 EDISON AVENUE!SAN FRANCI
SCO!CALIFORNIA 94876
?_ (remember to use "!" to separate
STREET!CITY!STATE ZIP-CODE)

TELEPHONE <EXAMPLE>
(415)978-2819
?_
```

   B. The data entered is put into a mailing format and you are asked:

```
IS IT ALL RIGHT?(Y/N)_
```

   Entering "N" will bring you back to the starting screen. Enter "Y" and the data will be recorded to disk (although the disk drive may not be working at this time).

   C. You are returned to the STARTING SCREEN.

5. OUTPUT SCREEN: For Information, press 1.

A. FOR NAME LISTING, press 1.
   1. Enter "Y" if you want an alphabetical name listing (this will take longer than a normal listing).
   2. Hold down any key to halt the listing.
   3. Pressing ENTER at any time will return you to the OUTPUT SCREEN.

B. FOR NAME SEARCH, press 2.
   1. Searches are very flexible. For example, if you wanted to find JOE J.K. SMITH, just answer the prompt.

      WHOSE ADDRESS WOULD YOU LIKE
      TO SEARCH FOR? (LAST,FIRST) _

      The inputs:

      SMITH,JOE
      JOE,SMITH
      JOE,JOE
      SMITH,
      J.K.,
      SM,OE

      would find Mr. Smith, and everyone else with those characters in their names.

      For accuracy, input: SMITH, J.K. JOE
   2. A name and an address will now appear on the screen. If it is not the right person, press any key except ENTER or FNCT P to proceed with search.
   3. Pressing FNCT P will cause the name and address to be printed out, in mailing form, to the optional printer.
   4. Pressing ENTER will return you to the OUTPUT SCREEN.

C. FOR TELEPHONE SEARCH, press 3.
   1. The following message should appear on the screen:

      SEARCH FOR WHAT NUMBER?
      _

      The inputs:

      (415)584-4035
      584-4035
      -4035
)584-

will search for any telephone number that has the inputted characters in the correct order. (Do not use spaces unless you have entered phone numbers with spaces.)

2. The name and address will be displayed.
3. Pressing FNCT P will print the name and address, in mailing form, out to an optional printer.
4. Pressing ENTER will return you to the OUTPUT SCREEN.
5. Pressing any other key will continue the searching.

D. FOR ADDRESS SEARCH, press 4.

1. You should see the following message on the screen:

```
ENTER ADDRESS TO BE SEARCHED
(STREET,CITY,STATE ZIP)
?_
```

The inputs:

```
289 EDISON AVENUE,SAN FRANCISCO,CALIFORNIA
28,SAN FRA,CA
EDIS,289,
94876,,FRAN
```

will find Mr. Smart's file and maybe others also, but the inputs:

```
28 EDISON,S FRAN,CA.
CA 94876,,
```

will not find his file.

2. Name and address will be displayed.
3. Press FNCT P to print the name and address, in mailing form, to an optional printer.
4. Press ENTER to return to the OUTPUT SCREEN.
5. Press any other keys to continue the search.

E. To return to the STARTING SCREEN, press 5.

6. EDIT SCREEN: To Edit Files, press 3.

A. TO DELETE FILE, press 1.

1. The following message should appear:

```
WHOSE ADDRESS WOULD YOU LIKE
TO DELETE (LAST,FIRST)
?_
```

Input data like you did for Name Search.

2. Name and address will be displayed.
3. Press D to delete file displayed and to return to the STARTING SCREEN.
4. Press ENTER to return to the STARTING SCREEN.
5. Press any other key to continue the search.

B. TO EDIT FILE, press 2.

1. Enter full name when asked:

```
WHOSE ADDRESS WOULD YOU LIKE
TO FIX (LAST,FIRST)?_
```

2. Follow the same procedures for inputs as in creating files (Step 4).
3. You will be returned to STARTING SCREEN when done.

C. TO RETURN to STARTING SCREEN, press 3.

7. TO END PROGRAM
   <IMPORTANT: You must exit this way or you may lose all data.
8. It is a good practice to enter BYE instead of pressing QUIT to get out of BASIC.
9. Use DISK MANAGER MODULE to periodically create a backup copy of ADDFILE1 and ADDFILE2.
10. You will not need FIRST MIDDLE INIT. LAST's file after you have one of a real person in ADDFILE1 and ADDFILE2. You may want to delete that file.

Program Explanation

0010-0040	Program name.
0050-0120	Initialize.
0130-0230	Define colors; starting screen.
0240-0540	Input files.
0550-0700	Output screen.
0710-0890	Sequential name listing (list as inputted).
0900-1190	Alphabetical name listing (sorted).
1200-1390	Name search.
1700-1930	Address search.
1940-2110	Subroutine to print a file (name, address, and phone number) on screen or printer.
2120	Load N$ array (for searches).
2130-2190	Load names.

2200-2260	Load addresses.
2270-2330	Load telephone numbers.
2340-2460	Edit screen.
2470-2720	Delete file.
2730-3010	Edit file.
3020-3050	End, close files.

Notes

Although *Address Inventory* is designed to store 200 names and addresses, roughly only 70 files can be manipulated by the program in TI BASIC when searching for addresses (depending on their lengths). When searching for names and telephone numbers, you may be able to work with roughly 100 files. In TI Extended BASIC, if you are fortunate enough to have the 32K Memory Expansion unit, you will be able to work with close to 200 files if your addresses aren't too long.

The reason for having a full memory in cases of address search is that the address string is longer than the name and telephone string. To search, *Address Inventory* loads either name, address, or telephone files into the computer's memory and then looks for the right word keys. Because addresses are usually pretty long, the computer may run out of memory while loading or manipulating them.

On instances when you are unable to search for an address (due to full memory), RUN the program again and look for the person living at that address or look for his telephone number.

If you ever get a MEMORY FULL error, you may have to delete some files. First of all, whenever you get any kind of error, you must close all files. You can do this very simply by entering RUN and then holding down CLEAR (FNCT 4/SHIFT C) or by editing any line of the program. To delete a file on a MEMORY FULL error, RUN *Address Inventory* and then choose the DELETE FILE option and delete some of the unnecessary files. In order to find out which names to delete, use the NAME LISTING procedure (nonalphabetical). You should have enough memory to do this.

ADDRESS INVENTORY
[BASIC]

```
10 REM//////////////////////
20 REM/ ADDRESS INVENTORY /
30 REM/  INITIALIZATION   /
40 REM//////////////////////
50 REM RUN THIS PROGRAM WITH DISKETTE YOU ARE GOING TO
   STORE ADDRESS INVENTORY ON IN DISK DRIVE 1
60 OPEN #1:"DSK1.ADDFILE1",INTERNAL,FIXED 20,RELATIVE 2
   00
70 OPEN #2:"DSK1.ADDFILE2",INTERNAL,RELATIVE 200
80 PRINT #2,REC 0:1
90 PRINT #1,REC 1:"LAST, MIDDLE INITIAL, FIRST"
100 PRINT #2,REC 1:"ADDRESS!CITY!STATE ZIP","(452)382-2
    838"
110 CLOSE #1
120 CLOSE #2
```

ADDRESS INVENTORY
[BASIC]

```
10 REM//////////////////////
20 REM/ ADDRESS INVENTORY /
30 REM//////////////////////
40 REM
50 REM//INITIALIZE//
60 DIM N$(200)
70 OPEN #1:"DSK1.ADDFILE1",INTERNAL,FIXED 20,RELATIVE 2
   00
80 OPEN #2:"DSK1.ADDFILE2",INTERNAL,RELATIVE 200
90 INPUT #2,REC 0:HN
100 FOR I=1 TO 8
110 CALL COLOR(I,2,16)
120 NEXT I
130 REM//STARTING SCREEN//
140 CALL CLEAR
150 CALL SCREEN(3)
160 PRINT : : : : : :TAB(6);"ADDRESS INVENTORY": : : :
170 PRINT TAB(5);"(1) FOR INFORMATION": :TAB(5);"(2) TO
    INPUT DATA": :TAB(5);"(3) TO EDIT FILES"
180 PRINT :TAB(5);"(4) TO END PROGRAM": : : : : : : :
190 CALL KEY(0,K,S)
200 IF (S=0)+(K>52)+(K<49)THEN 190
210 CALL VCHAR((K-49)*2+10,10,42)
220 CALL SOUND(50,440,0)
230 ON K-48 GOTO 560,250,2350,3030
240 REM//INPUT DATA//
250 CALL SCREEN(8)
260 IF HN<200 THEN 310
270 PRINT "SORRY, FILE FULL WITH 200   NAMES."
280 FOR I=1 TO 1000
290 NEXT I
300 GOTO 140
```

```
310 CALL VCHAR(1,3,32,672)
320 PRINT "  <<<<<<<DATA INPUT>>>>>>>": : : :
330 INPUT "LAST NAME? ":L$
340 PRINT :"MIDDLE INITIALS & FIRST NAME":" <EXAMPLE> I
    .M. SMART"
350 INPUT F$
360 PRINT :"ADDRESS <EXAMPLE>":"298 EDISON AVENUE!SAN F
    RANCISCO!CALIFORNIA 94876"
370 INPUT H$
380 I=POS(H$,"!",1)
390 IF (I=0)+(POS(H$,"!",I+1)=0)THEN 370
400 PRINT :"TELEPHONE <EXAMPLE>":"(415)978-2819"
410 INPUT P$
420 WN$=L$&" "&F$
430 CALL VCHAR(1,3,32,672)
440 GOSUB 2010
450 PRINT : : : : : : : : :
460 INPUT "IS IT ALL RIGHT?(Y/N)":A$
470 IF A$="N" THEN 140
480 IF A$<>"Y" THEN 460
490 HN=HN+1
500 PRINT #1,REC HN:WN$
510 PRINT #2,REC 0:HN
520 PRINT #2,REC HN:H$,P$
530 N$(HN)=WN$
540 GOTO 140
550 REM//OUTPUT SCREEN//
560 CALL SCREEN(11)
570 PRINT : : : : : : : :TAB(7);"OUTPUT SCREEN": : : :
580 PRINT TAB(5);"(1) NAME LISTING": :TAB(5);"(2) NAME
    SEARCH": :TAB(5);"(3) PHONE SEARCH"
590 PRINT :TAB(5);"(4) ADDRESS SEARCH": :TAB(5);"(5) RE
    TURN": : : : : :
600 CALL KEY(0,K,S)
610 IF (S=0)+(K>53)+(K<49)THEN 600
620 CALL SOUND(50,440,0)
630 CALL VCHAR((K-49)*2+10,10,42)
640 IF K=53 THEN 140
650 CALL SCREEN(K-37)
660 IF K>50 THEN 1470
670 IF N$(0)="N" THEN 690
680 GOSUB 2140
690 CALL VCHAR(1,3,32,672)
700 IF K-49 THEN 1210
710 REM//LISTING//
720 PRINT "  <<<<<<NAME LISTING>>>>>>": : : : :
730 F=0
740 INPUT "ENTER P FOR PRINTER OUTPUT:":P$
750 IF P$<>"P" THEN 790
760 OPEN #3:"RS232.DA=8.BA=4800"
770 F=3
780 PRINT #F: : : : :"ADDRESS INVENTORY NAME LIST": :
790 INPUT "ALPHABETICAL (Y/N)?":P$
800 IF P$="Y" THEN 910
810 IF P$<>"N" THEN 790
```

```
820 PRINT : :"  HOLD DOWN ANY KEY TO STARTOR ENTER TO E
    XIT": : : :
830 I=0
840 IF I=HN THEN 1150
850 I=I+1
860 PRINT #F:I;">";N$(I)
870 CALL KEY(0,K,S)
880 IF K=13 THEN 1150
890 IF S THEN 870 ELSE 840
900 REM//IN ORDER//
910 PRINT : :"HOLD DOWN ANY KEY TO HALT, OR ENTER TO EX
    IT.": : : : :
920 L=0
930 K=0
940 I=1
950 FOR S=2 TO HN
960 IF N$(I)<N$(S)THEN 980
970 I=S
980 NEXT S
990 IF L=ASC(N$(I))THEN 1020
1000 L=ASC(N$(I))
1010 PRINT #F:CHR$(L)
1020 PRINT #F:"  ";N$(I)
1030 CALL KEY(0,Q,S)
1040 IF Q=13 THEN 1090
1050 IF S THEN 1030
1060 N$(I)=CHR$(92)&N$(I)
1070 K=K+1
1080 IF K<>HN THEN 940
1090 CALL SOUND(100,888,0)
1100 PRINT : :"PLEASE WAIT..."
1110 FOR I=1 TO HN
1120 IF POS(N$(I),CHR$(92),1)=0 THEN 1140
1130 N$(I)=SEG$(N$(I),2,80)
1140 NEXT I
1150 IF F=0 THEN 1170
1160 CLOSE #3
1170 PRINT : :"PRESS ENTER TO RETURN"
1180 CALL KEY(0,K,S)
1190 IF K=13 THEN 560 ELSE 1180
1200 REM//NAME SEARCH//
1210 Q=0
1220 PRINT "  <<<<<<<NAME SEARCH>>>>>>>": : : : :
1230 INPUT "WHOSE ADDRESS WOULD YOU LIKETO SEARCH FOR?<
     LAST,FIRST>  ":A$,B$
1240 K=0
1250 K=K+1
1260 IF K>HN THEN 1390
1270 IF POS(N$(K),A$,1)=0 THEN 1250
1280 IF POS(N$(K),B$,1)=0 THEN 1250
1290 WN$=N$(K)
1300 INPUT #2,REC K:H$,P$
1310 GOSUB 1950
1320 PRINT : : : : : : : :"   PRESS ENTER TO RETURN":"O
     R ANY OTHER KEY TO CONTINUE";
```

```
1330 CALL KEY(0,I,S)
1340 IF S=0 THEN 1330
1350 IF I-34 THEN 1380
1360 GOSUB 2030
1370 GOTO 1330
1380 IF (I=13)+(K>HN)THEN 560 ELSE 1250
1390 IF Q THEN 1380
1400 REM//NO FILE MSG//
1410 CALL SCREEN(2)
1420 CALL SOUND(99,110,9)
1430 PRINT : : :"*******FILE NOT FOUND*******";
1440 FOR I=1 TO 1000
1450 NEXT I
1460 GOTO 140
1470 IF K-51 THEN 1710
1480 REM//PHONE SEARCH//
1490 IF N$(0)="P" THEN 1510
1500 GOSUB 2280
1510 CALL VCHAR(1,3,32,672)
1520 PRINT "  <<<<TELEPHONE SEARCH>>>>": : : : :
1530 INPUT "INPUT ANY PART OF PHONE     NUMBER: (XXX)XX
     X-XXXX        ":A$
1540 K=0
1550 Q=0
1560 K=K+1
1570 IF K>HN THEN 1690
1580 IF POS(N$(K),A$,1)=0 THEN 1560
1590 INPUT #1,REC K:WN$
1600 INPUT #2,REC K:H$,P$
1610 GOSUB 1950
1620 PRINT : : : : : : : :"   PRESS ENTER TO RETURN":"O
     R ANY OTHER KEY TO CONTINUE";
1630 CALL KEY(0,I,S)
1640 IF S=0 THEN 1630
1650 IF I-34 THEN 1680
1660 GOSUB 2030
1670 GOTO 1630
1680 IF (I=13)+(K>HN)THEN 560 ELSE 1560
1690 IF Q THEN 1680 ELSE 1410
1700 REM//ADDRESS SEARCH//
1710 IF N$(0)="A" THEN 1730
1720 GOSUB 2210
1730 CALL VCHAR(1,3,32,672)
1740 PRINT "  <<<<<ADDRESS SEARCH>>>>>": : : : :
1750 INPUT "INPUT ANY PART OF ADDRESS:  (ADDRESS,CITY,S
     TATE ZIP)     ":A$,B$,Z$
1760 Q=0
1770 K=0
1780 K=K+1
1790 IF K>HN THEN 1930
1800 IF POS(N$(K),A$,1)=0 THEN 1780
1810 IF POS(N$(K),B$,1)=0 THEN 1780
1820 IF POS(N$(K),Z$,1)=0 THEN 1780
1830 INPUT #1,REC K:WN$
1840 INPUT #2,REC K:H$,P$
```

```
1850 GOSUB 1950
1860 PRINT : : : : : : : :"   PRESS ENTER TO RETURN":"O
     R ANY OTHER KEY TO CONTINUE";
1870 CALL KEY(0,I,S)
1880 IF S=0 THEN 1870
1890 IF I-34 THEN 1920
1900 GOSUB 2030
1910 GOTO 1870
1920 IF (I=13)+(K)HN)THEN 560 ELSE 1780
1930 IF Q THEN 1920 ELSE 1410
1940 REM//PRINT SUBR//
1950 CALL VCHAR(1,3,32,672)
1960 PRINT " PRESS FNCT P FOR HARD COPY": : : : : :
1970 GOTO 1990
1980 CALL VCHAR(1,3,32,672)
1990 L$=SEG$(WN$,1,POS(WN$," ",2)-1)
2000 F$=SEG$(WN$,POS(WN$," ",2)+1,80)
2010 F=0
2020 GOTO 2050
2030 OPEN #3:"RS232.DA=8.BA=4800"
2040 F=3
2050 PRINT #F: : : :SEG$(" "&F$,POS(" "&F$,". ",1)+1,80
     );SEG$(" "&F$,1,POS(" "&F$,". ",1)+1);L$
2060 Q=POS(H$,"!",1)
2070 PRINT #F:"  ";SEG$(H$,1,Q-1):"  ";SEG$(H$,Q+1,POS(
     H$,"!",Q+1)-Q-1):"  ";SEG$(H$,POS(H$,"!",Q+1)+1,10
     0)
2080 PRINT #F:"  TEL. ";P$
2090 IF F=0 THEN 2110
2100 CLOSE #3
2110 RETURN
2120 REM//LOAD ARRAY N$(HN) SBRS//
2130 REM//LOAD NAMES//
2140 N$(0)="N"
2150 PRINT "PLEASE WAIT...";
2160 FOR I=1 TO HN
2170 INPUT #1,REC I:N$(I)
2180 NEXT I
2190 RETURN
2200 REM//LOAD ADDRESSES//
2210 N$(0)="A"
2220 PRINT "PLEASE WAIT...";
2230 FOR I=1 TO HN
2240 INPUT #2,REC I:N$(I)
2250 NEXT I
2260 RETURN
2270 REM//LOAD PHONES//
2280 N$(0)="P"
2290 PRINT "PLEASE WAIT...";
2300 FOR I=1 TO HN
2310 INPUT #2,REC I:Z$,N$(I)
2320 NEXT I
2330 RETURN
2340 REM//EDIT FILES//
2350 CALL SCREEN(4)
```

```
2360 PRINT : : : : : : : :TAB(8);"EDIT SCREEN": : : :
2370 PRINT TAB(5);"(1) TO DELETE FILE": :TAB(5);"(2) TO
     EDIT FILE": :TAB(5);"(3) RETURN": : : : : : : : :
     :
2380 CALL KEY(0,K,S)
2390 IF (S=0)+(K<48)+(K>51)THEN 2380
2400 CALL VCHAR((K-49)*2+10,10,42)
2410 CALL SOUND(50,440,0)
2420 IF K=51 THEN 140
2430 IF N$(0)="N" THEN 2450
2440 GOSUB 2140
2450 CALL VCHAR(1,3,32,672)
2460 IF K-49 THEN 2740
2470 PRINT "  <<<<<<DELETE FILES>>>>>>": : : :
2480 INPUT "WHOSE ADDRESS WOULD YOU LIKETO DELETE?<LAST
     ,FIRST>       ":A$,B$
2490 Q=0
2500 I=0
2510 I=I+1
2520 IF I>HN THEN 2630
2530 IF POS(N$(I),A$,1)=0 THEN 2510
2540 IF POS(N$(I),B$,1)=0 THEN 2510
2550 WN$=N$(I)
2560 INPUT #2,REC I:H$,P$
2570 GOSUB 1980
2580 PRINT : : : : : : : :"     PRESS D TO DELETE ":"
       ENTER TO RETURN":" ANY OTHER KEY TO CONTINUE";
2590 CALL KEY(0,K,S)
2600 IF K=68 THEN 2640
2610 IF (K=13)+(I>HN)THEN 2360
2620 IF S=0 THEN 2590 ELSE 2510
2630 IF Q THEN 2610 ELSE 1410
2640 CALL SOUND(1000,-7,9)
2650 PRINT #1,REC I:N$(HN)
2660 PRINT #2,REC 0:HN-1
2670 INPUT #2,REC HN:H$,P$
2680 PRINT #2,REC I:H$,P$
2690 N$(I)=N$(HN)
2700 HN=HN-1
2710 I=I-1
2720 GOTO 2510
2730 REM//EDITING//
2740 PRINT "  <<<<<<<EDITING>>>>>>>": : : : :
2750 INPUT "WHOSE ADDRESS WOULD YOU LIKETO FIX?<LAST,FI
     RST>          ":L$,F$
2760 K=0
2770 K=K+1
2780 IF K>HN THEN 1410
2790 IF POS(N$(K),L$,1)=0 THEN 2770
2800 IF POS(N$(K),F$,1)=0 THEN 2770
2810 INPUT #2,REC K:H$,P$
2820 PRINT : : : :N$(K): :"ADDRESS:";H$: :
2830 INPUT "ANY CHANGES?(Y/N)":A$
2840 IF A$="N" THEN 2890
2850 IF A$<>"Y" THEN 2830
```

```
2860 INPUT "NEW ADDRESS?":H$
2870 I=POS(H$,"!",1)
2880 IF (I=0)+(POS(H$,"!",I+1)=0)THEN 2860
2890 PRINT : : : :"TELEPHONE:";P$: :
2900 INPUT "ANY CHANGES?(Y/N)":A$
2910 IF A$="N" THEN 2940
2920 IF A$<>"Y" THEN 2900
2930 INPUT "TELEPHONE?":P$
2940 WN$=N$(K)
2950 GOSUB 1980
2960 PRINT : : : : : : : : : :
2970 INPUT "OK?(Y/N)":A$
2980 IF A$="N" THEN 2360
2990 IF A$<>"Y" THEN 2970
3000 PRINT #2,REC K:H$,P$
3010 GOTO 140
3020 REM//END//
3030 CLOSE #1
3040 CLOSE #2
3050 CALL CLEAR
```

CHAPTER 9

Word Search

(BASIC)

Introduction

Word search puzzles, or word sleuths, are sometimes small, sometimes large. It takes you from five minutes to days to find the words you want. This program is designed to help you to find your words in just seconds, whatever the size of your puzzle. It searches for words in all directions: forward, backward, up, down, and diagonally. There is no secret or hidden corner that cannot be detected.

This program is a combination of two small programs that work separately: "Word Search Input" which asks you to key in all the letters of every row of the puzzle that you want to search for words, and "Word Search" which is used to look for the words in the puzzle.

Features

1. The program is written with instructions for every step so that you can comfortably follow and solve your problem without any difficulty.
2. Puzzles that consist of rows and columns filled with words can be saved both on diskette or cassette.
3. The program is also devised to give you the option of searching for the same word more than once.
4. The puzzle could have a maximum matrix of 110 columns by 31 rows.

How To Play, Step-By-Step

1. At first, run the "Word Search Input" program. The options will appear to give you these choices:

A. **(1) TO INPUT PUZZLE.** Choose this option to input your word puzzle.
B. **(2) TO EDIT FILE.** Choose this option to fix any error in your puzzle.
C. **(3) TO TERMINATE.** This will end the program.

2. If you choose number 1, you will then be asked what device your puzzle will be recorded on: disk, or cassette.
3. Then, the computer will ask the number of rows to be input. Now, input your puzzle, row by row.
4. After you finish inputting the puzzle, the computer will ask if you want to change anything. If you input N for no, the puzzle will be saved onto the selected device and the program will return to the screen menu.
5. If you choose number 2 (TO EDIT FILE), the computer will ask the name of the puzzle you want to edit and the device it comes from. You then have to input the row you want to change and retype that row.
6. After you have finished with the inputting and editing of your puzzle, run the "Word Search" program to look for the words.
7. The computer will ask for the device in which the puzzle is stored, so it can place the puzzle into memory.
8. Now, it asks you for the word you want to find. If the word is present in the puzzle, it will display the number of the row and the column that the word starts on. Then, input Y or N when the computer asks if you want to keep looking for the same word.
9. When you are through looking for words, press ENTER to stop the program.

Program Explanation

Word Search Input

0010-0040	Program name.
0050-0090	Initialize array, clear screen, and then print out option screen.
0100-0140	If user wants to input puzzle, then ask for storage device.
0150-0190	Initialize disk.
0200-0250	Initialize cassette.
0260-0340	Puzzle retrieval from external device.

0350-0410	Inputting puzzle; ask for number of rows and letters in each row.
0420-0520	Puzzle editing.
0530-0630	Recording the puzzle; 560—initialize cassette for recording.
0640-0650	End.

Word Search

0010-0040	Program name.
0050-0070	Clear screen, initialize array PZ$ to 31 (rows).
0080-0170	Ask for storage device, then initialize that device.
0180-0220	Get puzzle into memory from external storage.
0230-0240	Clear and change color of screen.
0250-0270	Ask for word to look for and check if length of word <2; then go to end.
0280-0290	Set variable and get first letter of word (F$).
0300-0360	Loop for finding the first letter of the word in the puzzle. When the first letter is found, branch to the directional search routines.
0370-1090	SEARCH ROUTINES.
0380-0460	Search up.
0470-0550	Search right.
0560-0640	Search down.
0650-0730	Search left.
0740-0820	Search diagonally up,right.
0830-0910	Search diagonally down,left.
0920-1000	Search diagonally down,right.
1010-1090	Search diagonally up,left.
1100-1130	Found it! Print out the location and ask to keep looking for the same word from that position on.
1140-1170	Not found. Jump back to keep searching if it has not done the whole puzzle, or else print out NOT FOUND message and branch back to input word.
1180-1190	End.

WORD SEARCH

[BASIC]

```
10 REM///////////////
20 REM/ WORD SEARCH /
30 REM///////////////
40 REM BY KHOA TON
50 REM//INITIALIZE//
60 CALL CLEAR
70 DIM PZ$(31)
80 PRINT "YOUR PUZZLE IS IN:": :"(1) DISK": :"(2) CASSE
   TTE": : :
90 INPUT I
100 IF I<>1 THEN 150
110 REM//DISK INPUT//
120 INPUT "FILENAME? EXAMPLE:DSK1.FOOD ":IO$
130 OPEN #1:IO$,INTERNAL,INPUT ,SEQUENTIAL,VARIABLE
140 GOTO 180
150 IF I<>2 THEN 90
160 REM//CASSETTE INPUT//
170 OPEN #1:"CS1",INTERNAL,INPUT ,SEQUENTIAL,FIXED 128
180 INPUT #1:PS
190 FOR I=1 TO PS
200 INPUT #1:PZ$(I)
210 NEXT I
220 CLOSE #1
230 CALL CLEAR
240 CALL SCREEN(11)
250 PRINT :"----------------------------": :"WHAT WORD
    WOULD YOU LIKE ME TO LOOK FOR?": : :
260 INPUT "PRESS ENTER TO STOP:":W$
270 IF LEN(W$)<2 THEN 1190
280 F$=SEG$(W$,1,1)
290 L=0
300 L=L+1
310 LP=1
320 IF L>PS THEN 1150
330 P=POS(PZ$(L),F$,LP)
340 IF P=0 THEN 1150 ELSE 390
350 LP=P+1
360 IF LP>LEN(PZ$(L))THEN 1150 ELSE 330
370 REM///SEARCH///
380 REM//UP//
390 IF L-LEN(W$)<0 THEN 480
400 LX=L
410 LY=1
420 LX=LX-1
430 LY=LY+1
440 IF SEG$(PZ$(LX),P,1)<>SEG$(W$,LY,1)THEN 480
450 IF (LX=1)*(L<>LEN(W$))THEN 480
460 IF L+1-LX<>LEN(W$)THEN 420 ELSE 1110
470 REM//RIGHT//
480 IF P+LEN(W$)-1>LEN(PZ$(L))THEN 570
490 LX=P
```

```
500 LY=1
510 LX=LX+1
520 LY=LY+1
530 IF SEG$(PZ$(L),LX,1)<>SEG$(W$,LY,1)THEN 570
540 IF (LX=LEN(PZ$(L)))*(LX-P+1<>LEN(W$))THEN 570
550 IF LX+1-P<>LEN(W$)THEN 510 ELSE 1110
560 REM//DOWN//
570 IF L+LEN(W$)>PS+1 THEN 660
580 LX=L
590 LY=1
600 LX=LX+1
610 LY=LY+1
620 IF SEG$(PZ$(LX),P,1)<>SEG$(W$,LY,1)THEN 660
630 IF (LX=PS)*(PS+1-L<>LEN(W$))THEN 660
640 IF LX-L+1<>LEN(W$)THEN 600 ELSE 1110
650 REM//LEFT//
660 IF P-LEN(W$)<0 THEN 750
670 LX=P
680 LY=1
690 LX=LX-1
700 LY=LY+1
710 IF SEG$(PZ$(L),LX,1)<>SEG$(W$,LY,1)THEN 750
720 IF (LX=1)*(P-LX+1<>LEN(W$))THEN 750
730 IF P-LX+1<>LEN(W$)THEN 690 ELSE 1110
740 REM//UP,RIGHT//
750 IF (P-1+LEN(W$)>LEN(PZ$(L)))+(L-LEN(W$)<0)THEN 840
760 LX=L
770 LY=P
780 LX=LX-1
790 LY=LY+1
800 IF SEG$(PZ$(LX),LY,1)<>SEG$(W$,LY-P+1,1)THEN 840
810 IF ((LX=1)*(L-LX+1<>LEN(W$)))*((LY=LEN(PZ$(L)))*(LY
    -P+1<>LEN(W$)))THEN 840
820 IF (LY-P+1=LEN(W$))+(L-LX+1=LEN(W$))THEN 1110 ELSE
    780
830 REM//DOWN,LEFT//
840 IF (P-LEN(W$)<0)+(L-1+LEN(W$)>PS)THEN 930
850 LX=L
860 LY=P
870 LX=LX+1
880 LY=LY-1
890 IF SEG$(PZ$(LX),LY,1)<>SEG$(W$,P-LY+1,1)THEN 930
900 IF ((LX=PS)*(LX-L+1<>LEN(W$)))*((LY=1)*(P-LY+1<>LEN
    (W$)))THEN 930
910 IF (P-LY+1=LEN(W$))+(LX-L+1=LEN(W$))THEN 1110 ELSE
    870
920 REM//DOWN,RIGHT//
930 IF (P-1+LEN(W$)>LEN(PZ$(L)))+(L-1+LEN(W$)>PS)THEN 1
    020
940 LX=L
950 LY=P
960 LX=LX+1
970 LY=LY+1
980 IF SEG$(PZ$(LX),LY,1)<>SEG$(W$,LY-P+1,1)THEN 1020
990 IF ((LX=PS)*(LX-L+1<>LEN(W$)))*((LY=LEN(PZ$(L)))*(L
```

```
    Y-P+1<>LEN(W$)))THEN 1020
1000 IF (LY-P+1=LEN(W$))+(LX-L+1=LEN(W$))THEN 1110 ELSE
     960
1010 REM//UP,LEFT//
1020 IF (P-LEN(W$)<0)+(L-LEN(W$)<0)THEN 350
1030 LX=L
1040 LY=P
1050 LX=LX-1
1060 LY=LY-1
1070 IF SEG$(PZ$(LX),LY,1)<>SEG$(W$,P-LY+1,1)THEN 350
1080 IF ((LX=1)*(L-LX+1<>LEN(W$)))*((LY=1)*(P-LY+1<>LEN
     (W$)))THEN 350
1090 IF (P-LY+1=LEN(W$))+(L-LX+1=LEN(W$))THEN 1110 ELSE
     1050
1100 REM///FOUND IT!///
1110 PRINT : :"YOUR WORD STARTS ON":"ROW:";L:"COLUMN:";
     P: :
1120 INPUT "DO YOU WISH ME TO LOOK      FURTHER?(Y/N)":
     A$
1130 IF A$="Y" THEN 350 ELSE 250
1140 REM///NOT FOUND///
1150 IF L<PS THEN 300
1160 PRINT : :"WORD NOT SPOTTED": :
1170 GOTO 250
1180 REM//STOP//
1190 PRINT : :"GOODBYE...": : :
```

WORD SEARCH INPUT

[BASIC]

```
10 REM//////////////////////
20 REM/ WORD SEARCH INPUT /
30 REM//////////////////////
40 REM BY KHOA TON
50 REM///INITIALIZE///
60 DIM R$(31)
70 CALL CLEAR
80 PRINT TAB(5);"WORD SEARCH INPUT": : : :"(1) TO INPUT
    PUZZLE": :"(2) TO EDIT FILE": :"(3) TO TERMINATE":
    : : : : :
90 INPUT "YOUR CHOICE?":I
100 IF I=3 THEN 650
110 IF (I<(1)+(I>2)THEN 70
120 CALL CLEAR
130 PRINT : : :"STORAGE DEVICE:": :"(1) DISK": :"(2) CA
    SSETTE": :
140 INPUT J
150 REM///DISK///
160 IF J<>1 THEN 210
170 INPUT "FILENAME? EXAMPLE:DSK1.PUZZZ":IO$
180 OPEN #1:IO$,INTERNAL,SEQUENTIAL,VARIABLE
190 GOTO 220
200 REM///CASSETTE///
```

```
210 IF J<>2 THEN 140
220 IF I<>2 THEN 350
230 REM//RETRIEVE PUZZLE FROM EXTERNAL DEVICE//
240 IF J=1 THEN 260
250 OPEN #1:"CS1",INPUT ,INTERNAL,FIXED 128,SEQUENTIAL
260 INPUT #1:SP
270 CALL CLEAR
280 FOR I=1 TO SP
290 INPUT #1:R$(I)
300 PRINT R$(I)
310 NEXT I
320 IF J=1 THEN 430
330 CLOSE #1
340 GOTO 430
350 CALL CLEAR
360 REM//PUZZLE INPUT//
370 INPUT "HOW MANY ROWS:":SP
380 FOR I=1 TO SP
390 PRINT "ROW";I;
400 INPUT R$(I)
410 NEXT I
420 REM/EDIT INPUT/
430 PRINT :"CHANGE ANYTHING (Y/N)";
440 INPUT A$
450 IF A$="N" THEN 540
460 IF A$<>"Y" THEN 430
470 PRINT : :"WHICH LINE";
480 INPUT L
490 IF (L>31)+(L<1)THEN 480
500 PRINT "  ";R$(L)
510 INPUT R$(L)
520 GOTO 430
530 REM///OUTPUT///
540 CALL CLEAR
550 IF J=1 THEN 570
560 OPEN #1:"CS1",OUTPUT,INTERNAL,FIXED 128,SEQUENTIAL
570 RESTORE #1
580 PRINT #1:SP
590 FOR I=1 TO SP
600 PRINT #1:R$(I)
610 NEXT I
620 CLOSE #1
630 GOTO 70
640 REM///END///
650 CALL CLEAR
```

CHAPTER 10

Skeet Shoot

(BASIC)

Introduction

Skeet Shoot is a game of trapshooting in which clay targets are thrown from traps to simulate birds in flight and which are fired upon from many different directions by the shooter. This is a game of competition for two shooters who are stationed at two sides of an open field using three particle-beam guns firing on parallel and diagonal paths. The clay pigeon, instead of being thrown up, materializes and hovers in the air for a short time and then disappears. The winner of the game is the player who is credited with the first ten clay pigeons. Each shot that misses a clay pigeon will cause a point to be deducted from the player's score. If two players hit the pigeon at the same time, ½ point is added to both scores.

Features

1. Graphics, colors, sounds, and music are the main features.
2. The players can choose the speed of the skeet game.
4. The game is a competition for two players; each player uses three fingers to fire the guns at the clay pigeon.
5. There is a deduction of a point each time a shooter misses the target.
6. The first player with 10 clay pigeons shot down will hear the triumphal music designed to serenade the winner.

How To Play, Step-By-Step

1. Run the program.
2. Type Y for instructions, N for none.
3. Enter the speed of the clay pigeon.
4. Red player uses the S, D, and F keys to fire and blue player uses the J, K, and L keys to fire.

5. Fast reflexes are required in order to win.
6. If you miss a clay pigeon, one point will be subtracted from your score. A hit will add one point to your score. If both players hit the clay pigeon at the same time, each will get ½ point.
7. The player who reaches 10 points first will win. If both players reach 10 simultaneously, the game will keep going until one player beats the other.
8. The winner will be awarded with triumphal music.

Program Explanation

0010-0040	Program name.
0050-0120	Put value for pigeon position in array.
0130-0320	Redefine colors and characters.
0330-0370	Show opening screen and ask for instructions.
0380-0450	Instructions.
0460-0520	Choose speed, reset score.
0530-0730	Draw playfield.
0740-0750	Start.
0760-0830	Put up "real" skeet pigeon 2 out of 3 times (one that can be hit by players).
0840-0910	For 1 out of 3 times, put up a "fake" skeet pigeon (one that cannot be hit).
0920-0980	Check for fire keys.
0990-1030	Player 1 shoots.
1040-1080	Player 2 shoots.
1090-1110	Logic check.
1120-1160	Player 1 (red) shot at fake.
1170-1220	Player 2 (blue) shot at fake.
1230	Update score subroutines.
1240-1250	Get ready to display.
1260-1330	Subtract 1 point.
1340-1380	Add 1 point.
1390-1450	Add ½ point.
1460-1510	Timer for pigeon to disappear.
1520-1620	Tie—both players hit.
1630	Red shot . . .
1640-1700	and missed.
1710-1810	and hit.
1820	Blue shot . . .
1830-1890	and missed.

1900-1950	and hit.
1960-1970	Erase beams.
1980-2080	Check for winner. If no winner, then got to line 1490 (put up another skeet). Set value for winner.
2090-2250	End of match. Play song, flash winner, and then go to line 460 (replay).

SKEET SHOOT

[BASIC]

```
10 REM/////////////////
20 REM/ SKEET SHOOT /
30 REM/////////////////
40 REM BY KHOA TON
50 REM
60 REM///INITIALIZE///
70 DIM SA(1,8),SC(1)
80 RESTORE
90 FOR J=0 TO 1
100 FOR I=0 TO 8
110 READ SA(J,I)
120 NEXT I
130 NEXT J
140 DATA 2,3,12,2,3,12,2,3,12,12,12,12,3,3,3,2,2,2
150 CALL CHAR(94,"04060F1C3870FCFE")
160 CALL CHAR(95,"2060F0381C0E3F7F")
170 CALL CHAR(42,"FFFFFFFFFFFFFFFF")
180 CALL CHAR(51,"FFFFFFFFFFFFFFFF")
190 CALL CHAR(112,"00387CFEFEFE7C38")
200 CALL CHAR(113,"003830E0F8F87038")
210 CALL CHAR(114,"90AA548A5EBE9C38")
220 CALL CHAR(115,"09552A517A7D391C")
230 CALL CHAR(116,"18552875757124D")
240 FOR I=101 TO 106
250 CALL CHAR(I,"")
260 NEXT I
270 CALL COLOR(2,9,1)
280 CALL COLOR(3,5,1)
290 CALL COLOR(9,9,1)
300 CALL COLOR(10,5,1)
310 CALL COLOR(11,16,1)
320 F1$="01020C0C1060608"
330 F2$="8040303008060601"
340 REM///1ST SCREEN///
350 CALL CLEAR
360 PRINT TAB(8);"333333333333333":TAB(8);"* SKEET SHOO
    T 3":TAB(8);"****************": : : : : : : : : : :
    :
370 INPUT "NEED INSTRUCTIONS?(Y/N):":A$
380 IF A$<>"Y" THEN 470
390 REM//INSTRUCTIONS//
400 PRINT "   ***** SKEET SHOOT 33333  ": : :" THIS IS
```

```
    A TWO PLAYER GAME.": :" THE OBJECT OF THIS GAME IS"
410 PRINT "TO SHOOT DOWN TEN TARGETS   FASTER THAN YOUR
    OPPONENT.": :" TO ACTIVATE YOUR LASER"
420 PRINT "CANNON,YOU MUST DEPRESS THE LETTER KEY CORRE
    SPONDING TO THAT CANNON.  EACH PLAYER"
430 PRINT "HAS THREE CANNONS WHICH FIREDIAGONALLY.": :"
    TO SCORE ONE TARGET YOU   MUST HIT IT."
440 PRINT " HALF A POINT IS GIVEN TO   EACH PLAYER IF B
    OTH HIT THE TARGET."
450 PRINT " A MISS WILL SUBTRACT ONE   POINT FROM YOUR
    SCORE.": :
460 INPUT "GOOD LUCK! (PRESS ENTER)":A$
470 CALL CLEAR
480 PRINT "CHOOSE LEVEL:": :"   (1) FAST SKEET": :"   (
    2) SLOW SKEET": : : : : : : :
490 INPUT L
500 IF (L>2)+(L<1)THEN 490
510 L=L*5-3
520 SC(0)=.5
530 SC(1)=.5
540 REM///SET SCREEN///
550 CALL CLEAR
560 CALL SCREEN(12)
570 PRINT "   ***** SKEET SHOOT 33333": : : : : : : : :
    : : : : : : : : : : : : : :
580 PRINT " ^   ^   ^          _   _   _*S***D***F**** 3
    333J333K333L";
590 CALL HCHAR(24,31,51)
600 REM//LASER RAYS//
610 DATA 4,11,1,8,13,1,12,15,1,22,15,-1,26,13,-1,30,11,
    -1
620 RESTORE 610
630 Y=100
640 FOR I=1 TO 6
650 Y=Y+1
660 S=0
670 READ X,K,H
680 FOR J=22 TO K STEP -1
690 S=S+H
700 CALL VCHAR(J,S+X,Y)
710 NEXT J
720 NEXT I
730 CALL HCHAR(4,2,42,3)
740 CALL HCHAR(6,2,51,3)
750 REM///START///
760 RANDOMIZE
770 IF INT(RND*3)=0 THEN 860
780 REM//REAL SKEET//
790 SK=1
800 P=INT(RND*9)
810 I=INT(P/3)
820 J=P-I*3
830 X=I*2+12+J*2-2
840 GOTO 910
850 REM//FAKE SKEET//
```

```
860 SK=0
870 P=INT(RND*16)
880 I=INT(P/4)
890 J=P-I*4
900 X=I*2+10+J*2-2
910 W=(17-I*2)+J*2
920 CALL VCHAR(X,W,112)
930 REM//FIRE//
940 CALL KEY(1,K,S)
950 CALL KEY(2,K2,S2)
960 CALL KEY(1,K,S)
970 CALL KEY(2,K2,S2)
980 CALL KEY(1,K,S)
990 CALL KEY(2,K2,S2)
1000 REM/PLAYER1/
1010 IF S=0 THEN 1060
1020 IF (K=2)+(K=3)+(K=12)THEN 1030 ELSE 1060
1030 CALL SOUND(100,330,5)
1040 CALL CHAR(SGN(K-3)+102,F1$)
1050 REM/PLAYER2/
1060 IF S2=0 THEN 1100
1070 IF (K2=2)+(K2=3)+(K2=12)THEN 1080 ELSE 940
1080 CALL SOUND(-100,220,5)
1090 CALL CHAR(SGN(K2-3)+105,F2$)
1100 IF S THEN 1120
1110 IF S2=0 THEN 1480
1120 IF SK THEN 1540
1130 IF S=0 THEN 1180
1140 AX=-1
1150 PL=1
1160 GOSUB 1250
1170 CALL CHAR(SGN(K-3)+102,"")
1180 IF S2=0 THEN 1480
1190 AX=-1
1200 PL=2
1210 GOSUB 1250
1220 CALL CHAR(SGN(K2-3)+105,"")
1230 GOTO 1480
1240 REM//(S)DRAW SCORE//
1250 Y=SC(PL-1)+SC(PL-1)-INT(SC(PL-1))+5
1260 CALL GCHAR(PL*2+2,Y,CH)
1270 REM /SUBTRACT POINT/
1280 IF AX<>-1 THEN 1360
1290 IF SC(PL-1)>1.5 THEN 1320
1300 CH=32
1310 SC(PL-1)=1.5
1320 CALL VCHAR(PL*2+2,Y-1,CH)
1330 CALL VCHAR(PL*2+2,Y,32)
1340 GOTO 1450
1350 REM/ADD POINT/
1360 IF AX<>1 THEN 1410
1370 CALL VCHAR(PL*2+2,Y,112)
1380 CALL VCHAR(PL*2+2,Y+1,CH)
1390 GOTO 1450
1400 REM /ADD HALF POINT/
```

```
1410 IF CH<>113 THEN 1440
1420 CALL VCHAR(PL*2+2,Y,112)
1430 GOTO 1450
1440 CALL VCHAR(PL*2+2,Y,113)
1450 SC(PL-1)=SC(PL-1)+AX
1460 RETURN
1470 REM//TIMER//
1480 T=T+1
1490 IF T<L THEN 940
1500 T=0
1510 CALL VCHAR(X,W,32)
1520 GOTO 760
1530 REM//TIE//
1540 IF (K=SA(0,P))*(K2=SA(1,P))=0 THEN 1650
1550 CALL VCHAR(X,W,116)
1560 CALL SOUND(500,550,0,110,9)
1570 PL=1
1580 AX=.5
1590 CALL SOUND(500,550,9,110,0)
1600 GOSUB 1250
1610 PL=2
1620 GOSUB 1250
1630 GOTO 1970
1640 REM//RED HIT//
1650 IF K=SA(0,P)THEN 1720
1660 IF K=-1 THEN 1840
1670 AX=-1
1680 PL=1
1690 GOSUB 1250
1700 CALL CHAR(SGN(K-3)+102,"")
1710 GOTO 1840
1720 CALL VCHAR(X,W,114)
1730 CALL SOUND(500,440,0,110,5)
1740 PL=1
1750 AX=1
1760 CALL SOUND(500,330,0,110,5)
1770 GOSUB 1250
1780 IF S2=0 THEN 1970
1790 AX=-1
1800 PL=2
1810 GOSUB 1250
1820 GOTO 1970
1830 REM//BLUE HIT//
1840 IF K2=SA(1,P)THEN 1910
1850 IF K2=-1 THEN 1480
1860 AX=-1
1870 PL=2
1880 GOSUB 1250
1890 CALL CHAR(SGN(K2-3)+105,"")
1900 GOTO 1480
1910 CALL VCHAR(X,W,115)
1920 CALL SOUND(500,330,0,110,5)
1930 PL=2
1940 AX=1
1950 CALL SOUND(500,440,0,110,5)
```

```
1960 GOSUB 1250
1970 CALL CHAR(SGN(K-3)+102,"")
1980 CALL CHAR(SGN(K2-3)+105,"")
1990 REM//WIN?//
2000 IF SC(0)=SC(1)THEN 1500
2010 IF (SC(0)>10)+(SC(1)>10)=0 THEN 1500
2020 IF SC(0)<SC(1)THEN 2080
2030 REM//RED WIN//
2040 S=2
2050 X=9
2060 GOTO 2160
2070 REM//BLUE WIN//
2080 S=3
2090 X=5
2100 REM//ENDING SONG//
2110 DATA 262,250,349,375,392,375,440,250,392,500,330,6
     25,262,250,349,375,392,375,440,250
2120 DATA 392,875,262,375,349,375,392,375,440,250,392,5
     00,330,625,330,500,349,250,330,250
2130 DATA 262,375,262,625,523,375,494,375,440,375,392,3
     75,494,375,392,375,440,375,349,625
2140 DATA 392,625,523,375,494,375,440,375,392,250,494,7
     50,523,375,494,375,440,375,392,375
2150 DATA 494,500,392,375,440,500,349,625,392,750,330,2
     50,349,375,330,375,262,250,262,825,22366,1
2160 RESTORE 2110
2170 CALL SCREEN(8)
2180 FOR I=1 TO 51
2190 READ K,Y
2200 CALL SOUND(Y,K,0,K/2,9,K*2,13)
2210 IF I/2=INT(I/2)THEN 2240
2220 CALL COLOR(S,X,1)
2230 GOTO 2250
2240 CALL COLOR(S,16,1)
2250 NEXT I
2260 GOTO 470
```

CHAPTER 11

Biorhythm

(EXTENDED BASIC)

Introduction

The theory of biorhythm states that there are three repeating cycles, which begin at birth, in every person's life. There is the Physical Cycle, the Emotional Cycle, and the Intellectual Cycle. In *Biorhythm,* these cycles range from −1 (low) to 1 (high). Each of these three cycles represents one phase of a person's state of being.

The *Physical Cycle* shows one's physical status. When it is high, a person should feel strong and healthy. When one cannot get out of bed in the morning, his or her Physical Cycle is probably approaching its lowest peak. This cycle lasts 23 days (it takes 23 days to get back to zero while passing through the highest and lowest peaks of the cycle).

The *Emotional Cycle* lasts 28 days and shows one's emotional status. When a person feels secure and calm, his or her Emotional Cycle should be above zero. If one feels edgy and impatient, it could be that his Emotional Cycle is below zero (or he is drinking the wrong kind of coffee).

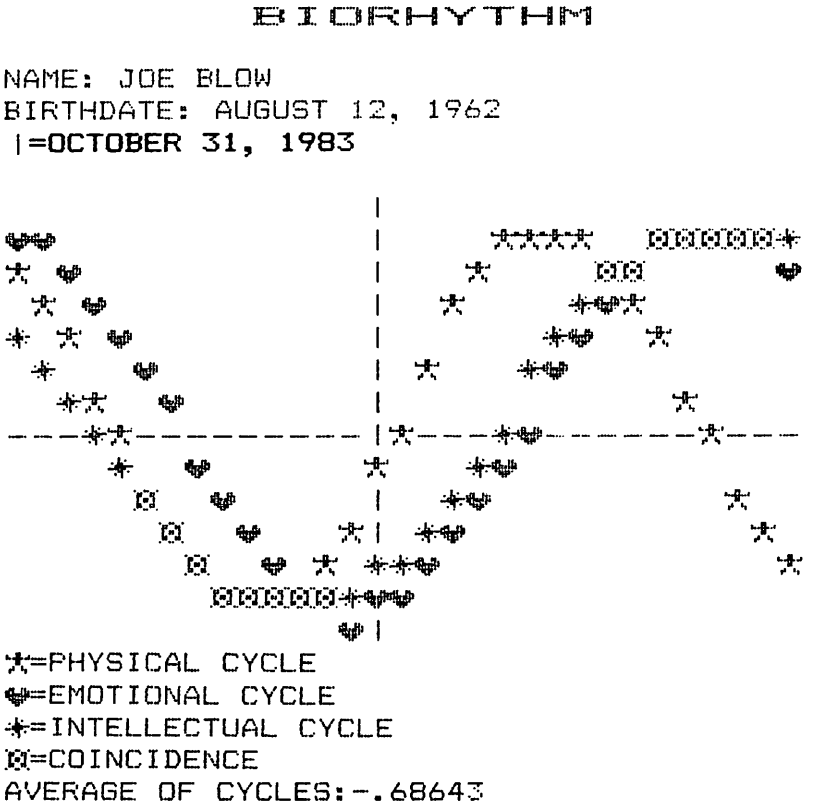
The *Intellectual Cycle* lasts 33 days and shows a person's mental attentiveness. It is said that great decisions are made when this cycle is at its peak.

When all three of these life cycles reach their maximum at the same time, which only happens about twice a year, one is said to be at his or her very best. There are rumors that a man who goes gambling when his cycles coincide at their peaks is bound to win big!

It is up to you as to whether or not you believe in the biorhythm theory, but it can't hurt to know your "great" days. . . .

Features

1. The program uses your birthdate to derive the three cycles (Fig. 11-1).



2. The Physical Cycle is symbolized by a strong human shape. The Emotional Cycle's symbol is a red heart and the Intellectual Cycle uses a sparkling star as its symbol.
3. The three cycles can be plotted daily, monthly, and yearly on the screen, along with the average of all three cycles for any given day.
4. If you have the TI 99/4 Impact Printer, a hard copy of the graph can be made. Also, an automatic printout of graphs for all 12 months of the year, with two months printed on each page, can be made on the TI 99/4 Impact Printer.
5. This biorhythm program is designed for use with the Grego-

rian calendar, which was adopted in 1582. Earlier dates will not be calculated correctly by the program.

6. Speech can also be used to touch up the program (the Speech Synthesizer is optional).

How To Use, Step-By-Step

1. Run the program.
2. Press Y for instructions, N for none.
3. Enter P for printer copy or S for screen copy of the instructions.
4. After all instructions, input your name.
5. If your birthdate is not in memory, input it in the format asked; for example:

```
MONTH OF BIRTH:    AUGUST
DATE OF BIRTH:     1
YEAR OF BIRTH:     1956
```

6. After you have finished entering your date of birth, you are asked to press any of these:

PROCEED (FNCT 6 / SHIFT V)	to continue on.
REDO (FNCT 8 / SHIFT R)	to re-input your birthdate.
" (Quotation mark. FNCT P / SHIFT P)	to print out biorhythm chart.

Your biorhythm, for a whole year, can be printed using a TI 99/4 Impact Printer. (The form length must be set at top of page by loading the paper with top of page at printhead; then, turn the printer on.) The prompts and inputs are:

```
IS PRINTER READY? ___
WHAT YEAR? ___
STARTING MONTH: ___
```

7. If you PROCEEDed, or your birthdate is in memory, input the projection date in the same manner.
8. Press PROCEED or REDO.
9. A biorhythm graph will be graphed on screen (see Fig. 11-1) for 15 days before and 15 days after the projection date. (In-

put 15 for projection date for the month graph and the day markings when printed on printer.)

10. When done, you will have to press any of these:

REDO (FNCT 8 / SHIFT R)	to see another date, go back to Step 3.
BEGIN (FNCT 5 / SHIFT W)	to see a different person's biorhythm, go back to Step 2.
BACK FNCT 9 / SHIFT Z)	to end program

Program Explanation

0010-0040	Program name.
0060-0100	Initialize arrays, colors, and characters.
0110-0160	Starting screen.
0170-0300	Instructions.
0310-0330	Get user's name.
0340-0410	DATA for months and regular user's birthdate.
0420-0500	Birthdate input; if user has birthdate in memory, then use it.
0510-0560	Check months for errors.
0570-0590	Print 12 months initialization.
0600-0670	Projection date inputs.
0680-0730	12 months projection routine.
0740-0790	Check months for error, then ask to proceed.
0800-0830	Calculate days lived.
0840-0910	Display biorhythm graph, CALL GRAPH to graph cycles, and then wait for key to be pressed.
0920	Print out to TI 99/4 Impact Printer.
0930-0990	Define pictures for printer in strings.
1000-1020	Print heading on printer.
1030-1050	If date of projection is 15, then print day markings.
1060-1090	Routine to read screen and print graph on printer.
1100-1150	Finish printing.
1160-1200	SUB GRAPH (to graph cycles).

Notes

To get fully formed graphic symbols on the TI 99/4 Impact Printer, you must set the printer to full graphic mode by setting internal

dip-switch 2, pin 1 to OFF (consult TI 99/4 printer manual, Appendix D). After setting the switch, you must always add .DA=8 in your OPEN statements for the printer.

Remember to change attributes for printer OPENs to match the characteristics of your printer.

Use lines 370 to 390 to put in names and birthdates of users for easy inputs. When the *biorhythm* program asks for name of user, it will check these lines for the name. If it already has the user's name and birthdate in memory, the user won't have to manually input his birthdate. Put data in program in the following way:

Name, month of birth, date of birth, year of birth

For example:

```
370 DATA JOE BLOW,JANUARY,1,1892,MARY BLOW,APRIL,
    4,1983
400 DATA ZZZ,ZZZ,0,0
```

The entry DATA ZZZ,ZZZ,0,0 shown on line 400 must be inserted after all the names and birthdates.

BIORHYTHM
[EXTENDED BASIC]

```
10 !/////////////
20 !/ BIORHYTHM /
30 !/////////////
40 !
50 !///INITIALIZE///
60 DIM DOM(12):: ON WARNING NEXT :: CALL INIT :: CALL L
   OAD(-31878,0)
70 DATA 31,28,31,30,31,30,31,31,30,31,30,31
80 FOR I=1 TO 12 :: READ S :: K=K+S :: DOM(I)=K :: NEXT
   I
90 CALL CHAR(39,"000000007E",96,"1899FF1818246642",104,
   "0066E7FFFF7E3C18",112,"1014583FFC1A2808",120,"817E6
   65A5A667E81")
100 CALL COLOR(9,11,1,10,9,1,11,5,1,12,13,1)
110 !///OPENING///
120 CALL CLEAR :: CALL SCREEN(16):: CALL COLOR(2,9,1)
130 A$=RPT$(" ",7):: B$=RPT$("*",13):: C$="*
    *" :: PRINT A$;B$:A$;C$:A$;"* BIORHYTHM *":A$;C$:A$
    ;B$: : : : : : : : : : : : :"NEED INSTRUCTIONS?(Y
    /N)*";
```

```
140 CALL CHARPAT(42,A$):: B$="000000103810"
150 CALL KEY(0,K,S):: IF K=78 THEN 320 ELSE IF K<>89 TH
    EN CALL SOUND(100,110,3): : CALL CHAR(42,B$):: CALL
    SOUND(200,110,16):: CALL CHAR(42,A$):: CALL SOUND(4
    00,44733,30):: GOTO 150
160 INPUT "TO PRINTER OR SCREEN?(P/S)":A$ :: IF A$="P"
    THEN FILE=1 :: @=23 :: OPEN #1:"RS232.DA=8.BA=4800"
    ELSE IF A$<>"S" THEN 160 ELSE FILE=0 :: @=1
170 !//INSTRUCTIONS//
180 DATA "********* BIORHYTHM ********",,"*THE THEORY O
    F BIORHYTHM","STATES THAT EVERYONE HAS","THREE CYCL
    ES WHICH START ON"
190 DATA THE DAY OF HIS BIRTH.,,*THESE CYCLES ARE:,----
    -PHYSICAL CYCLE(23 DAYS),----EMOTIONAL CYCLE(28 DAY
    S)
200 DATA -INTELLECTUAL CYCLE(33 DAYS),,"*IN THIS PROGRA
    M,CYCLES SPAN",FROM -1(LOW) TO 1(HIGH).
210 DATA ,"*""HIGH"" DAYS OF A CYCLE ARE",WHEN FIGURES
    OF THAT CYCLE,"RISE ABOVE THE MIDDLE LINE,","AND ""
    LOW"" DAYS ARE BELOW IT."
220 DATA ,*WHEN ALL 3 OF THE CYCLES,COINCIDE ABOVE THE
    MIDDLE,"LINE,PERSON IS AT HIS BEST!"
230 CALL CLEAR :: RESTORE 180 :: FOR I=1 TO 23 :: READ
    A$ :: PRINT #FILE:TAB(@);A$ :: NEXT I :: IF FILE=1
    THEN 250 ELSE PRINT "****PRESS ANY KEY";
240 CALL KEY(0,K,S):: IF S=0 THEN 240
250 CALL CLEAR :: PRINT "*WHEN ASKED FOR THESE KEYS, TH
    EY HAVE THE FOLLOWING":"FUNCTIONS:"
260 PRINT :"-PROCEED: CONTINUE PROGRAM": :"-REDO: BACK
    TO INPUT SCREEN": :"-"": IF IN INPUT, SCREEN WILL":
    "PRINT TO PRINTER 12 MONTHS"
270 PRINT "     IF IN GRAPH, SCREEN WILL PRINT TO PRINTE
    R BIORHYTHM  GRAPH": :"-BEGIN: ASK FOR DIFFERENT
    NAME": :"-BACK: END PROGRAM"
280 PRINT #FILE: :TAB(@);"*AVERAGE OF CYCLES IS THE":TA
    B(@);"AVERAGE ON DAY LISTED.";CHR$(12)
290 PRINT :"****PRESS ANY KEY";
300 CALL KEY(0,K,S):: IF S=0 THEN 300 ELSE IF FILE=1 TH
    EN CLOSE #1
310 !///START///
320 CALL CLEAR :: RD,PR=0 :: CALL CHAR(43,"001010101010
    1")
330 INPUT "WHAT IS YOUR NAME?":N$ :: CALL CLEAR :: PRIN
    T : : :"HELLO ";N$: : : : : : : : : : : :: CALL SAY
    ("HELLO")
340 !//MONTHS//
350 DATA JANUARY,3,FEBRUARY,0,MARCH,3,APRIL,2,MAY,3,JUN
    E,2,JULY,3,AUGUST,3,SEPTEMBER,2,OCTOBER,3,NOVEMBER,
    2,DECEMBER,3
360 !//BIRTHDATES//
370 !DATA**USE THIS SPACE TO PUT USUAL USERS' NAME AND
    BIRTHDATE HERE
380 !DATA**IN THIS FORM:(NAME,MONTH,DATE,YEAR). EXAMPLE
    : JOE BLOW,JANUARY,1,1892,MARY BLOW,APRIL,4,1983 ET
```

```
    C...
390 !DATA**BUT PUT IT BEFORE DATA ZZZ,ZZZ,0,0
400 DATA ZZZ,ZZZ,0,0
410 DISPLAY AT(1,1)ERASE ALL:"*********BIRTHDATE*******
    ***": : :"MONTH OF BIRTH:": :"DATE OF BIRTH:": :"YE
    AR OF BIRTH:"
420 IF RD THEN 450
430 RESTORE 370
440 READ A$,MO$,DA,YR :: IF A$="ZZZ" THEN 460 ELSE IF A
    $<>N$ THEN 440
450 DISPLAY AT(4,20)BEEP:MO$ :: DISPLAY AT(6,19):DA ::
    DISPLAY AT(8,19):YR :: GOTO 510
460 CALL SAY("PLEASE+ANSWER THEN PRESS+ENTER"):: ACCEPT
    AT(4,20)SIZE(-9)BEEP:MO$ :: ACCEPT AT(6,20)SIZE(-2)
    BEEP:DA :: ACCEPT AT(8,20)SIZE(-4)BEEP:YR
470 IF (DA>0 AND YR>0 AND DA=INT(DA)AND YR=INT(YR))THEN
    510
480 CALL SAY("#THAT IS INCORRECT# #TRY AGAIN#"):: GOTO
    460
490 IF I=13 THEN B$="WHAT IS A "&MO$&"?" ELSE B$=STR$(D
    A)&" DAYS IN "&MO$&"?!?"
500 DISPLAY AT(24,1):B$ :: GOTO 460
510 RESTORE 350
520 FOR I=1 TO 12 :: READ A$,BT :: IF A$=MO$ THEN MO=I
    :: I=14
530 NEXT I :: IF YR/4=INT(YR/4)AND BT=0 THEN BT=1
540 IF I=13 OR DA>BT+28 THEN 490
550 DISPLAY AT(24,1)BEEP:"PRESS PROCEED,REDO OR "''
560 CALL KEY(0,K,S):: IF K=6 THEN 460 ELSE IF K=34 THEN
    570 ELSE IF K=12 THEN 600 ELSE 560
570 DISPLAY AT(24,1):"IS PRINTER READY?(Y/N):N" :: ACCE
    PT AT(24,24)SIZE(-1)VALIDATE("YN"):A$ :: IF A$="N"
    THEN 550 ELSE DISPLAY AT(24,1):"WHAT YEAR?1983" ::
    ACCEPT AT(24,11)SIZE(-4)VALIDATE(DIGIT):YR1
580 IF YR1<YR THEN 570
590 DISPLAY AT(24,1):"STARTING MONTH:JANUARY" :: ACCEPT
    AT(24,16)SIZE(-9)BEEP:MO1$ :: GOTO 690
600 DISPLAY AT(12,1):"*******PROJECTION DAY*******": :
    :"THE MONTH:": :"THE DATE:": :"THE YEAR:": : : : :
    :
610 IF K=34 THEN 720 ELSE CALL SAY("PLEASE+ANSWER THEN
    PRESS+ENTER")
620 DISPLAY AT(19,20):"1983"
630 ACCEPT AT(15,20)SIZE(-9)BEEP:MO1$ :: ACCEPT AT(17,2
    0)SIZE(-2)BEEP:DA1 :: ACCEPT AT(19,20)SIZE(-4)BEEP:
    YR1
640 IF (DA1>0 AND YR1>0 AND DA1=INT(DA1)AND YR1=INT(YR1
    )AND YR<=YR1)THEN 740
650 CALL SAY("#THAT IS INCORRECT# #TRY AGAIN#"):: GOTO
    630
660 IF I=13 THEN B$="   INVENTED A NEW MONTH?" ELSE B$=
    "ONLY "&STR$(NT+28)&" DAYS IN "&MO1$
670 DISPLAY AT(24,1):B$ :: GOTO 630
680 !///LOOPS FOR 1 YEAR///
690 RESTORE 350
```

```
700 FOR I=1 TO 12 :: READ A$,NT :: IF A$=MO1$ THEN MO1=
    I :: I=14
710 NEXT I :: IF I=13 THEN 590
720 PR=1 :: DA1=15 :: FOR LFY=MO1 TO 12 :: RESTORE 350
    :: FOR K=1 TO LFY :: READ MO1$,NT :: NEXT K :: MO1=
    LFY
730 DISPLAY AT(15,20)BEEP:MO1$ :: DISPLAY AT(17,19):DA1
    :: DISPLAY AT(19,19):YR1 :: GOTO 790
740 RESTORE 350
750 FOR I=1 TO 12 :: READ A$,NT :: IF A$=MO1$ THEN MO1=
    I :: I=14
760 NEXT I :: IF YR1/4=INT(YR1/4)AND NT=0 THEN NT=1
770 IF I=13 OR DA1>NT+28 THEN 660
780 DISPLAY AT(24,1)BEEP:"   PRESS PROCEED OR REDO"
790 CALL KEY(0,K,S):: IF K=6 THEN 630 ELSE IF K<>12 AND
    PR=0 THEN 790
800 !//DAYS LIVED//
810 DTB=INT(365.25*YR+DOM(MO-1))+DA
820 DTN=INT(365.25*YR1+DOM(MO1-1))+DA1
830 DL=DTN-DTB
840 !///SINE GRAPH///
850 DISPLAY AT(1,1)ERASE ALL:"*********BIORHYTHM*******
    ***": :"NAME: ";N$:"+=";MO1$;" ";STR$(DA1);",";YR1
860 CALL HCHAR(12,2,39,31):: CALL VCHAR(5,16,43,14)
870 DISPLAY AT(20,1):CHR$(96);"=PHYSICAL CYCLE":CHR$(10
    4);"=EMOTIONAL CYCLE":CHR$(112);"=INTELLECTUAL CYCL
    E"
880 DISPLAY AT(24,1):"PRESS REDO,BEGIN,"",OR BACK"
890 CALL GRAPH(23,DL):: CALL GRAPH(28,DL):: CALL GRAPH(
    33,DL)
900 DISPLAY AT(23,1):"AVERAGE OF CYCLES:";SEG$(STR$((SI
    N(PI/180*360*DL/23)+SIN(PI/180*360*DL/28)+SIN(PI/18
    0*360*DL/33))/3),1,7)
910 CALL KEY(0,K,S):: IF K=6 THEN RD=1 :: GOTO 410 ELSE
    IF K=14 THEN 320 ELSE IF K=15 THEN CALL CLEAR :: ST
    OP ELSE IF K<>34 AND PR=0 THEN 910
920 !///PRINTOUT///
930 ?=16 :: GR$=CHR$(27)&"K"&CHR$(8)&CHR$(0):: B$=CHR$(
    96)&CHR$(35)&CHR$(38)&CHR$(248)&CHR$(248)&CHR$(38)&
    CHR$(35)&CHR$(96)!/MAN/
940 C$=CHR$(56)&CHR$(124)&CHR$(126)&CHR$(31)&CHR$(31)&C
    HR$(126)&CHR$(124)&CHR$(56)!/HEART/
950 D$=CHR$(8)&CHR$(40)&CHR$(26)&CHR$(252)&CHR$(63)&CHR
    $(88)&CHR$(20)&CHR$(16)!/CROSS/
960 E$=RPT$(CHR$(0),8)!/SPACE/
970 F$=CHR$(0)&RPT$(CHR$(8),6)&CHR$(0)!/DASH/
980 G$=RPT$(CHR$(0),4)&CHR$(255)&RPT$(CHR$(0),3)
990 H$=CHR$(129)&CHR$(126)&CHR$(102)&CHR$(153)&CHR$(153
    )&CHR$(102)&CHR$(126)&CHR$(129)!/COINC/
1000 OPEN #1:"RS232.DA=8.BA=4800.CR"
1010 PRINT #1:CHR$(14);TAB(16);"BIORHYTHM";CHR$(10);CHR
     $(10);TAB(?);"NAME: ";N$;CHR$(10);TAB(?);"BIRTHDAT
     E: ";MO$;" ";STR$(DA);",";YR;CHR$(10)
1020 PRINT #1:TAB(?);GR$&G$;CHR$(27);"E";"=";MO1$;" ";S
```

```
     TR$(DA1);",";YR1;CHR$(2);CHR$(10);CHR$(27);"F";CHR
     $(10)
1030 IF DA1<>15 THEN 1060 ELSE A$="123456789012345678901
     12345678" :: IF MO1=2 AND YR1/4=INT(YR1/4)THEN A$=
     A$&"9" ELSE A$=A$&SEG$("901",1,NT)
1040 PRINT #1:TAB(2);
1050 FOR I=1 TO LEN(A$):: PRINT #1:CHR$(27);"K";CHR$(2)
     ;CHR$(0);CHR$(0);CHR$(0);SEG$(A$,I,1):: NEXT I ::
     PRINT #1:CHR$(10)
1060 FOR I=5 TO 18 :: A$="" :: FOR K=2 TO 32 :: CALL GC
     HAR(I,K,S):: IF S=39 THEN A$=A$&F$ ELSE IF S=120 T
     HEN A$=A$&H$ ELSE IF S>95 THEN 1080 ELSE IF S=43 T
     HEN A$=A$&G$ ELSE A$=A$&E$
1070 GOTO 1090
1080 IF S=96 THEN A$=A$&B$ ELSE IF S=104 THEN A$=A$&C$
     ELSE A$=A$&D$
1090 NEXT K :: PRINT #1:TAB(2);CHR$(27);"K";CHR$(248);C
     HR$(0);A$;CHR$(10):: NEXT I
1100 PRINT #1:TAB(2);GR$&B$;"=PHYSICAL CYCLE";CHR$(10);
     TAB(2);GR$&C$;"=EMOTIONAL CYCLE";CHR$(10);TAB(2);G
     R$&D$;"=INTELLECTUAL CYCLE";CHR$(10)
1110 PRINT #1:TAB(2);GR$&H$;"=COINCIDENCE";CHR$(10)
1120 PRINT #1:TAB(2);"AVERAGE OF CYCLES:";SEG$(STR$((SI
     N(PI/180*360*DL/23)+SIN(PI/180*360*DL/28)+SIN(PI/1
     80*360*DL/33))/3),1,7);CHR$(13)
1130 IF PR=1 THEN IF LFY/2=INT(LFY/2)THEN PRINT #1:CHR$
     (12)ELSE PRINT #1:RPT$(CHR$(10),5)
1140 CLOSE #1 :: IF PR<>1 THEN 910
1150 NEXT LFY :: PR=0 :: GOTO 910
1160 SUB GRAPH(A,B)
1170 FOR Y=2 TO 32 :: X=INT(SIN(PI/180*360*(B+Y-16)/A)*
     -6)+12
1180 CALL GCHAR(X,Y,C):: IF C>95 THEN CALL SOUND(100,99
     9,0):: CALL VCHAR(X,Y,120)ELSE CALL VCHAR(X,Y,(A-2
     3)/5*8+96)
1190 NEXT Y
1200 SUBEND
```

CHAPTER 12

Destroyer Phoenix

(EXTENDED BASIC)

Introduction

The Destroyer Phoenix is a space battle simulator used to test the skills of space cadets in space battles. The simulator places a cadet before an attacking enemy force (of eleven ships). There, he must destroy the entire advancing force or perish in the attempt. In this game, you will be that space cadet.

Given a pilot's view of the battle, you must move your ship in such a way that the target comes into your stationary sight. The target will move erratically trying to avoid being hit. When the target gets into your gunsight cross-hairs, fire your dual pulsar cannons before it can escape. If you hit the ship right in its control tower, the enemy craft will be destroyed. If you hit it at its edges, it will only be partially damaged. Up to three damaging hits on one target will destroy the enemy craft. You will be given five shots at each target and between 20 to 40 counts on the timer, depending on the chosen difficulty level. The simulation will end when the timer reaches zero.

There are three levels of play. A player may choose either cadet mission, pilot mission, or captain's war mission, and is given 40, 30, or 20 counts on the timer for each target, respectively.

Features

1. The program uses various sound tones and vivid colors for the actions and display of aliens, the graphical scanner, and the timer—all in a space battle scene.
2. Pulsar cannons can fire from both the right and left corners to the target as the moving object is detected on a radar scanner.
3. The aliens explode in pieces when hit, accompanied by a full display of colors and sounds.

4. The program is designed to score partial damage of any alien that is not hit exactly. Three hits of this type will also destroy the alien.
5. Three levels of play are provided (novice, intermediate, and professional) depending on the speed of the timer.
6. Messages are given following the losing or winning of a game.
7. Limited energy is provided to shoot the cannons. Firing too much will exhaust the energy supply.
8. The program is designed to be played only with joystick number 1.

How To Play, Step-By-Step

1. Run the program. Press Y for further instructions, N for none.
2. Choose the level of play: (1) Novice, (2) Intermediate, or (3) Professional.
3. Using joystick number 1, try to maneuver the Destroyer Phoenix so that the target appears right in the gunsight of your scanner, and then press the button to fire.
4. When you hit an alien craft in the center or near the center, the craft will explode into many pieces, but if you hit it elsewhere, the craft will be only partially damaged. However, three hits like this will also destroy the craft.
5. The game ends when the timer in the upper right-hand corner of the screen reaches zero or when you have shot down all the alien craft.
6. A message will appear on the screen to tell you of your rank when the game ends.

Program Explanation

0010-0040	Program name.
0050-0100	Define characters.
0110-0120	Define colors.
0130-0170	Display title and ask for instructions.
0180-0240	Instructions.
0260-0280	Clear screen and ask for level.
0290-0330	Set up playing screen.
0340	P@=shots remaining. Main loop, start of computer countdown.
0350	Produce an engine sound and check for firing.
0360	Q,W = enemy's movement.

0370	Check for firing.
0380	Display time.
0390	Check if player has won (shot down more than 10 enemy ships).
0400	Looping of timer.
0410-0510	Ending messages.
0520	Check for key-press to replay.
0530-0640	Subroutines.
0540	Check for joystick and firing.
0550-0630	Firing; check for damage or direct hit on enemy.
0640	Move enemy ship according to your joystick control.
0650-0800	Subprograms.
0660-0730	Subprogram of explosion animation.
0740	Subprogram for randomly moving position of enemy ship.
0750	Subprogram for display of stars.
0760-0780	Subprogram of firing missile from wing.
0790	Delay.
0800	Winning subprogram.

DESTROYER PHOENIX

[EXTENDED BASIC]

```
10 !//////////////////////
20 !/ DESTROYER PHOENIX /
30 !//////////////////////
40 !
50 !//INITIALIZE//
60 A$="00010301012327FF0906"&RPT$("0",14)&"80C08080C4E4
   FF9060"
70 B$="00001B0D1D1F1F053F23204040E000002810D0F0F8F8F880
   FCC404070207"
80 CALL CHAR(42,"1010103838387CEE",44,"0C0C",46,"1",59,
   "",94,"FE1E1E1F1F1E1EFE",95,"818181FFFFFFFF18",96,A$)
90 CALL CHAR(112,"0000000000000FF",113,"7F7F7F7F7F7F7F
   7F")
100 CALL CHAR(136,"0C0C"&RPT$("0",60),140,"0C0C"&RPT$("
    0",16))
110 FOR I=5 TO 8 :: CALL COLOR(I,16,1):: NEXT I
120 CALL COLOR(2,14,1,3,2,9,4,2,9,11,7,1)
130 !//TITLE//
140 CALL CLEAR :: CALL SCREEN(2)
150 DISPLAY AT(9,5):RPT$("*",21):: DISPLAY AT(11,5):"*
    DESTROYER PHOENIX *" :: DISPLAY AT(13,5):RPT$("*",2
    1)
160 DISPLAY AT(24,1)BEEP:"REQUIRE INSTRUCTIONS? (Y/N)"
```

```
170 CALL KEY(0,K,S):: IF S=0 THEN 170 ELSE IF K=78 THEN
    260
180 !/INSTRUCTIONS/
190 CALL CLEAR :: PRINT " THIS GAME REQUIRES JOYSTICKNU
    MBER ONE": :
200 PRINT " YOU ARE IN COMMAND OF ONE  OF THE PHOENIX'S
    SPACECRAFT.": :" YOU MUST MANEUVER THE ALIENSPACECR
    AFT INTO YOUR FIRING"
210 PRINT "SCANNER BEFORE THE TIME IS  UP OR YOUR COMPU
    TER WILL BE OVERLOADED AND EXPLODE.": :" AFTER YOU
    HAVE SHOT DOWN AN"
220 PRINT "ALIEN, YOUR COMPUTER WILL   PICK UP ANOTHER
    ONE AND THE TIME STARTS OVER.": :
230 PRINT " YOU MUST DESTROY THE ALIENSWITH ONE DIRECT
    HIT OR THREEDAMAGING HITS. YOU WILL HAVEONLY 5 SHOT
    S AT EACH TARGET.": :"PRESS ANY KEY";
240 CALL KEY(0,K,S):: IF S=0 THEN 240
250 !//START//
260 CALL CLEAR :: SC=0
270 PRINT "WHICH LEVEL:":TAB(5);"(1) CADET TRAINING":TA
    B(5);"(2) PILOT MISSION":TAB(5);"(3) CAPTAIN WAR MI
    SSION"
280 CALL KEY(0,K,S):: IF S=0 THEN 280 ELSE IF K=49 THEN
    M=40 ELSE IF K=50 THEN M=30 ELSE M=20
290 CALL CLEAR :: CALL SCREEN(2)
300 DISPLAY AT(1,1):"  0" :: CALL HCHAR(2,2,112,11):: C
    ALL VCHAR(2,12,113,2):: CALL HCHAR(2,26,112,6):: CA
    LL VCHAR(2,26,113,2)
310 CALL STARS(25):: CALL HCHAR(1,8,42,5)
320 DISPLAY AT(12,14):CHR$(32);"+";CHR$(32)
330 RANDOMIZE :: CALL SHIP
340 P=5 :: FOR J=M TO 0 STEP -1
350 CALL SOUND(-1000,110,0,-7,5):: CALL KEY(1,K,ST):: I
    F ST THEN GOSUB 550
360 Q=RND*8-4 :: W=RND*8-4 :: GOSUB 540
370 CALL KEY(1,K,ST):: IF ST THEN GOSUB 550
380 DISPLAY AT(1,25):J
390 IF SC>10 THEN CALL WIN :: J=-5
400 NEXT J :: IF J+6 THEN 430
410 !/END MESSAGES/
420 IF M=20 THEN 490 ELSE IF M=30 THEN 470 ELSE 450
430 IF M=30 THEN 460 ELSE IF M=20 THEN 480 ELSE IF SC>1
    8 THEN 450
440 DISPLAY AT(10,1)BEEP:"CADET,YOU NEED MORE TRAINING,
    REPORT TO TRAINING SECTOR AT 0500 HOURS EVERY DAY."
    :: CALL DELAY(1500):: GOTO 500
450 DISPLAY AT(10,1)BEEP:"YOU HAVE BEEN PROMOTED TO   T
    HE PILOT'S QUADRANT." :: CALL DELAY(1500):: GOTO 50
    0
460 IF SC>10 THEN 470 ELSE DISPLAY AT(10,1)BEEP:"YOU AR
    E ALMOST DESTROYED,   RETURN TO PILOT'S QUADRANT."
    :: CALL DELAY(1500):: GOTO 500
470 DISPLAY AT(10,1)BEEP:"YOU HAVE DONE VERY WELL. YOU
    ARE NOW RANKED AS CAPTAIN." :: CALL DELAY(1500):: G
    OTO 500
```

```
480 IF SC>6 THEN 490 ELSE DISPLAY AT(10,1)BEEP:"YOU ARE
    DESTROYED, BUT YOU  WILL BE REMEMBERED AS A      BRA
    VE CAPTAIN." :: CALL DELAY(1500):: GOTO 500
490 DISPLAY AT(10,1)BEEP:"YOU HAVE DEFEATED THE ALIEN S
    PACE FORCE!! YOU ARE NOW A LIVING HERO OF SPACE AND
       EARTH!" :: CALL DELAY(1500)
500 CALL SCREEN(4):: DISPLAY AT(14,7):"INSERT COIN (PRE
    SS REDO)"
510 CALL SOUND(100,890,0):: DISPLAY AT(12,8):"GAME OVER
    " :: FOR D=1 TO 100 :: NEXT D :: DISPLAY AT(12,1):"
     "
520 CALL KEY(0,K,S):: IF S=0 THEN 510 ELSE IF K=6 THEN
    CALL DELSPRITE(ALL):: GOTO 260 ELSE CALL CLEAR :: P
    RINT "GOODBYE..." :: STOP
530 !/SUBROUTINES/
540 CALL JOYST(1,X,Y):: CALL KEY(1,K,S):: IF S=0 THEN 6
    40
550 IF P@<1 THEN CALL HCHAR(1,8,32,5):: RETURN ELSE DIS
    PLAY AT(1,6)SIZE(6):RPT$("*",P@)
560 CALL MOTION(#5,0,0):: DISPLAY AT(24,7):";;TARGET LO
    CKED;;"
570 CALL COINC(#5,88,125,8,H1):: CALL COINC(#5,88,125,3
    ,H2):: CALL HCHAR(15,15,95,5):: CALL VCHAR(10,20,94
    ,5)
580 P@=P@-1 :: CALL SOUND(100,4000,0,-6,2):: CALL FIRE
    :: CALL VCHAR(12,17,59)
590 IF H2 THEN 620
600 IF H1 THEN DISPLAY AT(24,6)BEEP:"    ;;DAMAGED;;" ::
     CALL SOUND(1000,110,15,-4,15):: DA=DA+1 :: IF DA>2
     THEN DA=0 :: GOTO 620
610 GOTO 630
620 SC=SC+1 :: CALL HCHAR(1,8,42,5):: DISPLAY AT(1,2)SI
    ZE(3):SC :: DISPLAY AT(24,7):" ;;DESTROYED;;" ::P@=
    5 :: J=M :: CALL VCHAR(12,17,43):: CALL BLOW
630 CALL SOUND(10,-6,0):: CALL VCHAR(12,17,43):: CALL V
    CHAR(10,20,32,5):: CALL HCHAR(15,15,32,5):: CALL HC
    HAR(24,3,32,30)
640 W=W-X :: Q=Q+Y :: CALL MOTION(#5,Q,W):: RETURN
650 !/SUBPROGRAMS/
660 SUB BLOW :: RANDOMIZE :: CALL SOUND(1000,110,0,-5,0
    ):: CALL MAGNIFY(1)
670 CALL POSITION(#5,L,P):: CALL SPRITE(#5,96,16,L,P,#6
    ,98,16,L,P+8,#7,97,16,L+8,P,#8,99,16,L+8,P+8)
680 FOR I=12 TO 17 :: CALL SPRITE(#I,46,16,L+7,P+7,INT(
    RND*10-5),INT(RND*10-5)):: NEXT I
690 Q1=INT(RND*3+1):: CALL SOUND(300,-6,0):: CALL SOUND
    (1000,-4,20):: CALL MOTION(#5,-Q1,-2,#6,-2,Q1,#7,Q1
    ,-2,#8,2,Q1)
700 FOR I=9 TO 15 :: CALL SPRITE(#I,44,16,L+6,P+4,INT(R
    ND*8-4),INT(RND*8-4)):: NEXT I
710 CALL SOUND(4250,-8,22):: FOR I=1 TO 4 :: CALL POSIT
    ION(#I+4,Q1,Q2)
720 CALL DELSPRITE(#I+4):: FOR W1=1 TO 5 :: CALL SPRITE
    (#I+9+W1,44,9,Q1,Q2,INT(RND*4-2),INT(RND*4-2)):: NE
```

```
    XT W1 :: NEXT I
730 FOR D=1 TO 350 :: NEXT D :: FOR I=5 TO 27::CALL DEL
    SPRITE(#I):: NEXT I :: CALL SHIP :: SUBEND
740 SUB SHIP :: CALL MAGNIFY(3):: RANDOMIZE :: CALL SPR
    ITE(#5,96,15,INT(RND*100+1),INT(RND*200+1)):: SUBEN
    D
750 SUB STARS(F):: RANDOMIZE :: FOR I=1 TO F :: CALL VC
    HAR(INT(RND*21+3),INT(RND*30+2),46):: NEXT I :: SUB
    END
760 SUB FIRE :: CALL SPRITE(#1,136,16,192,8,#2,140,16,1
    92,256):: CALL MOTION(#1,-90,106,#2,-90,-106)
770 CALL COINC(#1,#2,100,H):: IF H THEN CALL DELSPRITE(
    #1,#2)ELSE 770
780 CALL SOUND(100,-7,20):: SUBEND
790 SUB DELAY(D):: FOR I=1 TO D :: NEXT I :: CALL CLEAR
    :: SUBEND
800 SUB WIN :: FOR I=1 TO 15 :: CALL SCREEN(I):: CALL S
    OUND(-200-I,200+I,I):: NEXT I :: SUBEND
```

CHAPTER 13

Gunner

(EXTENDED BASIC)

Introduction

This is the story of the *Gunner*. From outer space, 27 spacecraft came flying, one at a time. They were approaching the earth in a suspicious manner. When their arrival was detected, the Gunner was alerted and ordered to follow the path of the craft. At a distance, the spacecraft appeared as small points on the horizon, which grew bigger and bigger as they approached. They glided, changed direction, swirled around in order to evade our defense border. The Gunner was ordered to fire. . . . Whew! That was too close. A shielding net was dropped to repulse the spacecraft when the firing range was too close to be effective for the Gunner. The Gunner with his shields was considered to be the most effective defender of the earth, in the years before the space war began.

Before the victory can be claimed and before triumphal music can play at the end of the space battle, 27 spacecraft have to be shot down. So, pay attention and be prepared to defend our home planet.

Features

1. The program is designed to illustrate a scene from outer space, with vivid colors, graphics, and sounds.
2. A joystick is required to play the game. It is necessary for maneuvering of the gunsight in each of its 8 directional movements.
3. The shield is an unpenetrable net used to repulse the spacecraft at the right time (when it appears at its largest). A full screen shield is used here for the most urgent situation only.
4. The spacecraft are designed to move faster and faster when

getting close to the earth. They also change direction continually, and swirl and glide in order to evade the defense Gunner.

5. A display of the score and a count of the remaining shields are also displayed. Shield time is limited for each use. Extra shields are generated as a bonus for every two spacecraft shot down.
6. A final message to judge your defense skill is given each time the game ends.

How To Play, Step-By-Step

1. Run the program.
2. Use the joystick to move the Gunner's sight and to search for the spacecraft. When far away, the craft is small, but slow and easy to shoot. When it comes closer, it is more active and faster. It changes directions all the time and is harder to shoot at. So, do not wait until it is too close.
3. When too close to shoot and your gunsight disappears, the spacecraft gets to its largest form, and you should use a shield (by pressing any key on the keyboard) to prevent the craft from penetrating your defense.
4. For every 2 spacecraft you shoot down, an extra shield is added for your defense.
5. The game ends when 3 spacecraft have slipped through the Gunner's defense or when you have destroyed all 27 of them. Final messages will appear on the screen to tell you how well you have defended the earth.

Program Explanation

0010-0040	Program name.
0050-0180	Redefine characters and define colors.
0190-0280	Play music, set stars on screen, draw invisible shield (to be called up with call color when shield is used).
0290	Start. Set variables.
0340-0350	Check for player input.
0360-0390	FIRE! Check for coincidence and GOSUB explosion routine if hit.
0400-0410	Move sight.

0420-0470	Change spacecraft size and check if craft has escaped (grown to full size).
0480-0530	Move spacecraft randomly.
0540	Check joystick and move sight.
0550	Shield.
0560-0580	Put shield on and check timer of shield.
0590-0640	Spacecraft hits shield and bounds back.
0650-0760	Spacecraft hit; explosion subroutine.
0770-0950	End message, song, and ask for replay.

GUNNER
[EXTENDED BASIC]

```
10 !//////////
20 !/ GUNNER /
30 !//////////
40 !
50 !///INITIALIZE///
60 DATA 0103010107094 8C1FD4808060000030180C000006010128
   F831290E08080C08
70 DATA 0103000006084 8C0C04808060000030180C000006010120
   303121060000 0C08
80 DATA 0000000000000001000000000000000000000000000008
90 DATA 0000000000000001070000000000000000000000000008
   0E
100 DATA 000000000000010 31E0000000000000000000000000000
    80F
110 DATA 000000000000010 1070CFF0000000000000000000000000
    00C060FE
120 DATA 00000000201 00E0D04020100000000000000000000040
    808040C0C0201
130 DATA 0000000000000201010203030408000000000000040870
    80204080
140 FOR I=96 TO 124 STEP 4 :: READ A$ :: CALL CHAR(I,A$
    ):: NEXT I
150 RANDOMIZE :: C=0 :: L=4 :: CALL CLEAR
160 CALL CHAR(136,"0102040810204080804020100804020 1")!S
    HIELD
170 CALL CHAR(36,"000010181C3830000000 10684C5C381800000
    8183070F8CC0080783CC6")!CRASH
180 CALL COLOR(10,1,1,14,1,1):: FOR I=1 TO 8 :: CALL CO
    LOR(I,16,1):: NEXT I
190 RESTORE 790 :: FOR I=1 TO 18 :: READ X,Y :: CALL SO
    UND(X,Y,0,110,5):: NEXT I
200 !///START///
210 CALL INIT :: CALL LOAD(-31878,2):: CALL SCREEN(2)::
    Q,SC=0 :: SH=1
220 !//STARS//
230 FOR I=1 TO 20 :: CALL HCHAR(INT(RND*20+2),INT(RND*2
    8+4),104,3):: CALL VCHAR(INT(RND*20+2),INT(RND*28+4
    ),106,3):: NEXT I
```

```
240 !//DRAW SHIELD//
250 FOR I=1 TO 21 STEP 4 :: FOR J=2 TO 30 STEP 4
260 CALL VCHAR(I,J,136):: CALL VCHAR(I,J+1,137):: CALL
    VCHAR(I+1,J-1,136):: CALL VCHAR(I+1,J+2,137):: CALL
    VCHAR(I+2,J-1,137)
270 CALL VCHAR(I+2,J+2,136):: CALL VCHAR(I+3,J,137):: C
    ALL VCHAR(I+3,J+1,136)
280 NEXT J :: NEXT I :: CALL COLOR(10,16,1)
290 !///GAME LOOP///
300 CALL MAGNIFY(3):: K=88 :: S=122
310 IF SC>26 OR Q>2 THEN 780
320 CALL KEY(0,I,J):: IF I=83 THEN DISPLAY AT(24,5)BEEP
    :"CREATED BY KHOA TON" :: DISPLAY AT(24,1): :: GOTO
    320
330 U,F,D,E=0 :: CALL SPRITE(#1,100,6,K,S,#2,104,7,INT(
    RND*145+30),INT(RND*200+30),RND*3-1,RND*3-1):: IF
    SC>1+C THEN L=L+2 :: C=SC
340 !//PLAYER INPUT//
350 CALL JOYST(1,Y,X):: CALL KEY(1,K,S):: IF S=0 THEN 4
    10 ELSE IF K<>18 THEN 560
360 !//FIRE//
370 CALL SOUND(-400,330,5,-5,3):: CALL PATTERN(#1,96)::
    CALL SOUND(-200,110,5,-5 ,3)
380 CALL COINC(#1,#2,D,HIT):: IF HIT THEN GOSUB 660 ::
    GOTO 310
390 CALL PATTERN(#1,100)
400 !//MOVE SIGHT//
410 CALL MOTION(#1,-2*X,2*Y):: GOSUB 430 :: IF F=2 THEN
    CALL MOTION(#2,0,0):: GOTO 300 ELSE 350
420 !//ALIEN SIZE//
430 IF D<4 THEN CALL PATTERN(#2,D*4+104):: GOTO 540
440 IF D<7 THEN 490 ELSE IF F=0 THEN CALL POSITION(#2,I
    ,J):: F=SGN(J-122):: CALL PATTERN(#2,F*2+122):: CAL
    L MOTION(#2,0,F*3):: RETURN
450 IF E>6400 THEN 540 ELSE CALL DELSPRITE(#1):: U=2 ::
    CALL MAGNIFY(4):: CALL MOTION(#2,0,F*15):: CALL POS
    ITION(#2,J,I):: IF I>16 AND I<240 THEN 470 ELSE CAL
    L DELSPRITE(#2):: F=2 :: Q=Q+1
460 FOR I=500 TO 2000 STEP 50 :: CALL SOUND(-300,I,0)::
    NEXT I
470 RETURN
480 !//ALIEN MOVE & PLAYER INPUT//
490 I=INT(RND*12):: IF I>3 THEN 540 ELSE ON I+1 GOTO 50
    0,510,520,530
500 CALL PATTERN(#2,116):: CALL MOTION(#2,INT(RND*5-2),
    INT(RND*5-2)):: GOTO 540
510 CALL PATTERN(#2,120):: CALL MOTION(#2,INT(RND*9-4),
    INT(RND*4+4)):: GOTO 540
520 CALL PATTERN(#2,124):: CALL MOTION(#2,INT(RND*9-4),
    -INT(RND*4+4)):: GOTO 540
530 CALL PATTERN(#2,124):: CALL POSITION(#2,I,J):: CALL
    MOTION(#2,SGN(88-I)*4,SGN(122-J)*5):: RANDOMIZE
540 CALL JOYST(1,X,Y):: CALL MOTION(#1,-2*Y,2*X):: E=E-
    L*15 :: D=0-INT(E/1000):: RETURN
```

```
550 !//SHIELD//
560 IF SH<.9 THEN 410 ELSE CALL COLOR(14,6,1):: J=0
570 J=J+1 :: CALL SOUND(-999,3999,12,-6,22):: CALL KEY(
    1,K,S):: GOSUB 430 :: IF U<>2 THEN IF S=0 OR J>5 TH
    EN 640 ELSE 570
580 IF F=2 THEN CALL COLOR(14,1,1):: GOTO 300
590 !/HIT AND BOUNCE/
600 CALL SOUND(800,110,0,-5,0,330,0,220,0):: CALL MOTIO
    N(#2,RN*5-2,F*-7):: FOR I=124 TO 116 STEP -4
610 CALL SOUND(-300,440,I/4-20,110,15,-1,I/4-20):: CALL
    PATTERN(#2,I):: NEXT I
620 CALL MAGNIFY(3):: FOR I=124 TO 104 STEP -4 :: CALL
    SOUND(-300,330,I/4-5,110,15,-1,I/4-5):: CALL PATTER
    N(#2,I):: NEXT I
630 U,E,F,D=0 :: CALL SPRITE(#1,100,6,88,122)
640 CALL COLOR(14,1,1):: SH=SH-1 :: DISPLAY AT(1,17):"S
    HIELD:";STR$(INT(SH)):: GOTO 410
650 !///EXPLODE///
660 CALL MOTION(#1,0,0,#2,0,0):: CALL SOUND(-1999,110,0
    ,-7,0):: CALL POSITION(#2,I,J,#1,K,S):: CALL MAGNIF
    Y(1)
670 CALL LOAD(-31878,8)
680 IF S>248 THEN S=248
690 IF K>184 THEN K=184
700 CALL SPRITE(#1,96,6,K,S,#7,98,6,K,S+8,#8,97,6,K+8,S
    ,#6,99,6,K+8,S+8)
710 CALL SPRITE(#2,36,7,I,J,#3,37,7,I,J,#4,38,7,I,J,#5,
    39,7,I,J)
720 RANDOMIZE :: CALL MOTION(#2,RND*5-2,RND*5-2,#3,RND*
    5-2,RND*5-2,#4,RND*5-2,RND*5-2,#5,RND*5-2,RND*5-2)
730 FOR I=1 TO 8 :: CALL SCREEN(7):: CALL SOUND(-400,11
    0,3,-6,3):: CALL SCREEN(2):: NEXT I
740 FOR I=1 TO 30 :: CALL SOUND(-300,-7,I):: NEXT I ::
    CALL DELSPRITE(#2,#3,#4,#5,#6,#7,#8):: CALL MAGNIFY
    (3)
750 SC=SC+1 :: SH=SH+.5 :: DISPLAY AT(1,1):"SCORE:";STR
    $(SC);TAB(17);"SHIELD:";STR$(INT(SH))
760 CALL LOAD(-31878,2):: RETURN
770 !///END MESSAGES///
780 CALL SCREEN(2):: A$=RPT$(" ",27)&"MESSAGE FROM STAR
    FLEET COMMAMD..." :: IF Q=3 THEN 850
790 DATA 500,349,430,523,150,466,150,440,150,392,230,34
    9,650,698,600,523,150,466,150,440,150,392,150,349
800 DATA 650,698,600,523,200,466,200,440,200,466,700,39
    2,1,44733
810 FOR I=1 TO 9 :: CALL COLOR(I,8,1):: NEXT I
820 RESTORE 790 :: FOR I=1 TO 18 :: READ X,Y :: CALL SO
    UND(X,Y,0,110,5):: NEXT I
830 M$=A$&"YOU HAVE SINGLE HANDEDLY DESTROYED THE WHOLE
    ATTACKING ZYLON FORCE AND ARE HEREBY PROMOTED TO FL
    EET COMMANDER. "
840 GOTO 920
850 IF SC>5 THEN 870
```

```
860 M$=A$&"YOU ARE A DISGRACE!!!       REPORT BACK TO BA
    SE AT ONCE. " :: GOTO 920
870 IF SC>10 THEN 890
880 M$=A$&"YOUR BATTLE SKILLS HAVE TO BE IMPROVED...  R
    EPORT TO BASE FOR TRAINING. " :: GOTO 920
890 IF SC>14 THEN 910
900 M$=A$&"NOT BAD SHOOTING CAPTAIN.   ONLY A FEW ZYLON
    S ESCAPED! " :: GOTO 920
910 M$=A$&"VERY GOOD FIGHTING CAPTAIN.   YOU HAVE DESTR
    OYED MOST OF THE ZYLON ATTACK FORCE.  REPORT TO BAS
    E. "
920 FOR I=1 TO 50 :: CALL SOUND(30,2990,0):: NEXT I
930 FOR I=1 TO LEN(M$):: DISPLAY AT(10,1)BEEP:SEG$(M$,I
    ,28):: NEXT I
940 DISPLAY AT(19,5):"BACK TO STOP":TAB(5);"REDO TO GO
    ON PATROL"
950 CALL KEY(0,K,S):: IF K=6 THEN 150 ELSE IF K=15 THEN
    CALL CLEAR.ELSE 950
```

CHAPTER 14

Space Battle

(EXTENDED BASIC)

Introduction

It is the start of an intergalactic war. Each side sends a squadron of 8 spaceships to attack the opponent. One ship at a time, from each side, confronts the other. Each spaceship is equipped with sufficient energy to fire 5 laser bolts. You are the Commander pilot of your squadron and your opponent is the Commander of his. Each side must use their skill, reflexes, and cunning to outmaneuver the other and to fire on him with the laser. If you have used up your laser energy without hitting your opponent or vice versa, you will have no chance to fire on him again until he has exhausted his laser bolts; additional energy will then be generated for your ship and your opponent's craft to continue the fight again.

The game required the use of joysticks for both sides.

Features

1. This is a competitive game of skill and accuracy for two players. Joysticks are required for both sides.
2. Graphics, colors, and sounds are the main features of the game.
3. The game allows the use of your joysticks' 8 directional movements to move your spacecraft, in order to chase and fire on your opponent.
4. The program simulates the effect of an explosion when a spaceship is hit by a laser bolt. This animates the battle scene.
5. Two counters are reserved to display the number of remaining ships and the number of remaining laser bolts available for each player.
6. Messages are programmed in the game to appear at the end of the game to congratulate the winner.

How To Play, Step-By-Step

1. Run the program. No instructions are written for this program due to the fact that it is so simple to play.
2. Use joystick number 1 for the left player (green spaceship) and joystick number 2 for the right player (blue spaceship).
3. Maneuver your spaceship close to the enemy. When you come into effective firing range, aim your spaceship in the direction of the enemy and fire by pressing the button on your joystick. You will learn to know the firing range of the laser after playing with it for a while.
4. The opponent does the same to fight you.
5. The battle is over when either player loses all the ships in his fleet.

Program Explanation

0010-0040	Program name.
0050-0230	Initialize arrays, shapes, and colors.
0240	Start. Set ship supply to 8 for each team.
0260-0270	Put 40 stars randomly on screen.
0280-0320	Display the playfield and make sprites, set laser counter to 5 for each ship, and set X,Y velocities.
0330-0350	First player's move. If shoot, then GOSUB 400 (shoot).
0360-0380	Second player's move. If shoot, then GOSUB 400 (shoot).
0390-0410	One-player-shoot subroutine. Subtract and display ammunition; stop all ships.
0420	Call sprite number 3 as first player's laser bolt.
0430	Call sprite number 3 as second player's laser bolt.
0440-0460	Let laser bolt fly for 40 count or until it hits opponent.
0470-0500	Laser hit! Explosion.
0510-0520	Player 1 was hit! Subtract ship, return (to line 340); if lost all ships, then set @ to 2.
0530-0540	Player 2 was hit! Subtract ship, return (to line 370); if lost all ships, then set @ to 2.
0550-0600	End messages.
0560	Play again? If yes, then go to line 250 (start).

SPACE BATTLE

[EXTENDED BASIC]

```
10 !/////////////////
20 !/ SPACE BATTLE /
30 !/////////////////
40 !
50 !///INITIALIZE///
60 !//ARRAYS//
70 DIM BU(2),DSH(2,2),BD(2,2)
75 CALL INIT :: CALL LOAD(-31878,3)
80 DB(1,2)=106 :: DSH(1,2)=96
90 DB(2,2)=108 :: DSH(2,2)=100
100 DB(2,1)=107 :: DSH(2,1)=98
110 DB(2,0)=109 :: DSH(2,0)=102
120 DB(1,0)=106 :: DSH(1,0)=97
130 DB(0,0)=108 :: DSH(0,0)=103
140 DB(0,1)=107 :: DSH(0,1)=99
150 DB(0,2)=109 :: DSH(0,2)=101
160 !//SHAPES//
170 DATA 0808081C1C3E2A22,222A3E1C1C080808,0000E0387F38
    E,0000071CFE1C07
180 DATA 010214789C28081,8040281E39141008,1008289C78140
    201,081014391E28408
190 DATA 12343EFC7F2F4A9,10195F2A37CC331,0010101010101,
    000000007E,0000060C183,000030180C06
200 CALL CHAR(46,"0000001"):: CALL CHAR(33,RPT$("F",16)
    )
210 FOR I=1 TO 9 :: CALL COLOR(I,16,1):: NEXT I
220 FOR I=96 TO 109 :: READ R$ :: CALL CHAR(I,R$):: NEX
    T I
230 !///START///
240 CALL CLEAR :: SH1,SH2=8
250 !//STARS//
260 FOR I=1 TO 40 :: CALL VCHAR(RND*22+2,RND*30+2,46)::
    NEXT I
270 !///1ST SCREEN///
280 CALL SCREEN(2):: DISPLAY AT(1,1):RPT$(CHR$(98),SH1)
    :: DISPLAY AT(1,21):RPT$(CHR$(99),SH2)
290 CALL SPRITE(#1,98,3,96,24,#2,99,5,96,232)
300 DISPLAY AT(2,1):"!!!!!!";TAB(21);"!!!!!!"
310 BU(1),BU(2)=5 :: ∂=0 :: X1,X2=0 :: Y1=4 :: Y2=-4
320 !//PLAYER 1//
330 CALL JOYST(1,Y,X):: CALL KEY(1,K1,S1):: IF K1=18 TH
    EN A=1 :: B=2 :: GOSUB 390 :: IF ∂=1 THEN 280 ELSE
    IF ∂=2 THEN 550 ELSE 360 ELSE IF X=0 AND Y=0 THEN 3
    60
340 X1=X :: Y1=Y :: CALL PATTERN(#1,DSH(SGN(Y1)+1,SGN(X
    1)+1)):: CALL MOTION(#1,-X1,Y1)
350 !//PLAYER 2//
360 CALL JOYST(2,Y,X):: CALL KEY(2,K2,S2):: IF K2=18 TH
    EN A=2 :: B=1 :: GOSUB 390 :: IF ∂=1 THEN 280 ELSE
    IF ∂=2 THEN 550 ELSE 330 ELSE IF X=0 AND Y=0 THEN 3
    30
```

```
370 X2=X :: Y2=Y :: CALL PATTERN(#2,DSH(SGN(Y2)+1,SGN(X
    2)+1)):: CALL MOTION(#2,-X2,Y2):: IF BU(1)=0 AND BU
    (2)=0 THEN 280 ELSE 330
380 !//FIRE//
390 I=0 :: IF BU(A)=0 THEN RETURN ELSE BU(A)=BU(A)-1 ::
    CALL MOTION(#A,0,0,#B,0,0):: CALL SOUND(100,220,0):
    : CALL POSITION(#A,X,Y)
400 DISPLAY AT(2,1):RPT$("!",BU(1));TAB(21);RPT$("!",BU
    (2)):: IF A=2 THEN 420
410 CALL SPRITE(#3,DB(SGN(Y1)+1,SGN(X1)+1),11,X,Y,-X1*3
    ,Y1*3):: GOTO 440
420 CALL SPRITE(#3,DB(SGN(Y2)+1,SGN(X2)+1),11,X,Y,-X2*3
    ,Y2*3)
430 !//FLYING LASER BOLT//
440 I=I+1 :: CALL SOUND(200,999,9):: CALL COINC(#B,#3,5
    ,HIT):: IF HIT THEN 470 ELSE IF I<10 THEN 440
450 CALL DELSPRITE(#3):: RETURN
460 !//EXPLODE//
470 CALL DELSPRITE(#3):: CALL SOUND(300,-5,0):: CALL PA
    TTERN(#B,104):: CALL COLOR(#B,16):: CALL SOUND(400,
    -7,0):: CALL PATTERN(#B,105):: CALL SOUND(-400,-6,0
    )
480 A=1 :: FOR X=1 TO 300 :: NEXT X :: CALL DELSPRITE(#
    B)
490 X1,X2=4 :: IF A=2 THEN 530
500 !/PLAYER 1 HIT/
510 CALL COLOR(#2,5):: SH2=SH2-1 :: IF SH2<1 THEN A=2 :
    : RETURN ELSE RETURN
520 !/PLAYER 2 HIT/
530 CALL COLOR(#1,11):: SH1=SH1-1 :: IF SH1<1 THEN A=2
    :: RETURN ELSE RETURN
540 !///END MESSAGE///
550 IF ABS(SH1-SH2)>3 THEN A$="SUPERIOR TO" ELSE A$="BE
    TTER THAN"
560 IF SH1=0 THEN B$="BLUE" :: C$="GREEN" ELSE B$="GREE
    N" :: C$="BLUE"
570 DISPLAY AT(5,3):"THE BATTLE IS OVER": :"THE ";B$;"
    FORCE IS ": :A$: :"THE "; C$
580 DISPLAY AT(23,1):"DO YOU WISH ANOTHER BATTLE?" :: D
    ISPLAY AT(1,1):RPT$(CHR$(98),SH1):: DISPLAY AT(1,21)
    :RPT$(CHR$(99),SH2)
590 CALL KEY(0,K,S):: IF K=78 THEN CALL CLEAR ELSE IF K
    =89 THEN 240 ELSE 590
```

CHAPTER 15

Auto Sprite Definition

(EXTENDED BASIC)

Introduction

To every game programmer, graphics are essential. In our opinion, with the TI computer, graphics programming is greatly simplified for the programmer by the redefining of characters. Simple though it may be, defining pictures is a tedious task (as you have probably found out when you programmed your own graphic games or typed in the many games in this book). One incorrect hexadecimal digit can turn a frog into a lilypad.

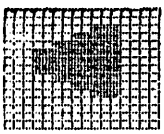
This program will relieve you from the complicated coding of sprites and will help you put the data code in your program (if you have a disk system). People who have the TI 99/4 Impact Printer can also get a picture printed out on paper. If you do not have a disk or printer, you will have to use paper and pencil to note down the code of the sprite you drew.

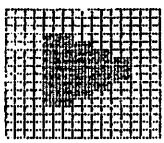
Features

1. Colors, graphics, sounds, and illustrations are the main features of the program.
2. The program also provides multiscreen options that can be used to match the artistic creation of the programmer.
3. A moving sprite is used to facilitate your design.
4. Keyboard control, with 8 directional cursor movements, is used with this program. No joystick required.
5. The picture option printout requires use of a TI 99/4 Impact Printer.
6. Data for the picture can be saved on disk (disk system required, of course).
7. When data is saved on disk, the data can be merged into your program.

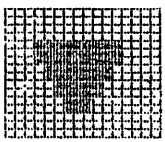
How To Use, Step-By-Step

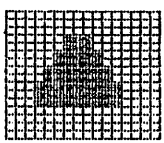
1. Run the program.
2. Use arrow keys (E, X, S, and D) for up, down, left, and right and use (W, R, C, and Z) for the diagonal movements of the cursor when drawing your graphical picture.
3. Press Q to change cursor color (thus leaving a trace of black or white).
4. If you are satisfied with your picture, go to Step 7.
5. The main screen options are:
   A. **WHITE.** Defines the color of your workspace to white (off).
   B. **BLACK.** Defines the color of your workspace to black (on).
   C. **OPTIONS.** These are more options. Go to Step 6.
   D. **STOP.** End the program.
   E. **DATA.** Pùt a picture of the code (on line 150 in the program) on the workspace. If you want a picture that you have saved on disk placed on the workspace, follow these steps.
      (1) Get the Auto Sprite Definition program into memory.
      (2) Type: MERGE"DSK1.'name of your picture' "
      (3) Find the data line of that picture in the program.
      (4) Move that line to line 150 in the program.
      (5) Run the program.
      (6) Press 5 for DATA (the picture should appear in the workspace).
6. Some additional options include:
   A. V MIRROR changes the drawing in the workspace as if there was a mirror placed vertically on one side.



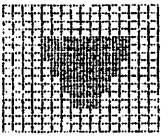


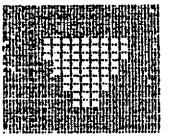
   B. H MIRROR changes the drawing in the wordspace as if there was a mirror placed horizontally either at the top or the bottom.



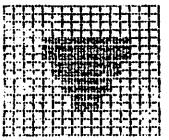


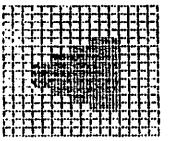
C. REVERSE reverses the color of the drawing.





D. ROTATE rotates the picture clockwise.





E. SCOOT scoots or moves the picture over one space in any of these directions: right, left, up, or down.

F. RETURN returns to the main screen.

7. When your picture is finished, press SHIFT on the TI 99/4 console or B on the TI 99/4A console.
8. Sprite will be defined. Code will appear at the bottom of the screen (copy this code down if you do not have a disk drive to record data).
9. Press R to record to disk; otherwise, press any other key.
10. If you pressed R, you will be asked:

NAME and LINE

For example:

NAME—CAT
LINE—1

(When you merge CAT, the data will be on line 1.) It is best to use a line number less than 100, so that when you want to merge your picture into Auto Sprite Definition, you won't destroy any of its lines. If you made a mistake when typing in NAME, enter 999 for LINE.

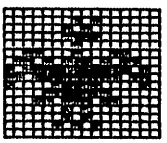
11. After you are finished with the recording, you will be asked: PRINT TO PRINTER? Press Enter if you do not want to print it or Y if you want to print your picture on paper.
12. Now, you will be returned to Step 2.

Program Explanation

0100-0130	Program name.
0140-0160	Merge data at line 150.
0170-0200	Initialize characters and colors.
0210-0260	Display workscreen.
0270-0330	Check keys and branch.
0340-0370	Display Option screen and check for keys and branch.
0380-0400	Display Scoot option screen and check for keys and branch.
0410-0480	Scoot up, down, right, and left options.
0490-0500	Put picture in SP(16,16) array for following options.
0510-0530	Mirror (vertical and horizontal) options.
0540-0550	Rotate clockwise option.
0560-0570	Most options branch here to put cursor back on screen; then go to line 260 (main loop).
0580-0600	Reverse option.
0610-0640	If arrow keys were pressed, branch here to move cursor; then go back to main loop.
0650-0680	Get definition of picture (64 hexadecimal characters) and display on screen.
0690-0730	Record to disk? If Yes, then save as merge file with line numbers 1–254 with input name.
0740-0780	Print picture to printer? If no, then go back to main loop (line 570). If yes, then GOSUB to use printer (line 930); then line 570.
0790-0840	Hexadecimal conversion subroutine.
0850-0920	Get merged file (at line 150) and convert to picture on screen subroutine.
0930-1040	Print to printer subroutine.

Printer Output

The following is an example of an output from the TI 99/4 Impact Printer:



```
NAME : PHOENIX

LINE : 1

HEX : 0001020301316BBFDF870F1B020205000040
      80C0001BACFAF6C2E0B080804000
```

AUTO SPRITE DEFINITION

[EXTENDED BASIC]

```
100 !//////////////////
110 !/ AUTO SPRITEDEF /
120 !//////////////////
130 !
140 !<<MERGE DATA BEFORE>>
150 DATA 0005020701316BBFDF870F1B0202050000008080001 8ACFAF6C2E0B080804000
160 !<<OR ON LINE 150>>
170 !///INITIALIZE///
180 OPTION BASE 1 :: DIM SP$(16),B(8,8),SP(16,16):: T=20 :: K,C=100 :: LX,LY=5 :: CALL INIT :: CALL LOAD(-31878,1)
190 CALL CHAR(100,RPT$("0",16)&RPT$("F",16)&"007E424242427E00FF81BDBDBDBD81FF")
200 HEX$="0123456789ABCDEF" :: CALL COLOR(9,2,16)
210 !///SCREEN///
220 CALL MAGNIFY(4):: CALL SCREEN(12):: CALL CLEAR :: CALL SPRITE(#1,96,9,10,10,0,4)
230 DISPLAY AT(1,4):"AUTO SPRITE DEFINITION": : :"  1234567812345678"
240 FOR I=0 TO 1 :: FOR I1=1 TO 8 :: CALL VCHAR(4+I1+8*I,4,48+I1):: NEXT I1 :: NEXT I
250 FOR I=5 TO 20 :: CALL HCHAR(I,5,K,16):: NEXT I :: C=K :: CALL VCHAR(LX,LY,C+2)
260 CALL SCREEN(12):: DISPLAY AT(5,22):"PRESS " :: DISPLAY AT(7,T):"1 WHITE" :: DISPLAY AT(9,T):"2 BLACK" :: DISPLAY AT(11,T):"3 OPTIONS" :: DISPLAY AT(13,T):"4 STOP" :: DISPLAY AT(17,T): :: DISPLAY AT(15,T):"5 DATA"
270 !//KEY INPUT//
280 CALL KEY(1,K,S):: IF S=0 THEN 280 ELSE X,Y=0 :: IF K=5 THEN X=-1 :: GOTO 620 ELSE IF K+1=1 THEN X=1 :: GOTO 620 ELSE IF K=2 THEN Y=-1 :: GOTO 620
290 IF K=3 THEN Y=1 :: GOTO 620 ELSE IF K=4 THEN X,Y=-1 :: GOTO 620 ELSE IF K=14 THEN X,Y=1 :: GOTO 620 ELSE IF K=15 THEN X=1 :: Y=-1 :: GOTO 620
300 IF K=6 THEN X=-1 :: Y=1 :: GOTO 620 ELSE IF K<>18 THEN 320 ELSE CALL SOUND(100,110,0,-6,0):: IF C=101 THEN C=100 ELSE C=101
310 CALL VCHAR(LX,LY,C+2):: GOTO 280
320 CALL SOUND(100,660,0):: IF K=19 THEN K=100 :: GOTO 250 ELSE IF K=16 THEN 660 ELSE IF K=7 THEN K=101 :: GOTO 250
330 IF K=9 THEN STOP ELSE IF K=8 THEN 350 ELSE IF K<>10 THEN 280 ELSE GOSUB 880 :: GOTO 570
340 !///OPTIONS///
350 CALL SCREEN(10):: DISPLAY AT(7,T):"1 VMIRROR" :: DISPLAY AT(9,T):"2 HMIRROR":: DISPLAY AT(11,T):"3 REVERSE" :: DISPLAY AT(13,T):"4 ROTATE" :: DISPLAY AT(15,T):"5 SCOOT" :: DISPLAY AT(17,T):"6 RETURN"
360 CALL KEY(0,K,S):: IF S=0 OR(K<49 OR K>54)THEN 360 E
```

```
    LSE CALL SOUND(100,330,0):: H=K-48 :: CALL VCHAR(LX
    ,LY,C):: CALL VCHAR(H*2+5,23,42)
370 IF H=6 THEN 570 ELSE IF H=3 THEN 590 ELSE IF H<>5 T
    HEN 500
380 !//SCOOT//
390 CALL SCREEN(6):: DISPLAY AT(7,T):"1 UP" :: DISPLAY
    AT(9,T):"2 DOWN" :: DISPLAY AT(11,T):"3 LEFT" :: DI
    SPLAY AT(13,T):"4 RIGHT" :: DISPLAY AT(15,T): :: DI
    SPLAY AT(17,T):
400 CALL KEY(0,K,S):: IF S=0 OR(K<49 OR K>52)THEN 400 E
    LSE CALL VCHAR((K-49)*2+7,23,42)
410 !/UP,DOWN/
420 CALL SOUND(100,110,0):: IF K>50 THEN 460 ELSE IF K=
    50 THEN K=20 :: S=-1 ELSE K=5 :: S=1
430 FOR I=K TO S*15+K STEP S :: FOR J=5 TO 20 :: CALL G
    CHAR(I+S,J,H):: IF H<>100 AND H<>101 THEN H=100
440 CALL VCHAR(I,J,H):: NEXT J :: NEXT I :: GOTO 570
450 !/RIGHT,LEFT/
460 IF K=51 THEN K=5 :: S=1 ELSE K=20 :: S=-1
470 FOR J=K TO S*15+K STEP S :: FOR I=5 TO 20 :: CALL G
    CHAR(I,J+S,H):: IF H<>100 AND H<>101 THEN H=100
480 CALL VCHAR(I,J,H):: NEXT I :: NEXT J :: GOTO 570
490 !//GET PICTURE//
500 FOR I=1 TO 16 :: FOR J=1 TO 16 :: CALL GCHAR(I+4,J+
    4,SP(I,J)):: NEXT J :: CALL VCHAR(I+4,21,42):: NEXT
    I :: CALL VCHAR(5,21,32,16)
510 !//MIRRORS//
520 IF H=1 THEN K=4 :: S=-21 ELSE IF H=2 THEN K=-21 ::
    S=4 ELSE 550
530 FOR I=1 TO 16 :: FOR J=1 TO 16 :: CALL VCHAR(ABS(K+
    I),ABS(S+J),SP(I,J)):: NEXT J :: NEXT I :: GOTO 570
540 !//ROTATE//
550 FOR J=20 TO 5 STEP -1 :: FOR I=5 TO 20 :: CALL VCHA
    R(I,J,SP(21-J,I-4)):: NEXT I :: NEXT J
560 !//PUT BACK CURSOR//
570 CALL GCHAR(LX,LY,C):: CALL VCHAR(LX,LY,C+2):: GOTO
    260
580 !//REVERSE//
590 FOR I=1 TO 16 :: FOR J=1 TO 16 :: CALL GCHAR(I+4,J+
    4,H):: IF H=100 THEN H=101 ELSE H=100
600 CALL VCHAR(I+4,J+4,H):: NEXT J :: NEXT I :: GOTO 57
    0
610 !//MOVE CURSOR//
620 IF LX+X=21 THEN X=-15 ELSE IF LX+X=4 THEN X=15
630 IF LY+Y=21 THEN Y=-15 ELSE IF LY+Y=4 THEN Y=15
640 CALL VCHAR(LX,LY,C):: LX=LX+X :: LY=LY+Y :: CALL VC
    HAR(LX,LY,C+2):: GOTO 280
650 !///GET SPRITEDEF///
660 CALL VCHAR(LX,LY,C):: CALL SOUND(100,128,0):: CALL
    SOUND(300,912,0)
670 FOR Q=5 TO 13 STEP 8 :: FOR W=5 TO 13 STEP 8 :: FOR
    E=0 TO 7 :: FOR R=0 TO 7 :: CALL GCHAR(W+E,Q+R,CH)::
    B(E+1,R+1)=CH-100
```

```
680 NEXT R :: NEXT E :: GOSUB 800 :: DISPLAY AT(22,1):M
    $ :: NEXT W :: NEXT Q
690 !//RECORD TO DISK?//
700 CALL CHAR(96,M$):: DISPLAY AT(7,T)BEEP:"R RECORD" :
    : FOR I=9 TO 15 STEP 2 :: DISPLAY AT(I,T): :: NEXT I
710 A$=""
720 CALL KEY(0,KEY,ST):: IF ST=0 THEN 720 ELSE IF KEY<>
    82 THEN 770
730 DISPLAY AT(17,T):" NAME?" :: DISPLAY AT(19,T):"LINE
    " :: ACCEPT AT(18,T)BEEP: NA$ :: ACCEPT AT(19,25)BE
    EP VALIDATE(DIGIT)SIZE(3):LI :: IF LI>254 THEN 730
740 !//SAVE AS MERGE//
750 OPEN #1:"DSK1."&NA$,VARIABLE 163 :: PRINT #1:CHR$(0
    )&CHR$(LI)&CHR$(147)&CHR$(200)&CHR$(LEN(M$))&M$&CHR
    $(0):: PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1
760 !//IMPACT PRINTER?//
770 DISPLAY AT(7,T):"PRINT ON" :: DISPLAY AT(9,T):"PRIN
    TER?N" :: ACCEPT AT(9,T+8)BEEP SIZE(-1):A$ :: IF A$
    ="N" THEN M$="" :: GOTO 570
780 GOSUB 950 :: M$="" :: GOTO 570
790 !///HEX SUBR//
800 FOR R=1 TO 8
810 LOW=B(R,5)*8+B(R,6)*4+B(R,7)*2+B(R,8)+1
820 HIGH=B(R,1)*8+B(R,2)*4+B(R,3)*2+B(R,4)+1
830 M$=M$&SEG$(HEX$,HIGH,1)&SEG$(HEX$,LOW,1)
840 NEXT R :: RETURN
850 !///FIGURE FROM DATA///
860 DATA 0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,1,0,1,0,0,0,1,0,
    1,0,1,1,0,0,1,1,1,1,0,0,0
870 DATA 1,0,0,1,1,0,1,0,1,0,1,1,1,1,1,0,0,1,1,0,1,1,1,1,
    0,1,1,1,1
880 RESTORE 860 :: FOR Q=1 TO 16 :: READ W,E,R,K :: SP$
    (Q)=CHR$(W+100)&CHR$(E+100)&CHR$(R+100)&CHR$(K+100)
    :: NEXT Q
890 RESTORE :: READ M$ :: DISPLAY AT(22,1):M$
900 FOR R=3 TO 11 STEP 8 :: FOR Q=5 TO 20 :: FOR W=0 TO
    7 STEP 4
910 A$=SP$(POS(HEX$,SEG$(M$,1,1),1)):: M$=SEG$(M$,2,LEN
    (M$)-1):: DISPLAY AT(Q,R+W)SIZE(4):A$
920 NEXT W :: NEXT Q :: NEXT R :: RETURN
930 !//PRINT TO PRINTER//
940 !***USE "RS232.CR" IF YOUR PRINTER IS NOT SET TO GR
    APHICS MODE***
950 OPEN #1:"RS232.CR.DA=8.BA=4800"
960 PRINT #1:CHR$(27);"A";CHR$(3)
970 FOR R=5 TO 20 :: PRINT #1:CHR$(10);CHR$(27);"K";CHR
    $(49);CHR$(0);CHR$(7);::FOR Q=5 TO 20 :: CALL GCHAR
    (R,Q,E):: IF E=101 THEN 990
980 PRINT #1:CHR$(4);CHR$(4);CHR$(7);:: GOTO 1000
990 PRINT #1:CHR$(7);CHR$(7);CHR$(7);
1000 NEXT Q :: IF R=9 THEN PRINT #1:" NAME : ";NA$ ELSE
     IF R=16 THEN PRINT #1:"LINE :";LI
1010 NEXT R :: PRINT #1:CHR$(10);CHR$(27);"K";CHR$(49);
```

```
     CHR$(0);
1020 FOR R=1 TO 49 :: PRINT #1:CHR$(4);:: NEXT R
1030 PRINT #1:CHR$(27);"0";CHR$(10);"HEX : ";M$;CHR$(13
     )
1040 CLOSE #1 :: RETURN
```

CHAPTER 16

Killer Crabs Attack

(EXTENDED BASIC)

Introduction

This program describes the lives and activities of underwater creatures. You manipulate directional keys on your computer keyboard, move your crab up, down, or diagonally, and avoid colliding with numerous other crabs, which are called killer crabs. Try to look for fish and keep gaining energy for a longer life or a better score. The so-called killer crabs are crowded on the screen and move at different speeds in both directions, left to right or right to left. When these crabs hit your crab, they hurt it and cause a tremendous decrease in the calories of your crab's body. Only the fish, which swim for a while and then disappear to reappear again at another location, will bring strength to your crab. When your crab catches the fish, it gains more energy and its body is strengthened. That means you will have a longer life and can get a better score. The best way to obtain high scores is to get more fish, so move your crab to look for the fish and do not wait for them to come to you.

Features

1. This program is designed for three levels of play: novice, intermediate, and professional. Depending on your skill, you can choose the level that suits you best the first time, and can then step up to another level when you get used to the game.
2. Fish will appear at random locations and will swim horizontally for a while. They will then disappear and will reappear again at a different place.
3. White fish (worth 20 points) will appear once in a while.
4. Your crab can move in any direction; up, down, right, and left.
5. The calories of your crab body will decrease fast as time

elapses. This feature makes the game more exciting to the player.
6. Your crab can be set to move at two speeds for the purpose of passing the killer crabs when necessary.
7. FISH EATEN and CALORIES remaining are displayed on the screen.
8. The program uses both vivid colors and sounds.

How To Play, Step-By-Step

1. Run the program. You will be asked to input your name as the player and, then, will be asked to choose the level of play, either 1, 2 or 3. The greater the number, the harder the play will be. (Killer crabs will start out faster!)
2. To move, you use the four directional keys on your computer keyboard; they are E, D, X, and S.
3. Pressing a directional key and then releasing it will cause your crab to go slow. If you continue to press a directional key, you will be able to move your crab faster.
4. Game ends when your crab runs out of body calories. Remember that the body calories will decrease fast with the elapse of time, so move your crab quickly and look for fish to catch.
5. The game will also end if you can survive 6 waves of killer crabs.

Program Explanation

0010-0040	Program name.
0050-0100	Initialize variable and define colors and characters.
0110-0150	Opening screen; ask name and level.
0160-0210	Draw playfield.
0220-0240	Put out killer crabs.
0250-0270	Put out fish.
0280-0370	Check keys and move player's crabs. Check for sprite hits.
0380-0390	Eat fish (call up EAT subprogram).
0400-0430	Subroutine: If score is evenly divisible by 4, then change killer crabs' color and screen color (next level).
0440-0460	Player got through all killer crabs waves, he is now a SUPER CRAB! End program.

0470-0490	Hit a killer crab! Turn crab white and subtract calories.
0500-0530	Out of calories; stop game and adjust high score.
0540-0560	Print score.
0570	Subprogram EAT: Make eating sound.
0580	Subprogram LOSE: Make effect when crab is touched by killer crabs.

HINT: To become a SUPER CRAB, you must survive six waves of killer crabs, each faster than the former. It is best to try to stay in the fish zones, the space between rows of killer crabs. Only in these zones are you safe from your predators, especially the sixth wave of crabs. (They are invisible!)

KILLER CRABS ATTACK

[EXTENDED BASIC]

```
10 !//////////////////////
20 !/ KILLER CRABS ATTACK /
30 !//////////////////////
40 !
50 !//INITIALIZE//
60 DIM CC(5):: CALL CLEAR :: HS=2 :: HN$="COMPUTER" ::
   CC(0)=13 :: CC(1)=6 :: CC(2)=11 :: CC(3)=14 :: CC(4)
   =16 :: CC(5)=1
70 FOR I=1 TO 8 :: CALL COLOR(I,2,8):: NEXT I
80 CALL CHAR(91,"000000060F1D3CFE")
90 CALL CHAR(104,"08107DDFFF791008"&RPT$("0",48))
100 CALL CHAR(100,"00000000008041271F1F2542800000000000
    0000000182E4F8F8A44201",96,"0000000000001679F9F254
    240000000000000000080E6F9F9A44202")
110 !//1ST SCREEN//
120 CALL MAGNIFY(3):: PRINT TAB(9);"KILLER CRABS": :TAB
    (12);"ATTACK": : : : : : : : :: CALL SPRITE(#1,100,
    2,8,122,6,0)
130 FOR I=1 TO 8 :: CALL SOUND(200,220,0):: CALL PATTER
    N(#1,100):: CALL SOUND(160,660,0):: CALL PATTERN(#1
    ,96):: NEXT I :: CALL MOTION(#1,0,0)
140 INPUT "WHAT IS YOUR NAME? ":N$
150 PRINT : : :"ARE YOU A: (1) BEGINNER":TAB(12);"(2) N
    OVICE":TAB(12);"(3) PROFESSIONAL";:: INPUT I :: IF
    I>3 OR I<1 THEN 150 ELSE SP=I*2+3
160 REM
170 CALL SOUND(100,999,0):: CALL CLEAR
180 CALL SCREEN(13):: EN=20+(9-SP)*5 :: SC=0 :: G=0
190 CALL HCHAR(1,1,91,32)
200 DISPLAY AT(2,1):"FISH EATEN:    -CALORIES:";EN
210 CALL SPRITE(#1,100,2,85,122)
```

```
220 !//KILLER CRABS//
230 RANDOMIZE :: FOR I=2 TO 10 :: Q=INT(RND*9-4)*SP ::
    IF Q=0 THEN I=I-1 :: GOTO 240 ELSE G=G+19 :: CALL S
    PRITE(#I,96,CC(SC),G,128,0,Q)
240 NEXT I
250 !//FISH//
260 FP=10 :: RANDOMIZE :: C=INT(RND*9):: IF C=0 THEN C=
    11 :: FP=20 ELSE C=0
270 CALL SPRITE(#11,104,C+5,INT(RND*8+1)*19-3,INT(RND*2
    49+4),0,-2)
280 !///MOVE///
290 FOR FD=1 TO 15 :: CALL KEY(0,X,Y):: IF Y=0 THEN CAL
    L MOTION(#1,0,0):: GOTO 360 ELSE CALL SOUND(-400,-8
    ,3)
300 IF X=69 THEN X=-4 :: Y=0 :: GOTO 340
310 IF X=88 THEN X=4 :: Y=0 :: GOTO 340
320 IF X=68 THEN X=0 :: Y=4 :: GOTO 340
330 IF X=83 THEN X=0 :: Y=-4 :: GOTO 340 ELSE X,Y=0
340 CALL MOTION(#1,X,Y):: CALL COINC(ALL,HIT):: IF HIT
    THEN 370
350 CALL PATTERN(#1,100):: CALL KEY(0,K,ST):: IF ST=-1
    THEN CALL MOTION(#1,X*3,Y*3)
360 CALL COINC(ALL,HIT)
370 IF HIT=0 THEN 480 ELSE CALL COINC(#1,#11,14,HIT)::
    IF HIT THEN CALL EAT ELSE EN=EN-10 :: CALL LOSE ::
    GOTO 470
380 EN=EN+FP :: SC=SC+1 :: IF SC/4=INT(SC/4)THEN GOSUB
    400
390 DISPLAY AT(2,12)SIZE(3)BEEP:SC :: FD=15 :: GOTO 490
400 X=SC/4 :: IF X=6 THEN 440
410 CALL SCREEN(CC(X)):: FOR I=2 TO 10 :: CALL COLOR(#I
    ,CC(X))
420 RANDOMIZE :: Q=INT(RND*9-4)*(SP+X/2):: IF Q=0 THEN
    420 ELSE CALL MOTION(#I,0,Q)
430 NEXT I :: RETURN
440 CALL CLEAR :: CALL DELSPRITE(ALL):: CALL MAGNIFY(4)
    :: CALL SPRITE(#1,100,2,75,110):: I=100
450 PRINT TAB(5);N$: :TAB(5);"IS A SUPER CRAB!!!": : :
    : : : : :: FOR X=0 TO 4.8 STEP .2 :: IF I=96 THEN
    I=100 ELSE I=96
460 CALL SOUND(300,I*10,0):: CALL PATTERN(#1,I):: CALL
    COLOR(#1,CC(INT(X))):: NEXT X :: CALL SOUND(900,100
    0,0):: CALL CLEAR :: STOP
470 CALL COLOR(#1,2)
480 CALL COINC(ALL,HIT):: IF HIT THEN 370 ELSE IF EN>0
    THEN 490 ELSE CALL LOSE :: FD=15
490 EN=EN-1 :: DISPLAY AT(2,25):EN :: CALL PATTERN(#1,9
    6):: NEXT FD :: IF EN>0 THEN 260
500 !///OUT OF FOOD///
510 FOR I=1 TO 8 :: CALL SOUND(100,3000,0):: NEXT I ::
    DISPLAY AT(11,6):"RAN OUT OF ENERGY!" :: GOTO 520
520 DISPLAY AT(2,25):EN :: DISPLAY AT(10,6):N$
530 IF SC>HS THEN HS=SC :: HN$=N$
540 !//PRINT SCORE//
550 DISPLAY AT(13,6):"PRESS REDO OR BACK" :: DISPLAY AT
```

```
    (17,9):"HIGH SCORE": :TAB(9);HN$;": ";HS
560 CALL KEY(0,K,S):: IF S=0 THEN 560 ELSE IF K=6 THEN
    140 ELSE IF K<>15 THEN 560 ELSE CALL CLEAR :: STOP
570 SUB EAT :: CALL MOTION(#1,0,0,#11,0,0):: FOR I=1010
    TO 210 STEP -100 :: CALL SOUND(-1000,I,0):: NEXT I
    :: CALL SOUND(200,110,0):: SUBEND
580 SUB LOSE :: CALL SOUND(2000,110,0,-8,0):: CALL COLO
    R(#1,16):: CALL MOTION(#1,0,0):: FOR D=1 TO 100 ::
    NEXT D
590 SUBEND
```

CHAPTER 17

Home Bound

(EXTENDED BASIC)

Introduction

Crossing a dangerous highway and an unknown lake in order to go home and rest safely in their nest is the purpose of the amphibians (or a successful player). A definite number of amphibians are generated initially for each game. One at a time, an amphibian will go to the center of the screen for its turn at an attempt to go home. You will act as an amphibian. Using the directional keys, choose your best moment and direction to cross the highway. When you come to the edge of the lake, jump on floating logs to get to and nest in the square spaces at the top of the screen.

A white bee running across the screen acts as a timer to limit your time for each amphibious crossing. The moving bee will increase its speed for every amphibian that follows; thus, the process of crossing the highway gets harder each time you try to pass. Your skill, your speed in manipulating the directional keys, and the accuracy of your finger movements are all required to save the amphibians from being smashed or drowned so that you can bring them home and obtain a higher score.

Reaching home at the precise moment that the timer bug passes nearby allows you to devour him and increase your score by 200 points each time.

Even with an extra amphibian as a bonus each time the score passes the 1000 mark, it is still hard for even a professional player to reach all six homes in one game.

A player who fills all the homes with amphibians will be a winner and will be awarded with the playing of special triumphal music.

This game has many aspects to consider in order to be successful and gain a high score. It is an imitation of the Atari® Frogger™ but it utilizes all the specific traits available in a powerful TI Extended BASIC home computer program.

Features

1. The program uses vivid color graphics, sprites, music, and sound. Vehicles, logs, a timer, a snake, and amphibians are moving sprites that fill the game with many interesting aspects.
2. The timer increases its speed for each play that follows and this allows different levels for the player; the game grows harder and more professional, play after play.
3. Random direction and speed are designed into the game for the vehicles so as to increase the unexpected aspects of the game.
4. Six bonuses are provided and there is a scoring display after every successful return home. An interesting feature, which permits the amphibian to catch the bee timer, is included but it requires precise calculation and skill on the player's part. An extra 200 points are given for each catch.
5. Extra amphibians are also given as a bonus for every 1000 points.
6. A sense of realism is introduced in the game when an amphibian is hit by a car; blood will flow and splash out on the road.
7. Three high scores, along with the player's name, are kept in memory and are displayed after every game.

How To Play, Step-By-Step

1. Run the program. Input Y for instructions, N for none.
2. Use directional keys (E, D, X, S) to move the amphibian. Avoid collision with vehicles on the freeway. Jump right on the log and move to any home location so you can jump in before the bee timer reaches the right side of the screen.
3. If the bee timer is nearby, you can get extra points (200) by licking (catching) it. Right after you jump in your home, press S (lick left) or D (lick right) to catch the bee timer.

ATARI is the registered trademark of Atari, Inc.
FROGGER is a trademark of Sega Enterprises, Inc.

4. For every jump you make on the street or grass, 5 points will be added to your present score. For every jump you make on the logs, you will receive 10 points. A bonus of 80 points will be added for every amphibian that reaches home, and a super bonus of 500 is awarded when all six homes are occupied.
5. Game ends when all your amphibians have been killed.
6. If your score is among one of the three highest, you will be asked to input your name.

Program Explanation

0010-0040	Program name.
0050-0802	Display first screen.
0090-0130	Instructions.
0140-0340	Initialize: Get speech, and define colors and characters.
0350-0420	Draw playscreen.
0430-0590	Put logs and cars on screen; then start them moving.
0600-0620	Put amphibian, random snake, and bug on screen.
0630-0670	Start. When player presses any key, start timer and game.
0680-0810	Land Section. Check keys and move amphibian.
0700-0710	Up. (If higher than ½ screen, go to Water Section.)
0720-0730	Right.
0740-0750	Left.
0760-0770	Down.
0780	Check against amphibian moving out of playfield.
0790	Subroutine to check for hit and add 5 points (move on land).
0800-0810	Check for coincidence 3 times (for accuracy), go to line 690 if not hit by cars. If hit, then go to line 1060.
0820-0880	Water Section. Check key and move amphibian.
0840	Up. (Go to line 900 if amphibian tries for home.)
0850	Right.
0860	Left.
0870	Down. (If needed, go back to Land Section.)
0890-0920	Land in home.
0930-0990	Lick.
1000-1050	Scoring and display score. Add bonus.
1060-1080	Amphibian dies.

1090-1120	Land on log. Check for coincidence 3 times (for accuracy). Add 10 points (for successful jump on logs) and move amphibian same direction and speed of log. Go back to line 830 (Water Section).
1130-1160	Subroutine for checking for extra amphibian.
1170-1370	DATA.
1380-1660	User defined subprograms.
1390-1400	SUB SONG. Play music.
1410-1440	SUB SQUISH. Amphibian hit by cars.
1450-1500	SUB DROWN. Amphibian falls in water.
1510-1590	SUB OVER. Adjust high score, ask for replay.
1600-1620	SUB FROGO. Make next amphibian move to middle of screen, at bottom, to start playing.
1630-1650	SUB DING. Add up bonuses with sound.
1660	SUB SPCH. Get croaking and bleeding sound.

HOME BOUND
[EXTENDED BASIC]

```
10 ! //////////////
20 ! / HOME BOUND /
30 ! //////////////
40 !
50 !//1ST SCREEN//
60 CALL CLEAR :: PRINT TAB(8);"<<<<<<<>>>>>>>":TAB(8);"
   < HOME BOUND >":TAB(8);"<<<<<<<>>>>>>>": : : : : : :
    : : : :
70 CALL SOUND(100,888,0):: PRINT " DO YOU NEED INSTRUCT
   IONS?";
80 CALL KEY(0,K,S):: IF S=0 THEN 80 ELSE IF K-89 THEN 1
   50
90 PRINT TAB(10);"HOME BOUND": :"  YOU ARE APPROACHED B
   Y A   GROUP OF AMPHIBIANS AND THEYASK YOU TO HELP TH
   EM GET    HOME..."
100 PRINT :"CONTROLS: ARROW KEYS": :TAB(12);"POINTS": :
    "HOP ON GROUND =   5 PTS":"  HOP ON LOGS =  10 PTS"
110 PRINT "    BLACK BUG =  80 PTS":"    WHITE BEE = 20
    0 PTS": :"GET HOME BEFORE BEE REACHES RIGHT EDGE OF
     SCREEN!"
120 PRINT :"USE ""S""/""D"" TO LICK BEE.": :"FREE FROG
    EVERY 1000 POINTS.": :"GOOD LUCK...PRESS ANY KEY";
130 CALL KEY(0,K,S):: IF S=0 THEN 130
140 ! // INITIALIZE //
150 DIM C$(1,1),N$(2),N(2):: CALL CHARPAT(63,A$):: CALL
    CHAR(59,A$)
160 CALL INIT :: CALL LOAD(-31878,0):: CALL SPCH(CR$,BL
    $):: CALL CLEAR :: RANDOMIZE :: CALL MAGNIFY(3):: A
    $=RPT$("0",42)
```

```
170 CALL CHAR(33,RPT$("0",12)&"C0C"&RPT$("0",14)&"30300
    285A7E7E5A2800",36,"183C6C397E396C3C18"&A$)
180 CALL CHAR(40,"7F9EF17F8FF16F3700000000000000003FFAC
    FFF96FFCD7E0")
190 CALL CHAR(44,"02051F1B1C0F21777F7F3E3F1E1F36362050F
    CEC1CF8C2F7FF7FBEFEBCFCB636",60,A$&A$)
200 CALL CHAR(128,"077BDD9D9DDD7B070000000000000000FFFD
    FFFFFFFFFDFF0")
210 CALL CHAR(132,"7073AEFDFDAE737000000000000000003CE4
    F8FCFCF8E43C0")
220 CALL CHAR(136,"3F151F3E3E1F153F0000000000000000B66E
    CEC6C6CE6EB60")
230 CALL CHAR(140,"38777C7F7F7C773800000000000000070F8
    BCDCDCBCF8700")
240 C$(0,0)=RPT$("0",39)&"77CE" :: C$(0,1)="00C0E0603E0
    F000000000000000000000000000077CE"
250 C$(1,0)="000000E03E07" :: C$(1,1)="000000E03E070000
    0000000000000000000307067CF"
260 CALL CHAR(112,"00425A3C7E9942C3"&A$)
270 CALL CHAR(116,"506B0C0E0E0C6B5"&A$)
280 CALL CHAR(120,"00C342817E3C5A42"&A$)
290 CALL CHAR(124,"0AD630707030D6"&A$)
300 CALL CHAR(96,"105438100083")
310 CALL CHAR(100,"00000000000183C3466C380000000000000
    0E0B0F0C848CD858783")
320 RESTORE 1310 :: READ A$ :: CALL CHAR(92,A$,104,RPT$
    ("F",22)&"07FFFFFFFFFFFFFF00FF00FFFFFF")
330 FOR I=3 TO 8 :: CALL COLOR(I,2,4):: NEXT I
340 CALL SCREEN(4):: CALL COLOR(1,2,6,2,2,6,9,3,4,10,2,
    12,11,13,4)
350 !//1ST SCREEN//
360 ! // SET UP //
370 SC=0 :: SC1=1000 :: S=7 :: FS=4
380 CALL HCHAR(23,1,96,128)
390 CALL HCHAR(14,1,96,64)
400 CALL HCHAR(16,1,104,224):: CALL HCHAR(17,1,105,32):
    : CALL HCHAR(19,1,106,32):: CALL HCHAR(21,1,105,32)
410 CALL LOAD(-31878,0):: FOR I=4 TO 31 STEP 5 :: CALL
    HCHAR(1,I,32,2):: CALL HCHAR(2,I,33):: CALL HCHAR(2
    ,I+1,34):: NEXT I
420 H=0 :: CALL SPRITE(#27,60,7,1,1)
430 ! /LOGS/
440 X=24
450 FOR I=1 TO 11 STEP 5 :: A=24 :: FOR J=I+1 TO I+5 ::
    CALL SPRITE(#J,40,15,A,X):: A=A+16 :: NEXT J :: X=X
    +87 :: NEXT I
460 ! /VEHICLES/
470 A=120
480 FOR I=17 TO 20 :: CALL SPRITE(#I,(I-17)*4+128,INT(R
    ND*13+3),A,1,#I+4,(I-17)*4+128,INT(RND*13+3),A,INT(
    RND*200+30)):: A=A+16 :: NEXT I
490 ! /ARRAY/
500 FOR I=0 TO 4 :: RANDOMIZE :: CALL COLOR(#I+2,15,#I+
    7,15,#I+12,15):: IF INT(RND*2)AND H THEN 540
510 M(I)=INT(RND*S+RND*4)+1 :: IF M(I)=0 THEN 510
```

```
520 B=(I/2=INT(I/2)):: IF B=0 THEN B=B+1
530 M(I)=M(I)*8
540 NEXT I :: CALL SCREEN(4)
550 FOR J=0 TO 4 :: CALL MOTION(#J+2,0,M(J),#J+7,0,M(J)
    ,#J+12,0,M(J)):: NEXT J
560 !/MOVE CARS/
570 FOR I=17 TO 20
580 RANDOMIZE :: IF INT(RND*3)=0 AND H THEN 590 ELSE B=
    SGN(I-18.5)*INT(RND*S+4+RND*4+1):: CALL MOTION(#I,0
    ,B,#I+4,0,B)
590 NEXT I :: GOTO 610
600 FOR I=0 TO 4 :: CALL COLOR(#I+2,15,#I+7,15,#I+12,15
    ):: NEXT I
610 CALL LOAD(-31878,26):: CALL SPRITE(#1,112,13,183,12
    6,#25,36,16,1,16):: IF INT(RND*3)>0 THEN CALL SPRIT
    E(#26,100,2,104,1,0,RND*6+3)
620 RANDOMIZE :: B=INT(RND*6):: CALL GCHAR(1,B*5+5,I)::
    IF I=32 THEN CALL VCHAR(1,B*5+5,35)ELSE B=-4/5
630 ! // START //
640 CALL HCHAR(23,1,96,32):: DISPLAY AT(24,3)SIZE(FS):R
    PT$(CHR$(112),FS-1)&CHR$(96)
650 DISPLAY AT(23,12)SIZE(6)BEEP:"READY;" :: FOR I=1 TO
    8 :: CALL KEY(0,K,ST)::IF ST THEN I=10
660 NEXT I :: CALL HCHAR(23,1,96,32):: IF I=9 THEN 650
    ELSE CALL HCHAR(15,10,96,15)
670 X=183 :: Y=126 :: CALL MOTION(#25,0,S/3)
680 !/ON LAND SECTION/
690 CALL KEY(0,K,ST):: CALL POSITION(#25,X1,Y1):: IF Y1
    >243 THEN CALL TIMEEND :: GOTO 1060 ELSE IF ST=0 TH
    EN 800
700 IF K-69 THEN 720 ELSE IF X<104 THEN 840 ELSE X=X-8
    :: CALL PATTERN(#1,112):: CALL LOCATE(#1,X,Y):: GOS
    UB 790
710 IF HIT THEN 1060 ELSE X=X-8 :: CALL LOCATE(#1,X,Y):
    : CALL SOUND(-100,690,8):: GOTO 800
720 IF K-68 THEN 740 ELSE IF Y>248 THEN 780 ELSE Y=Y+4
    :: CALL PATTERN(#1,116):: CALL LOCATE(#1,X,Y):: GOS
    UB 790
730 IF HIT THEN 1060 ELSE Y=Y+4 :: CALL LOCATE(#1,X,Y):
    : CALL SOUND(-100,680,8):: GOTO 800
740 IF K-88 THEN 760 ELSE IF X>180 THEN 780 ELSE X=X+8
    :: CALL PATTERN(#1,120):: CALL LOCATE(#1,X,Y):: IF
    X>96 THEN GOSUB 790 ELSE HIT=0
750 IF HIT THEN 1060 ELSE X=X+8 :: CALL LOCATE(#1,X,Y):
    : CALL SOUND(-100,880,8):: GOTO 800
760 IF K-83 THEN 800 ELSE IF Y<12 THEN 780 ELSE Y=Y-4 :
    : CALL PATTERN(#1,124):: CALL LOCATE(#1,X,Y):: GOSU
    B 790
770 IF HIT THEN 1060 ELSE Y=Y-4 :: CALL LOCATE(#1,X,Y):
    : CALL SOUND(-100,830,8):: GOTO 800
780 CALL SOUND(100,110,5):: GOTO 690
790 CALL COINC(ALL,HIT):: CALL SOUND(-100,220,7):: SC=S
    C+5 :: RETURN
800 CALL COINC(ALL,HIT):: IF NOT HIT THEN CALL COINC(AL
    L,HIT):: IF NOT HIT THEN CALL COINC(ALL,HIT):: IF N
```

```
    OT HIT THEN 690
810 GOTO 1060
820 !/IN WATER SECTION/
830 CALL KEY(0,K,ST):: CALL POSITION(#25,X1,Y1):: IF Y1
    >243 THEN CALL TIMEEND :: GOTO 1060 ELSE IF ST=0 TH
    EN 830 ELSE CALL SOUND(10,999,5)
840 IF K-69 THEN 850 ELSE CALL MOTION(#1,0,0):: CALL PO
    SITION(#1,X,Y):: IF X<24 THEN 900 ELSE CALL PATTERN
    (#1,112):: CALL LOCATE(#1,X-16,Y):: C=39 :: GOTO 11
    00
850 IF K-68 THEN 860 ELSE CALL MOTION(#1,0,0):: CALL PO
    SITION(#1,X,Y):: IF Y>248 THEN 1060 ELSE CALL PATTE
    RN(#1,116):: CALL LOCATE(#1,X,Y+8):: C=23 :: GOTO 1
    100
860 IF K-88 THEN 870 ELSE CALL MOTION(#1,0,0):: CALL PO
    SITION(#1,X,Y):: IF X<87 THEN CALL PATTERN(#1,120):
    : CALL LOCATE(#1,X+16,Y):: C=7 :: GOTO 1100 ELSE 74
    0
870 IF K-83 THEN 880 ELSE CALL MOTION(#1,0,0):: CALL PO
    SITION(#1,X,Y):: IF Y<12 THEN 1060 ELSE CALL PATTER
    N(#1,124):: CALL LOCATE(#1,X,Y-8):: C=23 :: GOTO 11
    00
880 GOTO 820
890 !/LAND IN HOME/
900 IF Y>250 THEN 1060 ELSE Y=Y+4 :: CALL MOTION(#25,0,
    0):: Y=INT(Y/8+.5)+1 :: CALL GCHAR(2,Y,HIT):: IF HI
    T-34 THEN 1060 ELSE CALL GCHAR(1,Y,HIT):: H=H+1 ::
    SC=SC+50
910 CALL DELSPRITE(#1):: CALL SOUND(60,-3,0,110,0):: CA
    LL VCHAR(1,Y-1,44):: CALL VCHAR(1,Y,46):: CALL VCHA
    R(2,Y-1,45):: CALL HCHAR(2,Y,47)
920 CALL SAY(,CR$,,CR$):: IF HIT=35 THEN CALL SOUND(200
    ,880,0,330,5):: CALL SOUND(100,770,0):: CALL SOUND(
    300,550,0)ELSE CALL VCHAR(1,B*5+5,32)
930 Y=(Y-1)*8 :: CALL KEY(0,K,I):: IF K-83 AND K-68 THE
    N 1000 ELSE CALL POSITION(#25,X,I):: CALL LOCATE(#2
    7,3,Y+16*(K=83))!/LICK/
940 CALL SOUND(-200,-7,19):: CALL CHAR(60,C$((K=83)+1,0
    )):: CALL SOUND(-240,-5,16):: CALL CHAR(60,C$((K=83
    )+1,1))
950 CALL COINC(ALL,I):: IF I=0 THEN CALL COINC(ALL,I)::
    IF I=0 THEN 980
960 CALL SOUND(50,-5,9,110,4):: SC=SC+200 :: CALL CHAR(
    36,"0C1E0202060C181E"&RPT$("0",16)&"44EEAAAAAAAAEE4
    4")
970 FOR I=800 TO 2000 STEP 100 :: CALL SOUND(99,I,4)::
    NEXT I :: CALL DELSPRITE(#25):: CALL CHAR(36,"183C6
    C397E396C3C18"&RPT$("0",15))
980 IF K=68 THEN CALL CHAR(62,"0")
990 CALL CHAR(60,RPT$("0",33)):: CALL LOCATE(#27,1,1)
1000 DISPLAY AT(15,10)SIZE(LEN(STR$(SC))+6):"SCORE:"&ST
     R$(SC):: IF HIT=35 THEN DISPLAY AT(23,10)SIZE(8):"
     BONUS:80" :: CALL DING(SC,80)
1010 GOSUB 1140 :: IF S<14 THEN S=S+1
1020 IF H-6 THEN 1050 ELSE CALL SCREEN(6):: DISPLAY AT(
```

```
     23,10)SIZE(10):"BONUS:500";CHR$(96):: CALL DING(SC
     ,500):: CALL SONG(55):: GOSUB 1140
1030 FOR I=4 TO 31 STEP 5 :: CALL SOUND(200,770-I*10,0)
     :: CALL HCHAR(2,I,33):: CALL HCHAR(2,I+1,34):: CAL
     L VCHAR(1,I,45):: CALL VCHAR(1,I+1,47)
1040 CALL SOUND(200,770+I*10,0):: CALL HCHAR(1,I,32,2):
     : NEXT I :: H=0 :: CALL SONG(14):: GOTO 610
1050 X=14 :: CALL SONG(X):: GOTO 1070
1060 CALL VCHAR(1,B*5+5,32):: CALL MOTION(#25,0,0):: IF
     X<104 THEN CALL DROWN(FS)ELSE CALL SQUISH(FS,BL$)
1070 CALL DELSPRITE(#25,#26):: DISPLAY AT(15,10)SIZE(LE
     N(STR$(SC))+6):"SCORE:"&STR$(SC):: GOSUB 1140
1080 IF FS=0 THEN CALL OVER(SC,N$(),N()):: GOTO 370 ELS
     E IF X=14 THEN 500 ELSE CALL FROGO(FS+3,15):: GOTO
      600
1090 !/LAND ON LOGS/
1100 CALL COINC(ALL,HIT):: IF NOT HIT THEN CALL COINC(A
     LL,HIT):: IF NOT HIT THEN CALL COINC(ALL,HIT):: IF
     NOT HIT THEN 1060
1110 SC=SC+10
1120 CALL MOTION(#1,0,M((X-C)/16)):: CALL SOUND(100,999
     ,7):: GOTO 830
1130 !//SUBR EXTRA FROG//
1140 IF SC<SC1 THEN RETURN ELSE CALL SCREEN(11):: DISPL
     AY AT(23,10)SIZE(10):"EXTRA";CHR$(96);"FROG" :: CA
     LL SOUND(200,440,6):: CALL SOUND(600,550,4):: CALL
      SOUND(99,770,0)
1150 CALL SOUND(999,880,9):: CALL SCREEN(4):: SC1=SC1+1
     000 :: FS=FS+1 :: CALL FROGO(1,FS+2)
1160 CALL DELSPRITE(#1):: CALL VCHAR(24,FS+3,112):: CAL
     L VCHAR(24,4,96):: RETURN
1170 !//SONG//
1180 DATA 261,250,349,250,349,250,391,250,391,250,440,3
     75,466,125,523,250,466,250,440,250,440,250
1190 DATA 391,250,391,250,349,500
1200 DATA 261,250,349,250,349,250,391,250,391,250,440,3
     75,466,125,523,250,466,250,440,250,440,250
1210 DATA 391,250,391,250,349,500
1220 DATA 523,125,466,125,440,250,440,250,440,250,466,1
     25,440,125,391,250,391,250,391,250
1230 DATA 523,125,466,125,440,250,440,250,440,250,466,1
     25,440,125,391,250,391,250,391,250
1240 DATA 523,250,440,250,349,250,391,250,391,250,349,3
     75,44733,1
1250 !/SPLASH/
1260 DATA 00000000010202040502010000000000000000000804
     0A02040C
1270 DATA 00010608090A12141422190403000000000804020904
     8282848C810E
1280 DATA 0708110020440811 90A0004844300906C02010C834128
     A449A2202860808D
1290 DATA 10609000A0002810800000004090 4026D00212410C0A0
     204002000060002142
1300 !/BLOOD/
1310 DATA 000000091F1F0F07071F090000000000000000080C0E0E
```

```
    0E0E0C0808
1320 DATA 000010392F1F0F040F1F1B0101000000000000808080F0F
    8E8F0E08080C
1330 DATA 000020D97F7F331F1F1F7F351B0100000000C0E0A0F8F
    EF6FEECE0A0C0E0C
1340 !/BLAH/
1350 DATA 20,19,182,85,142,245,155,41,251,82,201,56,30,
    70,78,68,197,155,183,177,201
1360 !/CROAK/
1370 DATA 22,21,40,83,141;70,98,90,82,28,230,237,117,31
    ,242,168,39,0,1,196,226,34,1
1380 !/SUBPROGRAMS/
1390 SUB SONG(X):: RESTORE
1400 FOR I=1 TO X :: READ A,B :: CALL SOUND(B,A,0):: NE
    XT I :: SUBEND
1410 SUB SQUISH(FS,B$):: CALL SCREEN(7):: CALL POSITION
    (#1,A,B):: CALL SPRITE(#28,92,7,A-4,B-4)
1420 CALL SCREEN(4):: FOR I=0 TO 30 :: CALL SOUND(-500,
    150,I,-7,I):: NEXT I :: RESTORE 1310 :: FS=FS-1
1430 FOR I=1 TO 3 :: READ A$ :: CALL CHAR(92,A$):: CALL
    SAY(,B$):: NEXT I :: FOR I=1 TO 300 :: NEXT I :: C
    ALL DELSPRITE(#1,#28)
1440 RESTORE 1310 :: READ A$ :: CALL CHAR(92,A$):: SUBE
    ND
1450 SUB DROWN(FS):: CALL SOUND(-4000,-6,20):: CALL POS
    ITION(#1,X,Y):: CALL SPRITE(#1,108,8,X-4,Y):: REST
    ORE 1260
1460 FOR I=1 TO 4 :: CALL SOUND(-4000,-5,I*5):: READ A$
    :: CALL CHAR(108,A$):: NEXT I
1470 CALL DELSPRITE(#1):: FS=FS-1 :: SUBEND
1480 SUB TIMEEND :: CALL MOTION(#25,0,0)
1490 FOR I=2 TO 6 :: FOR A=2 TO 16 :: CALL COLOR(#A,I):
    : NEXT A :: NEXT I
1500 CALL MOTION(#1,0,0):: SUBEND
1510 SUB OVER(SC,N$(),N()):: CALL DELSPRITE(ALL)
1520 IF SC<=N(2)THEN 1550 ELSE IF SC<=N(1)THEN X=2 :: N
    (X)=SC ELSE N(2)=N(1):: N$(2)=N$(1):: IF SC<=N(0)T
    HEN X=1 :: N(X)=SC ELSE X=0 :: N(1)=N(X):: N$(1)=N
    $(X):: N(X)=SC
1530 CALL SONG(55):: FOR I=3 TO 8 :: CALL COLOR(I,2,16)
    :: NEXT I
1540 CALL SCREEN(16):: DISPLAY AT(8,10):"YOU DID IT!":
    :"  YOU ARE ONE OF THE BEST!": :"YOUR HONORABLE NA
    ME;" :: ACCEPT AT(12,21):N$(X)
1550 FOR I=3 TO 8 :: CALL COLOR(I,2,6):: NEXT I
1560 DISPLAY AT(3,9):"HIGH SCORES" :: FOR I=0 TO 2 :: D
    ISPLAY AT(I+5,7):N$(I);TAB(16);N(I):: NEXT I
1570 DISPLAY AT(8,1): :TAB(10);"GAME OVER": :"      PRES
    S REDO OR BACK": : :: DISPLAY AT(15,10)SIZE(LEN(ST
    R$(SC))+6):"SCORE:"&STR$(SC)
1580 CALL KEY(0,K,ST):: IF K=15 THEN CALL CLEAR :: STOP
    ELSE IF K<>6 THEN 1580
1590 CALL CLEAR :: CALL SCREEN(4):: FOR K=3 TO 8 :: CAL
    L COLOR(K,2,4):: NEXT K :: CALL COLOR(1,2,6):: SUB
    END
```

```
1600 SUB FROGO(A,B):: CALL VCHAR(24,A+1,96):: CALL SPRI
     TE(#1,116,13,185,A/8+1)
1610 FOR I=A TO B STEP .5 :: CALL SOUND(-99,690,19):: C
     ALL LOCATE(#1,185,I*8+1):: CALL SOUND(-99,220,19):
     : NEXT I
1620 SUBEND
1630 SUB DING(A,B)
1640 FOR I=A TO A+B STEP 10 :: CALL SOUND(1,-1,19):: DI
     SPLAY AT(15,16)SIZE(LEN(STR$(I))):STR$(I):: NEXT I
1650 A=A+B :: SUBEND
1660 SUB SPCH(A$,S$):: RESTORE 1350 :: S$,A$=CHR$(96)&C
     HR$(0):: READ A :: FOR I=1 TO A :: READ B :: S$=S$
     &CHR$(B):: NEXT I :: READ A :: FOR I=1 TO A :: REA
     D B :: A$=A$&CHR$(B):: NEXT I :: SUBEND
```

CHAPTER 18

Dungeon

(EXTENDED BASIC)

Introduction

This is an adventure game. You are trapped in an underground dungeon infested with man-eating animals whose only purpose is to keep you from escaping. To get out of this ever-changing maze, you must exit through a door on the fourth level. You start on the first level and must descend through chambers and corridors toward your goal. WARNING: The monsters get more dangerous as you get deeper!

With a little strategy (and a lot of luck), you have a chance of getting out of the "Dungeon" alive.

Features

1. The program is designed for one to four players; the greater the number of players, the harder it is to escape.
2. All players move in a group.
3. All players will have choices they must make.
   A. Before game
      -Weapon(s); a maximum of four weapons can be selected.
      -Shield type (for armor).
   B. During Game
      -To fight or to retreat from monster.
      -To aim at monster.
      -To change weapons.
      -To move in four directions (as a group).
4. More than 25 different monsters are programmed to appear in the dungeon. The deeper you progress through the dungeon, the more dangerous the monsters become.
5. The program also sets a display of the monsters' and the characters' status by listing the following:

-Name of character/monster.
-Damage capability of weapon being used.
-Armor class.
-Strength remaining.
-Force of swing.
-Weapon holding (not applicable for monsters).

6. There are secret passages.
7. An arrow points in the direction you are heading.
8. Every level is distinguished by a specific screen color. Situation messages are also given at appropriate times.
9. Other features that are noted during combat are:
   -Critical hits on monster by player.
   -Player may break his/her weapon.
   -Monster may hit itself and cause damage to itself.

How To Play, Step-By-Step

1. Run program.
2. Enter number of adventurers and their names.
3. You must forge at least one weapon for each player. Forge weapon(s) by entering the appropriate number. Stop forging. For each weapon you forge, you will expend a certain amount of Health Points.
4. Next, each person, if he wishes, may forge shields. The larger the shield, the more protection—Armor Class.
5. After everyone has finished forging a weapon, you must enter which weapon you wish to hold in your hand. (If anyone has not forged a weapon, the program will go into an endless loop and must be CLEARed.)
6. Read the instructions!
7. When the game starts, everyone is transported to a corridor. In corridors, the screen color is always light green. In chambers or rooms, the screen color will be different for each level of the dungeon. The colors are:
   1st level = Medium red
   2nd level = Light red
   3rd level = Dark yellow
   4th level = Light yellow
8. Use arrow key letters to input the direction to walk. The inputs are:
   E = Enter room/forward
   X = Turn about face

D = Turn right
S = Turn left

9. Monsters like to live in rooms and chambers.
10. When you meet monster(s), you are asked:
    WHAT WILL YOU DO?
    The inputs are:
    F = Fight monsters
    R = Retreat
    C = Change weapon you are holding for another of your own. (Type your name, then one of the displayed weapons.)
    X = Exchange weapon with another player. Type your name, press enter, then type other player's name.)
11. If you have decided to fight or are unsuccessful in escaping, you will be asked if there is more than one player.
    WHO WILL FIGHT?
    Enter name of the brave character.
12. The fighter's status and his opponent's status will be displayed on screen:
    NAME
    DAMAGE:
    AC:
    HP:
    FORCE:
    USING: Shows weapon used. (Monsters have no weapons, they use teeth or claws!)

A. *DAMAGE:* This is the DAMAGE capability of the weapon being used. If monster scores a hit on the player, the player's Health Point will be deducted from the monster's DAMAGE. If player hits a monster, monster's Health Point is deducted (a number from 3 to the player's DAMAGE+3), unless hit was critical (see FORCE). DAMAGE for each weapon is the amount of HP required to forge it.

B. *AC:* This means Armor Class. To score a hit, both player and monster have to have a FORCE greater than or equal to his opponent's AC.

SHIELD	AC
None	9
Wooden	10
S. Plate	12
L. Plate	14

C. *HP:* HP is the Health Point. The greater this number, the healthier the player or monster is. When the HP decreases to zero or under, that being is dead. When a player kills a monster, his HP is increased by one half of the monster's initial HP. Thus, the stronger the monster is which one kills, the stronger the player will get.

D. *FORCE:* This is the FORCE of the attack; it is a number from 2 to 20. To hit an opponent, the FORCE must be greater than or equal to the opponent's AC. If a player gets a FORCE of 20, then he can make a critical hit and will usually kill the monster. A critical hit will cause DAMAGE to a monster for a number up to the player's DAMAGE*9+9.

Monsters cannot make critical hits on players.

If a player gets a FORCE of 1, then the weapon he is using breaks! If this happens, the character will automatically try to escape when monster attacks. If the character can escape, the player will be asked to choose another weapon if he has any left.

If monster gets a FORCE of 1, then its HP is subtracted from its DAMAGE. Player will still get credit if monster kills itself.

13. Press SPACE BAR to see who gets the first attack.
14. During melee, you will be asked:

    YOU ATTACK . . .

    Enter W for attack with weapon.

    Enter R to try to retreat (monster will attack if you don't succeed).

    Enter A to aim at monster. This will forfeit your attack but will get you a FORCE ranging from 8 to 28 instead of the regular 2 to 20. You may also get a 1 (weapon breaks)!
15. Proceed until you make it out of the DUNGEON or die at the hands of the monsters.

Program Explanation

0010-0040	Program name.
0050-0060	Initialize arrays, define arrows.

0070-0100	Ask how many players and set HP.
0110-0200	Armory.
0210-0290	Shields.
0300-0360	Ask for weapon in hand.
0370-0400	Instruction on the way out.
0410	Print commands.
0420	Check to put out stair.
0430-0700	Randomly put out situations.
0400-0480	Corridors.
0490-0510	Side passages.
0520-0540	Corners.
0550-0600	Doors and doorways.
0610-0640	Rooms.
0650-0700	Ask player for direction to move.
0710-0750	DATA for monsters.
0760-1360	MONSTER and FIGHT subroutine.
0760-0800	Generate random monster(s).
0810-0820	Subroutine for all retreats.
0830-0860	Generate monster's HP, then ask for command.
0870-0970	Exchange weapons.
0980-1020	Choose who to attack. If everyone's dead, then end.
1030-1110	Choose first attacker (character or monster).
1120-1160	Weapon broke, must escape, choose new weapon.
1170-1240	Character attacks, aims, or retreats.
1250-1260	Character breaks weapon.
1270-1290	Monster attacks.
1300-1320	Hit character!
1330-1340	Subroutine to generate FORCE for character.
1350	Subroutine to generate FORCE for monster.
1360	Give character credit for killing monster, then RETURN.
1370-1420	Subroutine to keep track of position in dungeon (N,S,E,W).
1430	YOU MADE IT OUT! ! !, end.
1440	SUB WAIT.
1450-1560	SUB TRANSlation between numbers and names of articles.
1570-1590	SUB DIRection (translation from numbers).
1600-1640	SUB DEScribe situations, rooms, and chambers with adjectives.

DUNGEON
[EXTENDED BASIC]

```
10 !////////////
20 !/ DUNGEON /
30 !////////////
40 !
50 REM
60 CALL CLEAR :: ON WARNING NEXT :: OPTION BASE 1 :: DI
   M PL$(4),HP(4),U(8),W(4,5):: CALL CHAR(96,"10387C92
   10101010080406FF06040800101010927C3810102060FF6020
   10")
70 PRINT "HOW MANY DARE TO BE AN":"ADVENTURER";:: INPUT
   A :: IF A>4 THEN 70
80 FOR I=1 TO A :: PRINT :"NAME OF PLAYER";I;":" :: ACC
   EPT AT(23,19)BEEP SIZE(7):PL$(I):: NEXT I
90 !//HIT POINTS//
100 CALL CLEAR :: FOR I=1 TO A :: HP(I)=38 :: NEXT I
110 !/WEAPONS/
120 DISPLAY AT(5,9):"   ARMORY..." :: CALL SCREEN(11)::
    CALL WAIT(400)
130 DISPLAY AT(3,4):"WEAPONS      HP. TO FORGE":RPT$("-"
    ,13)&" "&RPT$("-",14):"(1) DAGGER";TAB(20);"3":"(2)
    SWORD";TAB(20);"6"
140 DISPLAY AT(7,1):"(3) BATTLE AXE";TAB(20);"9": :"ENT
    ER ""0"" TO STOP FORGING"
150 FOR I=1 TO A :: S=0 :: CALL HCHAR(11,1,32,128)
160 S=S+1 :: IF S>4 THEN 200
170 DISPLAY AT(18,1)BEEP:PL$(I);",YOU HAVE";HP(I);"HP."
    : :"FORGE WHICH WEAPON?" :: ACCEPT AT(20,20)SIZE(1)
    VALIDATE("0123"):K :: IF K=0 THEN 200
180 W(I,S)=K*3 :: HP(I)=HP(I)-K*3 :: IF HP(I)<1 THEN W(
    I,S)=0 :: HP(I)=1 :: GOTO 200
190 CALL TRANS(W(I,S),A$):: DISPLAY AT(S+10,7):A$ :: GO
    TO 160
200 CALL WAIT(200):: NEXT I
210 !/SHIELD/
220 DISPLAY AT(3,4):"SHIELD TYPE  HP. TO MAKE":RPT$("-"
    ,15)&" "&RPT$("-",12):"(1) WOODEN";TAB(21);"5":"(2)
    S. PLATE";TAB(21);"10"
230 DISPLAY AT(7,5):"L. PLATE";TAB(21);"15" :: CALL HCH
    AR(10,1,32,192)
240 FOR I=1 TO A :: DISPLAY AT(13,5):""
250 DISPLAY AT(18,1)BEEP:PL$(I);",YOU HAVE";HP(I);"HP."
    : :"MAKE WHICH TYPE?" :: ACCEPT AT(20,17)SIZE(1)VAL
    IDATE("0123"):K :: IF K=0 THEN W(I,5)=9 :: GOTO 280
260 IF HP(I)<K*5 THEN 250 ELSE HP(I)=HP(I)-K*5 :: W(I,5
    )=K*2+8
270 CALL TRANS(K*2+8.I$):: DISPLAY AT(13,5):I$ :: CALL
    WAIT(300)
280 HP(I)=HP(I)+5
290 NEXT I
300 !/ENT.DUNG./
310 CALL CLEAR :: AI,LV=1 :: FOR I=1 TO A
320 J,U(I)=0 :: DISPLAY AT(10,1):"WHICH WEAPON WOULD YO
```

```
U LIKE TO HOLD IN YOUR HAND,";PL$(I);"?" :: CALL TR
ANS(W(I,1),X$):: DISPLAY AT(13,1):X$
330 U(I)=-1 :: ACCEPT AT(13,1)SIZE(-6):X$ :: CALL TRANS
(U(I),X$):: IF U(I)=0 THEN 330
340 J=J+1 :: IF J<>5 THEN IF W(I,J)=U(I)THEN 360 ELSE 3
40
350 DISPLAY AT(22,1)BEEP:" SORRY ";PL$(I):"YOU DO NOT H
AVE ANY ";X$ :: GOTO 320
360 NEXT I
370 CALL CLEAR :: PRINT " YOU HAVE BEEN TRANSPORTED  TO
A DUNGEON BY AN EVIL MAGIC-USER. TO ESCAPE,"
380 PRINT " YOU MUST VENTURE TO THE 4THLEVEL OF THIS DU
NGEON. IT IS ONLY THERE THAT YOU WILL    FIND A TUN
NEL OUT !!"
390 PRINT " STAIRS ARE FOUND AT THE    NORTH END OF LEV
EL ONE, EASTEND OF 2ND LEVEL, SOUTH END OF 3RD AND
WEST OF 4TH.": :"GOOD LUCK..."
400 CALL KEY(0,K,S):: IF S=0 THEN 400
410 CALL SCREEN(13):: CALL CLEAR :: DISPLAY AT(1,1):"F=
FIGHT":" W=USE WEAPON   NORTH":" A=AIM":" R=RETREAT
":"C=CHANGE WEAPON":"X=EXCHANGE"
420 S=(A+3):: IF (U(5)>S AND LV=1)OR(U(6)>S AND LV=2)OR
(U(7)>S AND LV=3)OR(U(8)>S AND LV=4)THEN 570
430 IF PR$="R" THEN 620
440 !/HALLS/
450 CALL SCREEN(13):: PR$="P" :: RANDOMIZE :: ON INT(RN
D*7+1)GOTO 470,500,530,560,560,460,460
460 GOSUB 760 :: D$="" :: GOTO 450
470 A$="IN A CORRIDOR GOING" :: B$=" STRAIGHT." :: D$="
EX"
480 GOTO 650
490 !/SIDE P./
500 A$="NEAR A SIDE" :: B$=" PASSAGE GOING " :: CALL DI
R(2,C$,D$)
510 D$=D$&"XE" :: GOTO 650
520 !/P. TURNS/
530 A$="COMING TO A WALL, AND" :: B$=" THE PASSAGE TURN
S " :: CALL DIR(2,C$,D$)
540 D$=D$&"X" :: GOTO 650
550 !/DOOR/
560 A$="UP AGAINST A " :: B$="DOOR." :: PR$="R" :: D$="
EX" :: GOTO 650
570 IF LV=4 THEN 1430
580 A$="TO A STAIR GOING " :: RANDOMIZE :: IF INT(RND*2
)THEN B$="UP." ELSE B$="DOWN."
590 IF LV=4 AND B$="DOWN." THEN B$="UP." ELSE IF LV=1 A
ND B$="UP." THEN B$="DOWN."
600 D$="EX" :: GOTO 650
610 !/ROOM/
620 CALL SCREEN(LV+8):: IF INT(RND*2)=0 THEN GOSUB 760
ELSE CALL DES(11,X$):: RANDOMIZE :: IF INT(RND*2)TH
EN A$=" ROOM." ELSE A$=" CHAMBER."
630 A$="IN A "&X$&A$ :: RANDOMIZE :: I=INT(RND*4+1):: B
$="IT HAS " :: IF I>1 THEN B$=B$&"SEVERAL DOORS." E
LSE B$=B$&"ONE DOOR."
```

```
640 FOR J=1 TO I :: CALL DIR(4,X$,Y$):: C$=X$&","&C$ ::
    D$=D$&Y$ :: NEXT J :: C$=SEG$(C$,1,LEN(C$)-1):: PR$
    ="P"
650 CALL VCHAR(3,21,Z+96)
660 CALL DES(9,X$):: DISPLAY AT(18,1):"YOU ARE ";X$;" "
    ;A$;B$;C$: : : : :: DISPLAY AT(24,1):"WHICH WAY WIL
    L YOU WALK?" :: ACCEPT AT(24,25)SIZE(1)VALIDATE(D$)
    :I$
670 IF I$="" THEN 650 ELSE IF PR$="R" AND I$="X" THEN P
    R$="P"
680 IF B$="UP." AND I$="E" THEN LV=LV-1 ELSE IF B$="DOW
    N." AND I$="E" THEN LV=LV+1
690 GOSUB 1370 :: A$,B$,C$,D$=""
700 GOTO 420
710 !/MONSTERS/
720 DATA KOLBOLDS,1,GOBLINS,1,ORCS,1,SKELETONS,1,ZOMBIE
    S,1,GNOLLS,2,STIRGES,1,FIRE BEETLES,2
730 DATA HOBGOBLINS,2,SNAKES,2,GHOULS,3,HUGE SPIDERS,2,
    LARGE SPIDERS,3,LIZARD MEN,3,TROGLODYTES,2,GIANT TO
    ADS,2
740 DATA CREEPY CRAWLERS,4,DOPPLEGANGERS,4,WITCHES,3,GI
    ANT COBRAS,3,EVIL WIZARDS,4,BUGBEARS,3,WOLVES,3
750 DATA DRAGONS,6,TROLLS,5,SPECTRES,5,HARPIES,4,MUMMIE
    S,5,MINOTAURS,4,OGRES,4,HYDRAS,5
760 RANDOMIZE :: MST=INT(RND*A+RND*6-LV):: AC=INT(RND*3
    )+LV+9 :: IF MST<1 OR MST>6 THEN MST=1
770 IF LV=1 THEN RESTORE 720 ELSE IF LV=2 THEN RESTORE
    730 ELSE IF LV=3 THEN RESTORE 740 ELSE RESTORE 750
780 RANDOMIZE :: FOR I=1 TO INT(RND*8)+1 :: READ M$,J :
    : NEXT I :: B$=SEG$(M$,1,LEN(M$)-1):: I=0
790 IF MST=1 THEN D$="IS A " :: M$=B$ ELSE D$="ARE "
800 DISPLAY AT(18,1):"THERE ";D$;M$;" IN HERE!": : : :
    :: GOTO 830
810 !//(S)ESCAPE//
820 FOR S=220 TO 110 STEP -10 :: CALL SOUND(100,S,0)::
    NEXT S :: RANDOMIZE :: K=INT(RND*2):: RETURN
830 I=I+1 :: MH=0 :: FOR S=1 TO J :: MH=MH+INT(RND*6+LV
    ):: NEXT S :: MS=INT(MH/2)
840 DP=INT(RND*(J+LV))+1
850 DISPLAY AT(23,1): :"WHAT WILL YOU DO?F" :: ACCEPT A
    T(24,18)SIZE(-1)VALIDATE("FRXC"):I$
860 IF I$="R" THEN GOSUB 820 :: IF K=0 THEN CALL HCHAR(
    8,1,32,384):: RETURN
870 IF I$<>"C" THEN 900 ELSE IF A=1 THEN AP=1 :: GOTO 1
    130 ELSE DISPLAY AT(24,1)BEEP:"WHO WILL CHANGE?"
880 ACCEPT AT(24,17):I$ :: S=0
890 S=S+1 :: IF I$=PL$(S)THEN AP=S :: GOTO 1130 ELSE IF
    S=A THEN 880 ELSE 890
900 IF I$<>"X" THEN 980 ELSE IF A=1 THEN 850 ELSE DISPL
    AY AT(23,1):"WHO WILL EXCHANGE WITH WHOM?": : :: AC
    CEPT AT(24,1)SIZE(7):I$ :: S=0 :: IF I$="" THEN 850
910 S=S+1 :: IF I$=PL$(S)THEN AP=S ELSE IF S=A THEN 900
    ELSE 910
920 CALL TRANS(U(AP),A$):: DISPLAY AT(23,1):A$ :: ACCEP
    T AT(24,15)SIZE(7):I$ :: K,S=0 :: IF I$="" THEN 850
```

```
930 S=S+1 :: IF I$<>PL$(S)THEN IF S=A THEN 920 ELSE 930
940 K=K+1 :: IF U(S)=W(S,K)THEN W(S,K)=U(AP):: K=0 ELSE
    940
950 K=K+1 :: IF U(AP)=W(AP,K)THEN W(AP,K)=U(S)ELSE 950
960 IF U(S)=0 THEN 970 ELSE CALL TRANS(U(S),I$):: DISPL
    AY AT(23,15):I$ :: CALL WAIT(200):: DISPLAY AT(23,1
    ):I$;TAB(15);A$ :: CALL WAIT(250)
970 K=U(AP):: U(AP)=U(S):: U(S)=K :: GOTO 850
980 IF A=1 THEN 1020 ELSE DISPLAY AT(24,1)BEEP:"WHO WIL
    L ATTACK? (!):"
990 ACCEPT AT(24,22):I$ :: S=0 :: IF I$="!" THEN 1010
1000 S=S+1 :: IF I$=PL$(S)THEN AP=S :: GOTO 1030 ELSE I
     F S=A THEN 900 ELSE 1000
1010 CALL CLEAR :: PRINT "NICE TRY, BUT THEN NONE OF  Y
     OU WERE VERY SMART." :: STOP
1020 AP=1 :: IF PL$(1)=(CHR$(27)&"1")THEN 1010
1030 !/FIGHT/
1040 CALL TRANS(U(AP),D$):: DISPLAY AT(8,1):PL$(AP);TAB
     (13);B$;" #";STR$(MST+1-I);RPT$("-",LEN(PL$(AP)));
     TAB(13);RPT$("-",LEN(B$)+3)
1050 DISPLAY AT(10,1):"DAMAGE:";STR$(U(AP));TAB(13);"DA
     MAGE:";STR$(DP):"AC:";STR$(W(AP,5));TAB(13);"AC:";
     STR$(AC):"HP:";STR$(HP(AP));TAB(13);"HP:";STR$(MH)
1060 DISPLAY AT(13,1):"FORCE:0";TAB(13);"FORCE:0":"USIN
     G ";D$
1070 DISPLAY AT(20,1)BEEP:"YOU BOTH SWING AT EACH OTHER
     ":PL$(AP);",PRESS 'SPACE'..."
1080 CALL KEY(0,K,S):: IF K<>32 THEN 1080 ELSE GOSUB 13
     30
1090 DISPLAY AT(20,1):M$;" HAS A FORCE OF...": : :: GOS
     UB 1350
1100 IF F>K THEN 1270
1110 DISPLAY AT(20,1)BEEP:PL$(AP);" IS FASTER,": :: CAL
     L WAIT(100)
1120 IF U(AP)>0 THEN 1170 ELSE DISPLAY AT(20,1):"YOU CA
     NNOT HIT":"YOUR WEAPON IS BROKEN!":"YOU TRY TO ESC
     APE..." :: GOSUB 820 :: IF K THEN 1270
1130 S=0 :: FOR K=1 TO 4 :: CALL TRANS(W(AP,K),D$):: DI
     SPLAY AT(19+K,1):D$ :: IF D$="" THEN S=S+1
1140 NEXT K :: IF S=4 THEN 850 ELSE DISPLAY AT(24,1):"W
     HICH NEXT?"
1150 ACCEPT AT(24,12):D$ :: U(AP)=-1 :: CALL TRANS(U(AP
     ),D$):: IF U(AP)=0 THEN 1150 ELSE S=0
1160 S=S+1 :: IF U(AP)=W(AP,S)THEN CALL HCHAR(20,1,32,1
     28):: GOTO 850 ELSE IF S>4 THEN 1150 ELSE 1160
1170 DISPLAY AT(20,1):"YOU ATTACK...W": : : : : :: ACCE
     PT AT(20,14)SIZE(-1)VALIDATE("WRA"):I$ :: CALL SOU
     ND(300,550,9,-6,9):: IF I$="A" THEN 1190 ELSE IF I
     $="R" THEN GOSUB 820 ELSE 1200
1180 IF K=0 THEN GOTO 850 ELSE DISPLAY AT(20,1):"YOU WE
     RE NOT FAST ENOUGH..." :: CALL WAIT(200):: GOTO 12
     70
1190 AI=8 :: DISPLAY AT(13,7)SIZE(3):"AIM" :: CALL SOUN
     D(1000,440,15):: GOTO 1270
1200 GOSUB 1330 :: IF K=20 THEN 1210 ELSE IF K=1 THEN 1
```

```
250 ELSE IF K<AC THEN DISPLAY AT(20,1):"YOU MISSED
    ." :: GOTO 1270 ELSE 1220
1210 AI=3 :: FOR S=1 TO 8 :: CALL SOUND(100,-I,0,110,5,
    110,5,440,5):: NEXT S
1220 DISPLAY AT(20,1):"YOU HIT IT!" :: CALL SAY("YOU HI
    T IT"):: CALL SOUND(200,220,0):: CALL SOUND(310,44
    0,0):: CALL SOUND(430,550,0)
1230 MH=MH-INT(RND*(U(AP)*AI)+3)*(AI):: AI=1
1240 DISPLAY AT(12,16):STR$(MH):: CALL WAIT(400):: IF M
    H<1 THEN DISPLAY AT(20,1):"YOU KILLED THE ";B$: :
    : : :: GOTO 1360 ELSE 1270
1250 CALL SOUND(300,-7,9):: CALL SOUND(200,-6,6):: CALL
    SOUND(300,-5,2):: DISPLAY AT(21,1):"YOU BROKE YOUR
    ";D$ :: S=0
1260 S=S+1 :: IF U(AP)=W(AP,S)THEN W(AP,S),U(AP)=0 ELSE
    IF S<5 THEN 1260
1270 DISPLAY AT(20,1)BEEP:B$;" ATTACKS...": :: CALL WAI
    T(300):: CALL SOUND(100,110,0,-6,9):: GOSUB 1350 :
    : IF F>1 THEN 1290
1280 MH=MH-DP :: CALL SOUND(1000,-5,0):: DISPLAY AT(12,
    16):STR$(MH):: DISPLAY AT(20,1):B$;" ACCIDENTALLY
    HIT":"ITSELF!!" :: CALL WAIT(300):: IF MH>0 THEN 1
    120 ELSE 1240
1290 IF F<W(AP,5)THEN DISPLAY AT(20,1):B$;" MISSED." ::
    CALL WAIT(300):: GOTO 1120
1300 CALL SAY("IT GOT YOU"):: CALL SOUND(300,330,0,110,
    7):: CALL SOUND(400,220,0,110,7):: CALL SOUND(500,
    110,0):: HP(AP)=HP(AP)-DP
1310 DISPLAY AT(12,4)SIZE(2):STR$(HP(AP)):: IF HP(AP)>0
    THEN 1120 ELSE DISPLAY AT(20,1):PL$(AP);" IS DEAD.
    " :: PL$(AP)=CHR$(27)&STR$(AP)
1320 CALL SAY("GOOD+BYE+TO+YOU"):: CALL SOUND(1000,440,
    0,-4,0):: CALL SOUND(500,110,0):: GOTO 900
1330 K=INT(RND*19+AI+1):: IF K=9 AND AI=8 THEN K=1
1340 CALL SOUND(100,440,0,-5,9):: DISPLAY AT(13,7)SIZE(
    3):STR$(K):: AI=1 :: RETURN
1350 RANDOMIZE :: F=INT(RND*20+1):: CALL SOUND(100,220,
    0,-5,9):: DISPLAY AT(13,19):STR$(F):: RETURN
1360 HP(AP)=HP(AP)+MS :: DISPLAY AT(12,4)SIZE(2):STR$(H
    P(AP)):: CALL WAIT(500):: IF I<MST THEN 830 ELSE D
    ISPLAY AT(8,1): : : : : : : : : : : : :: RETURN
1370 IF I$<>"X" THEN 1380 ELSE IF Z<2 THEN Z=Z+2 ELSE Z
    =Z-2
1380 IF I$="D" THEN Z=Z+1 ELSE IF I$="S" THEN Z=Z-1
1390 IF Z>3 THEN Z=0 ELSE IF Z<0 THEN Z=3
1400 U(Z+5)=U(Z+5)+1 :: IF U(5)-U(7)<0 THEN U(7)=U(7)-U
    (5):: U(5)=0 ELSE U(5)=U(5)-U(7):: U(7)=0
1410 IF U(6)-U(8)<0 THEN U(8)=U(8)-U(6):: U(6)=0 ELSE U
    (6)=U(6)-U(8):: U(8)=0
1420 RETURN
1430 CALL SCREEN(8):: CALL CLEAR :: CALL SOUND(4000,440
    ,2,550,2,330,2,-1,3):: PRINT "YOU MADE IT OUT!!":
    : : : : :"THAT WAS VERY CLEVER!!" :: END
1440 SUB WAIT(A):: FOR D=1 TO A :: NEXT D :: SUBEND
```

```
1450 SUB TRANS(A,A$):: IF A=-1 THEN 1530
1460 IF A=10 THEN A$="WOODEN SHIELD" :: SUBEXIT
1470 IF A=12 THEN A$="SMALL SHIELD" :: SUBEXIT
1480 IF A=14 THEN A$="LARGE SHIELD" :: SUBEXIT
1490 IF A=6 THEN A$="SWORD" :: SUBEXIT
1500 IF A=9 THEN A$="AXE" :: SUBEXIT
1510 IF A=3 THEN A$="DAGGER" :: SUBEXIT
1520 IF A=0 THEN A$=""
1530 IF A$="SWORD" THEN A=6 :: SUBEXIT
1540 IF A$="AXE" THEN A=9 :: SUBEXIT
1550 IF A$="DAGGER" THEN A=3 ELSE A=0
1560 SUBEND
1570 SUB DIR(A,A$,B$):: RESTORE 1580 :: RANDOMIZE :: FO
     R I=1 TO INT(RND*A+1):: READ A$,B$ :: NEXT I
1580 DATA RIGHT,D,LEFT,S,FRONT,E,BACK,X
1590 SUBEND
1600 SUB DES(A,A$):: IF A=9 THEN RESTORE 1620 ELSE REST
     ORE 1630
1610 RANDOMIZE :: FOR I=1 TO INT(RND*A+1):: READ A$ ::
     NEXT I
1620 DATA NOW,"","","",WALKING,NOW,WALKING,MOVING,STAND
     ING
1630 DATA VERY DIRTY,LARGE,DARK,BRIGHT,MUSTY,DUSTY,SMAL
     L,MEDIUM SIZED,RECTANGULAR,ROUND,GIANT
1640 SUBEND
```

CHAPTER 19

Help

(EXTENDED BASIC)

Introduction

This is a type of hangman game. The game consists of two players who try to guess a hidden word which their opponent has entered in the computer. One at the time, each player secretly inputs his or her word into the computer so that the word can be guessed during play. Two man-like shapes represent the two players on the screen. A wrong guess by a player will move his man some steps closer to the edge of doom, which is an opening in the word that is to be guessed. To prevent their falling through the opening, the players are forced to think hard so they can guess each letter and close the open space(s). The concentration and memory needed during play is excellent in building and enriching each player's vocabulary.

This is a "learning while playing" game.

Features

1. This is a game designed for play by two players.
2. With the use of the optional speech synthesizer, you can hear the pronunciation of each letter that a player enters on his turn.
3. A graphical display of a man's shape is shown on two paths. When wrong letters are entered, one shape moves forward on its path. The game is lost, the figure that represents the losing player falls through the opening left by a missing letter and into the sea below.
4. The program also keeps track of the letters used by both players and displays them on the screen.

How To Play, Step-By-Step

1. Run the program. No instructions are written for the players because of the simplicity of the game.
2. Enter the names of both players when asked.
3. The red player enters his word in the blue player's absence and vice versa. A player can only input a maximum of 10 letters for a word.
4. During his turn, a player will guess a letter. If the word contains that letter, the computer will generate a rock to fill the gap at the specific location of that letter in the word. If there are two identical letters in the word, two rocks will appear to fill two gaps.
5. When one of the words has been correctly guessed, both men begin to walk down the path. The loser will fall through a hole in his unfinished bridge of missing letters.
6. Game ends when one player falls into the sea.

Program Explanation

0010-0040	Program name.
0050-0220	Define characters and colors.
0230-0260	Opening screen.
0270-0360	Ask for names and words from each player and display the playscreen.
0370-0380	Main loop; alternate turns to guess.
0390	If any word was finished (WAL=1), go to walk routine.
0400-0460	Ask for letter from appropriate player.
0470-0490	If letter inputted was used, go to to walk routine.
0500-0560	Right letter. Display letter(s) in correct position on screen.
0570-0600	Set WAL to 1 if word is finished, else go to line 370 (main loop). Walk routine.
0610-0750	Fall routine (ending). Ask for replay.
0760	Delay subprogram.

HELP
[EXTENDED BASIC]

```
10 !////////
20 !/ HELP /
30 !////////
40 !
50 !MADE 5/7/82
60 !//SHAPES//
70 DATA 0605060C0E1F3D346C0C0C0C0C0C0E0000000000000000808
   00
80 DATA 0605060C0C0C0C0C0C0E1B1B313038000000000000000000
   0000000A0C08
90 DATA 0605060C0C1E1E160C1C1E3676261700000000000000000000
   00
100 DATA 6121331E0C4C241E0D0C120C0000000080000000000000
    00008
110 DATA 1831331E0C0C0C1F688C120C0000000080800000000C0
    000
120 DATA 000000182C77F9DF8E0E0505050E1EFF00000000E0ECD6
    EA848000000080C0FF
130 DATA 0000000000000000019468D97050B1BFF00000000000000
    0088CC6C040080C0FF
140 DATA 0000000000000000000000009307FFF00000000000000
    00000000C000C8E6FF
150 !//MAKE SHAPES//
160 CALL CLEAR :: OPTION BASE 1 :: DIM N$(2),W$(4),X(4)
    ,Y(2):: CALL INIT :: CALL LOAD(-31878,2)
170 FOR I=96 TO 124 STEP 4 :: READ A$ :: CALL CHAR(I,A$
    ):: NEXT I
180 CALL CHAR(128,"FFFFFFFF",136,"FFFFFFFF")
190 CALL CHAR(129,"0000000E197CFEFF000000060D1C3EFF")
200 CALL CHAR(33,"0000004021925AFF0000402459AA3EFF")
210 CALL CHAR(60,"0000003C42DBA5FF")
220 CALL COLOR(1,3,1,13,5,1,14,9,1)
230 !//1ST SCREEN//
240 N$(1)=CHR$(129)&CHR$(130):: CALL SCREEN(11):: DISPL
    AY AT(7,9):RPT$(N$(1),5):TAB(9);N$(1);" HELP ";N$(1
    ):TAB(9);RPT$(N$(1),5)
250 CALL SAY("HELP")
260 CALL WAIT(2)
270 !//START//
280 W$(3),W$(4)="" :: CALL CLEAR :: CALL MAGNIFY(4):: C
    ALL SPRITE(#1,96,5,5,40,#2,96,9,5,184):: WAL=0
285 X(3),X(4)=0
290 PRINT RPT$(CHR$(129)&CHR$(130),14);
300 DISPLAY AT(23,1)BEEP:"ENTER BLUE'S NAME." :: CALL S
    AY("BLUE,WHAT+IS+YOUR+NAME"):: ACCEPT AT(1,3)SIZE(8
    ):N$(1)
310 DISPLAY AT(23,7)BEEP:"RED'S NAME." :: CALL SAY("RED
    ,WHAT+IS+YOUR+NAME"):: ACCEPT AT(1,21)SIZE(8):N$(2)
320 CALL MAGNIFY(3):: DISPLAY AT(8,1):RPT$(CHR$(33)&CHR
    $(34),8): : : : :RPT$(CHR$(34)&CHR$(33),8):: CALL L
    OCATE(#1,49,24,#2,89,24)
330 DISPLAY AT(2,3):RPT$(CHR$(128),LEN(N$(1)));TAB(21);
    RPT$(CHR$(136),LEN(N$(2)))
```

```
340 DISPLAY AT(23,7):N$(2);"'S WORD." :: CALL SAY("WHAT
    IS YOUR WORD,RED"):: ACCEPT AT(9,17)SIZE(11):W$(1):
    : DISPLAY AT(8,17+LEN(W$(1)))BEEP:RPT$(CHR$(34),28-
    (17+LEN(W$(1)))): :
350 DISPLAY AT(23,7):N$(1);"'S WORD." :: CALL SAY("WHAT
    IS YOUR WORD,BLUE"):: ACCEPT AT(14,17)SIZE(11):W$(2
    ):: DISPLAY AT(13,LEN(W$(2))+17)BEEP:RPT$(CHR$(33),
    28-(17+LEN(W$(2)))): :
360 DISPLAY AT(23,1): :: RANDOMIZE :: T=INT(RND*2+1)
370 !//LOOP//
380 IF T=1 THEN T=2 :: Z=18 ELSE T=1 :: Z=3
390 IF WAL=1 THEN 580
400 DISPLAY AT(17,1): : : :: DISPLAY AT(17,Z):N$(T)&","
    :TAB(Z);"LETTER?" :: ACCEPT AT(18,Z+8)SIZE(1)BEEP:L
    $
410 IF L$="" THEN 580
420 IF POS(W$(T),L$,1)=0 THEN 430 ELSE IF POS(W$(T+2),L
    $,1)=0 THEN 510 ELSE 480
430 IF POS(W$(T+2),L$,1)THEN 480 ELSE CALL SAY("THERE+I
    S+NO "&L$)
440 CALL SOUND(200,110,0,-7,5):: CALL VCHAR(X(T)+19,Y(T
    )+Z,ASC(L$))
450 Y(T)=Y(T)+1 :: IF Y(T)>7 THEN X(T)=X(T)+1 :: Y(T)=0
    :: IF X(T)>5 THEN X(T)=0 :: Y(T)=0
460 W$(T+2)=W$(T+2)&L$ :: GOTO 580
470 !/USED LETTER/
480 CALL SAY("YOU+CAN+NOT+USE "&L$&" AGAIN")
490 GOTO 580
500 !/RIGHT LETTER/
510 CALL SAY(L$&" IS+IN+THE1+WORD"):: W$(T+2)=W$(T+2)&L
    $
520 CALL SOUND(300,330,0,-6,9):: CALL SOUND(500,440,0,-
    5,9)
530 I=1
540 P=POS(W$(T),L$,I):: IF P=0 THEN 550 ELSE CALL VCHAR
    (T*5+4,P+18,ASC(L$)):: CALL VCHAR(5*T+3,18+P,60)::
    X(T+2)=X(T+2)+1 :: I=P+1 :: GOTO 540
550 IF X(T+2)<>LEN(W$(T))THEN 380
560 WAL=1 :: GOTO 380
570 !/WALK/
580 CALL POSITION(#T,X1,Y1):: CALL PATTERN(#T,100):: CA
    LL LOCATE(#T,X1,Y1+5):: CALL WAIT(.4)
590 CALL PATTERN(#T,104):: CALL LOCATE(#T,X1,Y1+10):: C
    ALL WAIT(.2):: CALL GCHAR(T*5+3,INT((Y1+10)/8)+1,CH
    ):: IF CH=32 THEN 620
600 CALL PATTERN(#T,96):: CALL LOCATE(#T,X1,Y1+16):: CA
    LL GCHAR(T*5+3,INT((Y1+16)/8)+1,CH):: IF CH=32 THEN
     620 ELSE 380
610 !/FALL/
620 CALL MOTION(#T,1,0)
630 CALL SAY("UHOH,A1+HELP"):: I=2000 :: CALL MOTION(#T
    ,7,0)
640 IF Z=1 THEN Z=2 ELSE Z=1
650 I=I-25 :: CALL SOUND(-500,I,5)
660 CALL PATTERN(#T,Z*4+104):: CALL POSITION(#T,X1,Y1):
```

```
    : IF X1<183 THEN 640
670 CALL COLOR(#T,5):: CALL MOTION(#T,0,0):: CALL PATTE
    RN(#T,116):: CALL LOCATE(#T,174,Y1)
680 CALL SOUND(300,110,10,-5,10):: CALL SOUND(400,-5,23
    ):: CALL PATTERN(#T,120)
690 CALL SOUND(300,-5,23):: CALL PATTERN(#T,116):: CALL
    SOUND(200,-5,24):: CALL PATTERN(#T,120):: CALL SOUN
    D(400,-5,25)
700 CALL SOUND(500,-5,28):: CALL PATTERN(#T,124):: CALL
    SOUND(200,-5,29):: CALL DELSPRITE(#T)
710 DISPLAY AT(3,1):"HOPE YOU CAN SWIM,";N$(T):: IF T=1
    THEN T=2 ELSE T=1
720 DISPLAY AT(5,1):N$(T);" IS THE WINNER."
730 DISPLAY AT(9,17):W$(1):: DISPLAY AT(14,17)BEEP:W$(2
    )
740 DISPLAY AT(23,1):"PRESS ENTER TO PLAY AGAIN."
750 CALL KEY(0,K,S):: IF K=-1 THEN 750 ELSE CALL CLEAR
    :: IF K=13 THEN 280 ELSE STOP
760 SUB WAIT(A):: CALL SOUND(A*1000,44733,30):: CALL SO
    UND(1,110,30):: SUBEND
```

CHAPTER 20

Black Tunnel

(EXTENDED BASIC)

Introduction

This game is designed to improve the skill and reaction of the player as he maneuvers a spaceship through a tunnel in space. This Black Tunnel is the only way that leads to victory. A stream of space vehicles are flowing at different speeds in an opposite direction from yours; a collision means total destruction. Using four directional keys, you try to move your vehicle up, down, forward, or backward, depending on the speed and location of the other vehicles, to prevent a collision so you can advance and reach the far end (left end) of the tunnel. It is as simple as that but your attention and your skill are required, and sometimes, everything depends on luck. There may not always be a way to get through. It is fun.

Features

1. Graphic displays, moving sprites, colors, and sound are selected for the game.
2. Explosions from a collision are caused in three different ways: by hitting other vehicles, by colliding with the upper tunnel wall, and by colliding with the lower tunnel wall.
3. The three levels for each play are shown by three tunnels, of different colors, that appear one after another. A different tunnel appears after each successful attempt at "getting through." The space vehicles that appear will move faster for each succeeding tunnel.
4. The player is given four vehicles to play with for each game; after every three successful passages through the tunnel, you will come to the victorious ending. Your supply of vehicles is displayed at the top of the screen.
5. The game is designed to use the directional keys (E, D, X, S) to play the game.

How To Play, Step-By-Step

1. Run the program. Press Y for instructions, N for none. The tunnel shape and color will be displayed on the screen ready for playing.
2. Using the four directional keys (E, D, X, S), you will try to maneuver your vehicle through the tunnel without colliding with the other vehicles or hitting the tunnel walls.
3. After you are safely through the first tunnel, a second tunnel will appear with a different tunnel color and with faster space vehicles. You will then be at a more difficult playing level. The third tunnel is the last one and is also the hardest one to go through.
4. Game ends when you have gone through three tunnels and won, or when all four of your ships are destroyed.

Program Explanation

0100-0040	Program name.
0050-0090	Define characters.
0100	Clear screen and magnify to single-size magnification.
0110-0160	Display title screen and ask for instructions.
0170-0220	Instructions.
0230	Clear screen and turn it black. Set variables.
0240	Draw tunnel.
0260-0310	Put ship sprite on, display score, put object sprites on, and move them randomly.
0330	Check for keypress.
0340-0370	Check which key and move ship that direction.
0380	Check for colliding with objects.
0390	Check for hitting wall.
0400	Check for reaching left and right side of tunnel.
0420-0440	Damage animation for hitting objects; subtract a ship.
0450-0600	Subprograms.
0460-0510	Damage animation for hitting wall.
0520-0550	Subprogram for passing a tunnel.
0560-0600	End game when no more ships are available; ask for relay.

BLACK TUNNEL
[EXTENDED BASIC]

```
10 ! ////////////////
20 ! / BLACK TUNNEL /
30 ! ////////////////
40 !
50 !///INITIALIZE///
60 CALL CHAR(96,"FF818181818181FF",128,"00020F2767C1")
70 CALL CHAR(100,"0000031FF90E",104,"0000E0FFE0",124,"0
   080472F2FC701")
80 CALL COLOR(9,16,1,10,5,1)
90 RANDOMIZE
100 CALL CLEAR :: CALL MAGNIFY(2)
110 !///TITLE///
120 CALL SCREEN(13):: PRINT RPT$(CHR$(100)&" ",14):: PR
    INT :TAB(8);"BLACK TUNNEL " :: PRINT :RPT$(CHR$(100)
    &" ",14): : : : : : : : :
130 CALL SPRITE(#1,100,5,79,210,0,-8)
140 PRINT ;"DO YOU NEED INSTRUCTIONS     (Y/N)"
150 CALL KEY(0,K,S):: IF S=0 THEN 150 ELSE IF K<>89 THE
    N 230
160 CALL DELSPRITE(#1)
170 !///INSTRUCTIONS///
180 PRINT : : :" YOU ARE IN A TUNNEL IN      SPACE.  THI
    S TUNNEL IS FULL OF ROCKETS THAT ARE KEEPING YOU FR
    OM GETTING ACROSS TO"
190 PRINT "THE OTHER END TO GET OUT.": :" YOU MUST GET
    OUT WITH THE  FOUR SHIPS THAT YOU HAVE.": :"  USING
    THE ARROW KEYS"
200 PRINT "(E,D,X,S), TRY TO AVOID THE ROCKETS AND GET
    THROUGH THE TUNNEL THREE TIMES.": :" BE CAREFUL! SO
    METIMES THE"
210 PRINT "TUNNEL MAY SEEM TOO SMALL   FOR YOU.": : :;"
    PRESS ANY KEY..."
220 CALL KEY(0,K,S):: IF S=0 THEN 220
230 SH=4 :: SP=4 :: CALL CLEAR :: CALL SCREEN(2):: CALL
    COLOR(9,16,1)
240 CALL HCHAR(16,1,96,608):: CALL VCHAR(11,3,96,5)
250 !///START///
260 !//SCREEN SET UP//
270 CALL SPRITE(#1,100,5,88,240,0,0)
280 CALL HCHAR(1,1,32,64):: DISPLAY AT(1,12):RPT$(CHR$(
    100)&" ",SH)&"       "
290 RESTORE 300 :: FOR I=2 TO 4 :: READ A :: CALL SPRIT
    E(#I,104,INT(RND*13+3),A,1):: CALL SPRITE(#I+3,104,
    INT(RND*13+3),A,INT(RND*60+RND*30+30)):: NEXT I
300 DATA 80,93,106
310 FOR I=2 TO 4 :: A=INT(RND*SP+RND*10+1):: CALL MOTIO
    N(#I,0,A):: CALL MOTION(# I+3,0,A):: NEXT I
320 !//GAME LOOP//
330 CALL KEY(0,K,S):: IF S=0 THEN 380
340 IF K<>69 THEN 350 ELSE CALL MOTION(#1,-8,0):: GOTO
    380
```

```
350 IF K<>68 THEN 360 ELSE CALL MOTION(#1,0,8):: GOTO 3
    80
360 IF K<>88 THEN 370 ELSE CALL MOTION(#1,8,0):: GOTO 3
    80
370 IF K<>83 THEN 380 ELSE CALL MOTION(#1,0,-8)
380 CALL COINC(ALL,HIT):: IF HIT THEN 420 ELSE CALL COI
    NC(ALL,HIT):: IF HIT THEN 420
390 CALL POSITION(#1,X,Y):: IF X>111 OR X<75 THEN CALL
    HBOARDER(SH,X,Y):: GOTO 430
400 IF Y<12 THEN CALL WIN(SP)ELSE IF Y>245 THEN 420 ELS
    E 330
410 GOTO 270
420 CALL SOUND(1000,-7,0):: CALL SPRITE(#1,100,7,X,Y,0,
    20):: FOR D=1 TO 25 :: NEXT D :: CALL PATTERN(#1,12
    4):: FOR D=1 TO 100 :: NEXT D
430 CALL DELSPRITE(ALL):: SH=SH-1 :: IF SH=0 THEN CALL
    END ELSE 270
440 GOTO 230
450 !//SUBPROGRAMS//
460 SUB HBOARDER(SH,X,Y)
470 IF X<113 THEN 500
480 CALL SOUND(1000,-7,0):: CALL SPRITE(#1,100,7,X,Y,-8
    ,-10):: FOR D=1 TO 25 :: NEXT D :: CALL PATTERN(#1,1
    28):: FOR D=1 TO 300 :: NEXT D :: CALL DELSPRITE(ALL
    )
490 SUBEXIT
500 CALL SOUND(1000,-7,0):: CALL SPRITE(#1,100,7,X,Y,8,
    -10):: FOR D=1 TO 25 :: NEXT D :: CALL PATTERN(#1,1
    28):: FOR D=1 TO 300 :: NEXT D :: CALL DELSPRITE(AL
    L)
510 SUBEND
520 SUB WIN(SP):: CALL SOUND(800,300,0):: CALL DELSPRIT
    E(ALL):: SP=SP+3 :: CALL COLOR(9,SP-1,1):: CALL SOU
    ND(1000,300,0,500,4)
530 IF SP<>13 THEN SUBEXIT ELSE CALL CLEAR :: CALL SCRE
    EN(16):: PRINT "        YOU'VE MADE IT !": : : : : :
    : : : : :
540 FOR I=110 TO 310 STEP 10 :: CALL SOUND(I-15,I,0)::
    NEXT I :: CALL SOUND(1200,400,0,200,5,-5,5):: FOR D
    =1 TO 100 :: NEXT D :: STOP
550 SUBEND
560 SUB END
570 CALL SCREEN(7):: PRINT : : : : : : : : : :TAB(7);
    "YOU HAVE LOST": : : : : : : : : : : :
580 CALL SOUND(600,290,3):: CALL SOUND(1200,110,0):: CA
    LL CLEAR
590 PRINT "TRY AGAIN (Y/N)?" :: INPUT Y$ :: IF Y$="N" T
    HEN CALL CLEAR :: STOP
600 SUBEND
```

CHAPTER 21

Meteor Rescue

(EXTENDED BASIC)

Introduction

This program is designed to create a simulation of activities on an imaginary planet which has been newly discovered by a space mission. A space base is anchored in an orbit around the planet which is full of uranium deposits. Workmen were transported to the planet to mine for the uranium. Unknowingly, the miners dug too deeply through the surface and caused the meteors to orbit in an asteroid belt around the planet. The planet is due to break apart at any time because of a chain reaction effect of the uranium core. Ten workers on the planet are waiting for the rescue. They have built a platform where the rescue spaceship can land. You are piloting the "EAGLE SPACESHIP" and are trying to penetrate the layers of deadly meteors in hopes of saving and bringing the miners back to the home base.

Be aware of the effects of the impact force caused by the collision of meteors on your spaceship. Three hits will cause the Eagle Spaceship to crash. Your ship has to land exactly on the platform to avoid a collision impact which counts as a hit. After each landing, a person will be rescued as he climbs into the spaceship, which, in turn, has to maneuver up and through the asteroid belt again to reach the home base. The planet will explode only when all the Eagle Landers are exhausted; then there is no hope of further rescue.

Hope you save everyone.

Features

1. The program uses moving sprites for spaceships and orbiting meteors. Sound and colors are used extensively to accentuate the scene.

2. The display, on the screen, of the remaining spaceships, the number of people saved, and the landing platforms are the major features of the program.
3. To animate the rescue mission, each time the spaceship lands on the rescue platform, a person will run out and climb into the ship; this person remains on the ship until you have succeeded in rescuing him or have failed.
4. A speech synthesizer can be used to make additional announcements to accentuate the activities. (If a speech synthesizer is not attached, no speech will be heard.)
5. You begin with three spaceships, but after playing for a while, a bonus of an extra spaceship can be acquired for every two men saved.
6. The spaceship can collide with either the asteroids or the platform on which it lands; three collisions of this type will cause the spaceship to crash.
7. Joysticks are required to play the game.
8. When the planet comes to its destruction, fast animation will end the game.

How To Play, Step-By-Step

1. Run the program. The computer will ask whether you need instructions or not. Press Y for yes or N for no.
2. If you do not have the memory expansion unit, change the CALL INIT, CALL LOAD (−31878,X) entries on lines 260 and 590 to REM.
3. Using a joystick, fly your spaceship from your home base (left corner of screen) through the asteroid belt and land on the rescue platform (at the lower middle part of the screen).
4. You will observe that three collisions with any of the cosmic rocks will destroy your spaceship. Try to land the spaceship exactly on the center of the platform; one mistake could damage your ship.
5. Use the button on your joystick to land your spaceship on the platforms or to stop it in midair.
6. When you perform a perfect landing, you will hear sounds. Then, a person will come running to the platform and will climb into the spaceship.
7. You are now ready to try to get back to your base, but be careful to avoid a collision.

8. For every two persons you save, you will get another spaceship as a bonus.
9. The game ends when all spaceships have been destroyed or when you have brought all 10 persons safely back to the base.

Program Explanation

0010-0040	Program name.
0050-0200	Define characters.
0210-0240	GOSUB 690 (to display first screen and the instructions). Then, set up stars and spaceship.
0250-0300	Set up rocks and GOSUB 670 (move them).
0310-0390	Draw random ground, put invisible sprite on ground (so spaceship cannot get to bottom through top). Define colors and set variables.
0400	Game loop.
0410-0420	Check joystick and move ship.
0430-0450	Check for hits. If hit, then add number of hit (3 hits will destroy the lander).
0460-0520	Fire button is pressed; check to see if lander is close enough to ground to land.
0530-0650	End messages; meteor destruction after ship breaks orbit.
0660-0670	Move rocks subroutine; increase rocks speed (difficulty).
0680-0770	Subroutine for first screen and the instructions.
0780-1020	User Defined Subprograms.
0790-0860	SUB LAND. Lander lands on meteor and miner runs out.
0870-0900	SUB LS. Lander lands on mother ship and miner runs out.
0910-1020	SUB BLOW. If lander scraped against rocks 3 times, it drops to ground (if carrying miner, he runs out). New ship appears at mother ship (if any are left).

METEOR RESCUE

[EXTENDED BASIC]

```
10 !/////////////////
20 !/ METEOR RESCUE /
30 !/////////////////
40 !
50 !///INITIALIZING///
60 CALL MAGNIFY(3):: CALL CLEAR :: DIM DI(6)
70 CALL CHARPAT(96,A$):: IF POS(A$,"02B0D",1)THEN 220
80 !/LANDING MODULE/
90 DATA 00002B0D1D1F1F053F23204040E000002810D0F0F8F8F8A
   0FCC404020207
100 !/MAN/
110 DATA 101438501828282,1050381430282808
120 !/ASTEROIDS/
130 DATA 00000000181F0F1B1B3F1F1F040000000000000C7CD8F8
    F0F874F8F830,00000000090F070D0505070701000000000000
    0080C0F8E4F8B8FC3EFE3E02
140 DATA 0000010203070707080B1E0E0D0E0E0440C0C0C0E0E0E0
    B8F8E0C0808
150 DATA 0018270F1B0A17694F2070F66FE77FFE50404082060A48
    C8BFEDCF766DE974
160 DATA 0000000000000000000001030201000100000000000000
    0000008040C0008
170 DATA 0000C995DD95D50000000000000000000000D312F352DB
    00
180 DATA FEFFFFFFFFFFFFFF01091D3F37230102000080C0E0F0F8
    FC80C89CFEF6E2C0A0
190 DATA FF00000000000000000000000000000FF
200 FOR I=96 TO 136 STEP 4 :: READ A$ :: CALL CHAR(I,A$
    &RPT$("0",64-LEN(A$))):: NEXT I
210 !///SCREEN SET UP///
220 GOSUB 690 :: CALL SCREEN(2)
230 CALL CLEAR :: FOR I=1 TO 30 :: CALL VCHAR(RND*19+2,
    RND*29+2,43):: NEXT I
240 CALL CHAR(47,"48386C6C384482",41,"FEFFF1E0E0C0C080"
    &RPT$("F",16)&"1",140,RPT$("F",64))
250 !//SET UP METEOR//
260 H=-1 :: CALL INIT :: CALL LOAD(-31878,16)
270 FOR I=56 TO 152 STEP 16 :: RANDOMIZE
280 F=INT(RND*3-1):: IF F THEN DI(INT((H+2)/2))=F :: F=
    F*INT(RND*SP+SP-3)ELSE 280
290 RANDOMIZE :: H=H+2 :: CALL SPRITE(#H,INT(RND*3+1)*4
    +104,RND*5+10,I-14,252,#H+1,INT(RND*3+1)*4+104,RND*
    5+10,I-14,RND*200+25):: NEXT I
300 GOSUB 670
310 !//GROUND//
320 FOR J=1 TO 32 :: CALL VCHAR(24,J,INT(RND*12+108)):
    NEXT J
330 CALL VCHAR(22,16,119,2):: CALL VCHAR(22,15,114):: C
    ALL VCHAR(22,17,116)
340 CALL HCHAR(4,1,42,5):: CALL VCHAR(4,6,41):: CALL VC
    HAR(1,1,143,2):: CALL VCHAR(1,2,143,3)
```

```
350 CALL HCHAR(1,22,47,2-SH)
360 MS=184 :: HI=12 :: FOR H=28 TO 19 STEP -2 :: MS=MS+
    1 :: FOR F=0 TO 1 :: CALL SPRITE(#H-F,136,1,MS,HI)::
    HI=HI+26 :: NEXT F :: NEXT H
370 CALL COLOR(1,16,1,2,16,1,10,12,1,11,1,12,12,15,1,14
    ,7,1,9,7,1):: FOR I=3 TO 8 :: CALL COLOR(I,16,1):: N
    EXT I
380 CALL SPRITE(#15,96,15,11,30,#17,128,3,24,26,#18,132
    ,7,17,10,#24,140,1,171,116)
390 HI,MS,F,H=0
400 !///GAME LOOP///
410 CALL KEY(1,X,Y):: IF Y THEN 470 ELSE CALL JOYST(1,Y
    ,X):: IF X=0 AND Y=0 THEN 430 ELSE XVEL=-X :: YVEL=
    Y :: CALL SOUND(300,110,18,-4,20)
420 CALL MOTION(#15,XVEL,YVEL)
430 CALL COINC(ALL,HIT):: IF HIT THEN 440 ELSE 410
440 CALL SOUND(200,-5,0):: HI=HI+1 :: IF HI>2 THEN CALL
    BLOW(SH,MS):: CALL HCHAR(1,22,32,9):: IF SH>2 THEN
    540 ELSE HI,MS=0 :: CALL HCHAR(1,22,47,2-SH)
450 GOTO 410
460 !//LAND?//
470 CALL MOTION(#15,0,0):: CALL SOUND(500,-5,20):: CALL
    COINC(#15,155,117,1,HIT):: CALL POSITION(#15,X,Y)::
    CALL SPRITE(#16,124,10,X,Y)
480 CALL COINC(ALL,HIT2):: IF HIT2 THEN CALL DELSPRITE(
    #16):: GOTO 440 ELSE CALL COINC(#15,11,31,1,HIT2)::
    IF HIT2 AND MS THEN CALL LS :: F=F+1 :: HI,MS=0 ELS
    E 510
490 IF F=10 THEN 550 ELSE IF F/2=INT(F/2)THEN GOSUB 670
    :: SH=SH-1 :: CALL HCHAR(1,22,47,2-SH)
500 GOTO 520
510 IF HIT=0 OR MS=1 THEN CALL DELSPRITE(#16):: GOTO 41
    0 ELSE CALL LAND :: MS=1
520 XVEL,YVEL=0 :: CALL HCHAR(1,9,100,F):: GOTO 410
530 !//END MESSAGES//
540 Y$="YOU HAVE FAILED. YOU'LL HAVE TO LEAVE THE REST
    HERE.   PRESS ANY KEY TO BREAK ORBIT. " :: GOTO 560
550 Y$="YOU HAVE SUCCEEDED IN SAVING THE MEN!    YOU AR
    E A HERO!!!!!       NOW BREAK ORBIT BY PRESSING ANY
    KEY.   HURRY!!!!!! "
560 Y$=RPT$(" ",27)&Y$ :: FOR I=1 TO LEN(Y$):: DISPLAY
    AT(9,1)BEEP:SEG$(Y$,I,28):: NEXT I
570 CALL KEY(0,K,S):: IF S=0 THEN 570 ELSE CALL DELSPRI
    TE(#15,#17,#18):: CALL PATTERN(#16,112,#15,114,#17,
    112,#18,120)
580 FOR I=6 TO 1 STEP -1 :: DISPLAY AT(4,1):SEG$("****)
    ",7-I,I):: CALL SOUND(-999,-7,0):: NEXT I :: CALL V
    CHAR(1,1,32,48)
590 IF F=10 THEN 620 ELSE CALL LOAD(-31878,28)
600 FOR I=1 TO 10-F :: CALL SPRITE(#15+I,104,7,178,20+I
    *20):: NEXT I
610 FOR J=1 TO 3 :: FOR I=1 TO 10-F :: CALL PATTERN(#15
    +I,100):: NEXT I :: CALL SAY("GOODBYE"):: FOR I=1 T
    O 10-F :: CALL PATTERN(#15+I,104):: NEXT I :: NEXT J
```

```
620 FOR I=1 TO 28 :: CALL SOUND(-999,-5,0):: CALL MOTIO
    N(#I,RND*127-63,RND*127-63):: NEXT I
630 FOR I=1 TO 30 :: CALL SOUND(-4250,-6,I):: NEXT I ::
    CALL DELSPRITE(ALL)
640 CALL HCHAR(22,1,32,96):: CALL SOUND(1,44733,0):: DI
    SPLAY AT(1,1):"PRESS ""ALPHA LOCK"" DOWN AND PRESS
    ""A"" FOR ANOTHER METEOR."
650 CALL KEY(0,K,S):: IF S=0 THEN 650 ELSE IF K<>65 THE
    N RUN "DSK1.LOAD" ELSE MS,SH,F,HI=0 :: GOTO 230
660 !//MOVE ROCKS//
670 SP=SP+1 :: FOR X=0 TO 13 STEP 2 :: RANDOMIZE :: Y=D
    I(INT(X/2))*(RND*SP+SP-3):: CALL MOTION(#X+1,0,Y,#X
    +2,0,Y):: NEXT X :: RETURN
680 !///INSTRUCTIONS///
690 PRINT TAB(7);"<<<<<<<<<<>>>>>>>>>":TAB(7);"< METEOR R
    ESCUE >":TAB(7);"<<<<<<<<<<>>>>>>>>>": : : : : : : :
    : : :
700 INPUT "NEED INSTRUCTIONS(Y/N)?":Y$
710 IF Y$<>"Y" THEN 770
720 PRINT : : :" YOU ARE ON A RESCUE MISSIONTO A METEOR
    WHICH IS ON THE VERGE OF ANNIHILATION.": :" YOUR MI
    SSION IS TO RESCUE"
730 PRINT "THE TEN MINERS ON THE METEORWITH THREE LANDE
    RS.": :" EACH LANDER CAN SUSTAIN    THREE HITS FROM
    THE FLYING  ROCKS.": :
740 PRINT " BE CAREFUL! IF YOU MANEUVERAND LAND WELL, Y
    OU'LL BE    SAFE, BUT IF YOU DON'T, A   HIT COULD B
    E FATAL!"
750 PRINT :" AN EXTRA LANDER WILL BE    BUILT FOR EACH
    TWO MINERS   SAVED. GOOD LUCK.": : :"PRESS ANY KEY";
760 CALL KEY(0,K,S):: IF S=0 THEN 760
770 SP=4 :: RETURN
780 !///SUBPROGRAMS///
790 SUB LAND :: CALL LOCATE(#15,155,117)
800 FOR I=0 TO 30 :: CALL SOUND(-100,200-I,I):: NEXT I
    :: CALL SPRITE(#16,100,7,178,1)
810 RANDOMIZE :: I=INT(RND*3-1):: IF I=0 THEN 810 ELSE
    CALL MOTION(#16,0,I*4)
820 IF I=104 THEN I=100 ELSE I=104
830 CALL PATTERN(#16,I):: CALL SOUND(-200,-4,15):: CALL
    COINC(#16,178,125,2,HIT):: IF HIT THEN CALL MOTION(
    #16,0,0)ELSE 820
840 CALL MOTION(#16,-2,0):: FOR X=1 TO 20 :: IF I=104 T
    HEN I=100 ELSE I=104
850 CALL PATTERN(#16,I):: CALL SOUND(80,-5,28):: NEXT X
    :: CALL DELSPRITE(#16)
860 SUBEND
870 SUB LS :: CALL LOCATE(#15,11,31):: CALL SPRITE(#16,
    100,7,19,35,0,-3)
880 IF I=100 THEN I=104 ELSE I=100
890 CALL PATTERN(#16,I):: CALL POSITION(#16,X,Y):: IF Y
    >7 THEN 880 ELSE CALL DELSPRITE(#16):: CALL SAY("#G
    OOD WORK#")
900 SUBEND
```

```
910 SUB BLOW(A,B):: CALL SOUND(500,-7,0):: CALL SOUND(3
    00,-6,0,110,3):: CALL SOUND(900,-5,20,110,20)
920 CALL MOTION(#15,3*(RND*3+3),RND*5-2)
930 CALL POSITION(#15,X,Y):: IF X<169 THEN CALL COINC(A
    LL,HIT):: IF HIT THEN CALL SOUND(60,-5,1):: GOTO 93
    0 ELSE 930 ELSE CALL MOTION(#15,0,0)
940 CALL POSITION(#15,X,Y):: CALL LOCATE(#15,11,1):: Y=
    Y/8+1 :: IF Y>30 THEN Y=30 ELSE IF Y<2 THEN Y=2
950 CALL VCHAR(23,Y,120):: CALL VCHAR(24,Y,121):: CALL
    VCHAR(23,Y+1,122):: CALL VCHAR(24,Y+1,123)
960 FOR X=0 TO 15 STEP 5 :: CALL SOUND(X*50+1,-7,X):: N
    EXT X :: FOR X=10 TO 30 :: CALL SOUND(-100,-7,X)::
    NEXT X
970 A=A+1 :: IF A>2 THEN SUBEXIT ELSE IF B=0 THEN 1000
    ELSE CALL SPRITE(#16,104,7,178,(Y-1)*8,0,2*SGN((Y-1
    )*8-122))
980 IF I=104 THEN I=100 ELSE I=104
990 CALL PATTERN(#16,I):: CALL SOUND(-200,-4,15):: CALL
    COINC(#16,178,256,3,HIT):: IF HIT THEN CALL DELSPRI
    TE(#16)ELSE 980
1000 CALL MOTION(#15,0,1)
1010 CALL POSITION(#15,X,Y):: IF Y>29 THEN CALL MOTION(
     #15,0,0)ELSE CALL SOUND(-200,330,9,-3,9):: GOTO 10
     10
1020 SUBEND
```

BOOK AND CASSETTE TAPE COMBINATION PACK

In the event that you purchased the combination book and tape cassette package, refer to the following regarding the contents and loading of the tape.

This high-quality cassette is ready to run without any modification, and contains every program you see in the book. With it, you'll be able to enjoy many additional hours running and modifying programs instead of spending that time manually keying-in each program from the book and looking for any typing errors afterward.

This tape has been recorded on two sides. Side 1 is a listing of the the programs exactly as they appear in the book. Many of the programs could contain additional entries to improve operation and provide additional features. If your interest is in programming, use this side of the tape. Then make your own additions and changes to the basic program so you can see just what happens when these changes are made. Side 2 is an "enhanced" version of the programs. This version contains additions to the basic programs to provide more error checking and easier operation for those users whose primary interest is not in programming. However, for the programmer, this side will also provide a handy learning tool. By comparing the two sides of the tape, you can see the enhancements made to provide for good programming practices and incorporate similar procedures in you own programs.

HOW TO LOAD THESE CASSETTE PROGRAMS INTO YOUR TI-99/4A

This tape will load according to the OLD instructions in your **TI-99/4A User's Reference Guide** (the one that came with your computer).

1. Connect and power up your TI-99/4A, a TV screen, and cassette recorder(s). Set the tape counter on CS1 to zero and insert the cassette. With your TI-99/4A in Command Mode, type **OLD CS1** and press ENTER.
2. Your screen will show a series of instructions, the first of which tells you to rewind the cassette. Instead of rewinding, proceed to the next step on this sheet.
3. Suppose the name of the program you want to load is **Cosmic Guns,** which should be ready to load when 065 appears on the tape counter. Using the "Fast Forward" control on Recorder CS1, run the tape ahead until the counter indicates 065. Stop the tape. Press ENTER.
4. Now proceed with the next screen instruction, which is PRESS

CASSETTE PLAY. Depress the "Play" button on the recorder. Then press ENTER on the computer.

5. Your screen will display READING as the program **Cosmic Guns** is loaded and checked. On completion, you'll see DATA OK, followed by PRESS CASSETTE STOP . . . THEN PRESS ENTER. Depress the "Stop" button on the recorder. Then press ENTER on the computer. Your program is now loaded into memory, ready for running.

FINDING PROGRAMS ON THE TAPE

The counter numbers will vary from recorder to recorder. For your convenience, the program locations on a given recorder for Side 1 are given. If you find your numbers agree, use these numbers when loading in programs. In the event your counter gives a different reading, the list can still be used to give the order of the programs and, through interpretation of your numbers, can help in finding the approximate location. Be sure to note any differences for your future reference.

CAUTION: Before you try to run any of the programs using Extended BASIC be sure to reread page 12 if you do not have the memory expansion unit. You will need to make the changes listed under "Memory Expansion" before the programs will run without the Memory Expansion unit.

PROGRAMS ON SIDE 1, WITH LOCATIONS

The programs in the ENHANCED VERSION of the tape (Side 2) are listed in the same sequence as above, but they will not be at the same tape location. (See the screen menu for approximate locations.)

More Books for TI-99/4A® Owners!

THE TI-99/4A USER'S GUIDE

You'll like this badly needed guide to all aspects of the TI-99/4A! Covers everything from system setup to expansion options, including languages, software, and peripherals. By Carol Ann Casciato and Donald J. Horsfall. 224 pages, 5½ × 8½, soft. ISBN 0-672-22071-7. © 1983.

Ask for No. 22071 ..... **$11.95**

TI-99/4A: 24 BASIC PROGRAMS

An inexpensive source of fun-and-useful, completely tested BASIC programs that take full advantage of your TI-99/4A's sound and graphics capabilities. Also covers fundamental programming commands, debugging, utilities, and more. By Carol Ann Casciato and Donald J. Horsfall.

BOOK ONLY: 160 pages, 5½ × 8½, comb. ISBN 0-672-22247-7. © 1983.

Ask for 22247 ..... **$12.95**

BOOK PLUS TAPE CASSETTE: Packed in 6 × 9 hardcover vinyl binder with cassette storage feature. ISBN 0-672-26172-3.

Ask for No. 26172 ..... **$19.95**

TI-99/4A: 51 FUN AND EDUCATIONAL PROGRAMS

Run all 51 BASIC programs as-is on the TI-99/4A or adapt to almost any other computer. Begin with easy, short programs and progress to long, more complex ones. Ideal for first-time computer users of any age. By Gil M. Schechter.

BOOK ONLY: 80 pages, 5½ × 8½, soft. ISBN 0-672-22192-6. © 1983.

Ask for 22192 ..... **$4.95**

BOOK PLUS TAPE CASSETTE: Packed in 6 × 9 hardcover vinyl binder with cassette storage feature. ISBN 0-672-26168-5.

Ask for No. 26168 ..... **$11.95**

ENTERTAINMENT GAMES IN TI BASIC AND EXTENDED BASIC

Highly organized, fully listed collection of 20 original game programs for the TI-99/4A computer, 9 of which are in standard TI BASIC with the remainder in Extended BASIC. Each line of code clearly explained. About half are arcade-type games, with the remainder assorted. By Khoa Ton and Quyen Ton.

BOOK ONLY: 176 pages, 5½ x 8½, soft. ISBN 0-672-22204-3. © 1983.

Ask for No. 22204 ..... **$8.95**

BOOK PLUS TAPE CASSETTE: Packed in 6 x 9 hardcover vinyl binder with cassette storage feature. ISBN 0-672-26169-3.

Ask for No. 26169 ..... **$15.95**

USER'S GUIDE TO MICROCOMPUTER BUZZWORDS

Handy quick-reference that provides an understanding of the basic terminology you need to become "computer literate." Contains many illustrations. By David H. Dasenbrock. 110 pages, 5½ x 8½, soft. ISBN 0-672-22049-0. © 1983.

Ask for No. 22049 ..... **$9.95**

USING COMPUTER INFORMATION SERVICES

Shows you how to use your microcomputer to communicate with everything from local bulletin boards to the national computer networks. Clearly explains what to expect on-screen, how to retrieve it, and more. By Larry Sturtz and Jeff Williams. 240 pages, 5½ x 8½, soft. ISBN 0-672-21997-2. © 1983.

Ask for No. 21997 ..... **$12.95**

MEGABUCKS FROM YOUR MICROCOMPUTER

Shows you how to make money using your creative talents through your microcomputer, and includes details for doing your own writing, reviewing, and programming. By Tim Knight. 80 pages, 8½ x 11, soft. ISBN 0-672-22083-0. © 1983.

Ask for No. 22083 ..... **$3.95**

These and other Sams Books and Software products are available from better retailers worldwide, or directly from Sams. Call 800-428-SAMS or 317-298-5566 to order, or to get the name of a Sams retailer near you. Ask for your free Sams Books and Software Catalog!

Prices good in USA only. Prices and page counts subject to change without notice.

SAMS

Entertainment Games in TI BASIC and Extended BASIC

This book provides the TI home computer user with a way to get and play arcade-type games at a very cheap price. With just the addition of joysticks and the TI Extended BASIC module, you have:

- Two programming languages—TI BASIC (built-in your machine) and TI Extended BASIC (a powerful, high-level language)—that you can learn while keying in the games.
- Twenty programs that let you enjoy the features and imagination of an adventure game or the skill and educational benefits of a school or business program.
- A way to increase your physical and mental dexterity while typing in and playing the many different games.
- A method of increasing your knowledge with word and spelling games while relaxing with pinball and other family-type games.
- A selection of family games, arcade-type games, educational games, plus business, graphics, and personal-use programs.

Each game is self-explanatory as you run it; each program is fully explained with "Features" sections, "How To Play" sections, and other details that allow you to get the full benefit of each program. By using this book, you can stay out of the arcades and keep your quarters at home in the piggy bank.

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

$8.95/22204

ISBN: 0-672-22204-3