```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!     B A T S                                                        !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Author: Bryce Martens
!
! Platform: Internet Basic running under Comet runtime system
!
! OPEN SOURCE
!
! --------------------------------------------------------------------
! Theory of operation:
!
! This game is based on a 20 x 20 grid. Each cell in the grid contains
! a numeric value that represents the contents of the cell. If the
! cell contains nothing, the grid value is 0. If the cell contains a
! fence, the grid value is 1. If the cell contains a vampire bat, the
! grid contains a 2. If the cell contains the player ("YOU"), the grid
! contains a 3.
!
! The game begins by assigning fences to their proper locations. All
! of the border cells contain a 1, and the middle of the grid is laced
! with a random number of fences (a value between 35 and 50) in random
! locations.
!
! The 12 bats are placed in the grid in random locations, and the
! the player ("YOU") is also placed in the grid at a random location.
!
! The game proceeds as the player moves to a new location in the grid.
! If the new location contains a bat or a fence, the player loses
! the game. However, if the new location is empty, the move is allowed
! and the game proceeds with the bats giving chase. The bats move
! toward the player in an attempt to electrocute him/her. If the bat
! lands on a cell where the player has moved, the player loses. But,
! if the bats hits a fence (or another bat), the bat is killed.
!
```

```
! The player can win the game by dodging all of the bats and (with
! some practice) driving all 12 of them to hit a fence or another bat.
!
! Programming notes:
!
! 1. Use of COMMON vs. LOCAL
!
!    Most of the variables in this program are LOCAL, but several are
!    COMMON. The CLEAR LOCAL instruction initializes (clears) the
!    LOCAL variables, but does not affect the COMMON ones, thus
!    preserving the COMMON values from game to game. The COMMON
!    variables include the number of wins and losses, the vampire
!    bat symbol (^@^), and the path names for the audio and wallpaper files
!
! 2. Use of array variables
!
!    The playing grid is a two-dimensional array named GRID. The
!    following chart describes how this array is used:
!
!    GRID(i,j) = 0  means nothing is located in the cell
!    GRID(i,j) = 1  means a fence is located in the cell
!    GRID(i,j) = 2  means a bat is located in the cell
!    GRID(i,j) = 3  means the player is located in the cell
!
!    Two additional array varaibles are used in this program. LINE
!    records the line locations of the bats, player, and fences;
!    COLUMN records the column locations of the bats, player, and
!    fences. The following chart describes how these arrays are used:
!
!    LINE(1) through LINE(12)  store the line locations for the bats
!    COLUMN(1) through COLUMN(12)  store the column locations for bats
!
!    LINE(13)  stores the line location for the player
!    COLUMN(13)  stores the column location for the player
!
!    LINE(14) through LINE(63)  stores the line locations for fences
```

```
!    COLUMN(14) through COLUMN(63)  stores column location for fences
!
!    These arrays are manipulated during the course of the game.
!
! 3. Screen formatting vs. grid coordinates
!
!    Each object on the screen (fence, bat, player) takes up 3 screen
!    positions. Therefore, a formula is required to determine the
!    correct screen position (as opposed to the grid coordinate).
!
!    The following general formula is used to determine the column
!    location:  (column - 1) * 3 + 1
!
!    No formula is required for the line location, as each object is
!    only one line high.
!
! 4. Use of "random" number routine
!
!    This game uses a pseudo-random number generator that is based
!    on the RND function.
!
! 5. Use of Easy Scan codes
!
!    The (EasyScan) mnemonic returns information about mouse clicks.
!    If the mouse has been clicked, the STS function returns the row and
!    column location of the click. This feature is used for the player's
!    selection (the matrix of values from 1 through 9), as well as the
!    other options in the program (help, foreit, boss, and quit). A mouse
!    click can also be used to close a window (help window and end-of-game
!    window). Note: It's easier to play this game using the keyboard
!    rather than the mouse, but the option is availble for those who want
!    to use it.
!
! 6. Use of extended Comet windows
!
!    The help message and ending messages make use of the
```

```
!    extneded Comet windows feature.
!
! 7. Use of single key transmit
!
!    The data entry in this program uses the single key transmit
!    feature. This means that the player does not have to press the
!    Enter (or F10) key after each move; simply pressing a key
!    triggers the input.
!
! 8. The hidden option
!
!    Some people have a tough time winning this game. For those
!    folks there's the hidden "J" option. This option allows the
!    player to jump to a new (random) location at any time during
!    the game.


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!     DATA DIVISION                                                          !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!    COMMON VARIABLES

     LENGTH 3   & COMMON VAMPIRE$       ! VAMPIRE CHARACTER ^@^
     LENGTH 3.0 & COMMON WINS, LOSSES   ! NUMBER OF WINS AND LOSSES
     LENGTH 6.6 & COMMON A,B,C          ! RANDOM NUMBER SEEDS

      LENGTH 60 & COMMON PATH$,AUDIOFILE$,WALLPAPER$


!    LOCAL VARIABLES

     LENGTH 1  & LOCAL MOVE$            ! MOVE DIRECTION/OPTION

     LENGTH 4   & LOCAL FLAG$           ! GENERAL-PURPOSE FLAG
     LENGTH 4   & LOCAL HANDLE$ ! Window handle
```

```
     LENGTH 1.0 & LOCAL GRID(20,20)     ! THE MASTER GRID
                  LOCAL MOVE, CHOICE    ! MOVE AND CHOICE
                  LOCAL ERR             ! ERROR ARGUMENT FOR NUM FCN.
     LENGTH 2.0
                  LOCAL LN,LNN,COL,COLL ! SCREEN AND GRID COORDINATES
                  LOCAL ALIVE, DEAD     ! COUNTERS
                  LOCAL LOC, FENCE      ! COUNTERS
                  LOCAL I               ! LOOP INDEX
                  LOCAL LINE(63)        ! MATRIX FOR LINE VALUES
                  LOCAL COLUMN(63)      ! MATRIX FOR COLUMN VALUES
     LENGTH 3.0 & LOCAL TOTAL           ! TOTAL NUMBER OF GAMES
                  LOCAL GAMENUMBER      ! COUNTER
                  LOCAL LOOP            ! COUNTER
                  LOCAL RAND            ! COUNTER
     LENGTH 4.3 & LOCAL PCTG            ! WIN/LOSS PERCENTAGE


      LENGTH 3.0 & LOCAL MouseRow,MouseColumn
      LENGTH 1.0 & LOCAL GameNumberMOD2, GameNumberMOD4
      LENGTH 2.0 & LOCAL GameNumberMOD

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!     FORMAT SECTION                                                        !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SKT:     FORMAT (SINGLE KEY TRANSMIT)

OPTIONS: FORMAT _
                "Use mouse/10-key",@(63,2);_
                "     U P     ",@(63,4);_
                "L           R",@(63,5);_
                "E           I",@(63,6);_
                "F           G",@(63,7);_
                "T           H",@(63,8);_
                "            T",@(63,9);_
                "   D O W N   ",@(63,10);_
```

```
                "Other options:",@(63,12);_
                "H = Help",@(63,13);_
                "B = Boss walks in",@(63,14);_
                "F = Forfeit game",@(63,15);_
                "Q = Quit",@(63,16);_
                "Your choice:",@(63,18)

ClickOptions:     FORMAT _
            (Change color="@1F@");_
            "7",@(67,6);_
            "8",@(69,6);_
            "9",@(71,6);_
            "4",@(67,7);_
            "5",@(69,7);_
            "6",@(71,7);_
            "1",@(67,8);_
            "2",@(69,8);_
            "3",@(71,8);_
            "H",@(63,13);_
            "B",@(63,14);_
            "F",@(63,15);_
            "Q",@(63,16)

STATS: FORMAT _
            "Game number: " ,@(1,22);GAMENUMBER;_
            "No. of bats:  ",@(1,23);ALIVE;_
            "No. of fences:",@(1,24);FENCE;_
            "Wins:       ",@(25,22);WINS;_
            "Losses:     ",@(25,23);LOSSES;_
            "Percentage: ",@(25,24);PCTG

BORDER: FORMAT (SB);"FEN           CAUTION: HIGH VOLTAGE ",@(1,LNN);_
                    "FENCE                FEN",@(37,LNN)

ClearTheField1: FORMAT _
            " ",54,@(4,2);_
```

```
            " ",54,@(4,3);_
            " ",54,@(4,4);_
            " ",54,@(4,5);_
            " ",54,@(4,6);_
            " ",54,@(4,7);_
            " ",54,@(4,8);_
            " ",54,@(4,9);_
            " ",54,@(4,10);_
            "@00@"

ClearTheField2: FORMAT _
            " ",54,@(4,11);_
            " ",54,@(4,12);_
            " ",54,@(4,13);_
            " ",54,@(4,14);_
            " ",54,@(4,15);_
            " ",54,@(4,16);_
            " ",54,@(4,17);_
            " ",54,@(4,18);_
            " ",54,@(4,19);_
            "@00@"

FENCE:   FORMAT (SB);"FEN",@(COLL,LNN)                                  ! FENCE

VAMPIRE: FORMAT (CHANGE COLOR="@F4@");VAMPIRE$,@(COLL,LNN)              ! BAT

YOU:     FORMAT (CHANGE COLOR="@F1@");"YOU",@(COLL,LNN)                 ! YOU

BLANKS:  FORMAT "   ",@(COLL,LNN);"@00@"                                ! BLANKS

HELP: FORMAT _
            "You are in a field with 12 deadly vampire bats",@(1,1);_
            "and lots of high voltage fences. Each time you",@(1,2);_
            "move, the bats move closer to you in order to",@(1,3);_
            "bite your neck. However, the bats die when they",@(1,4);_
            "fly into one of the electric fences. The fences",@(1,5);_
```

```
            "will also kill you if you happen to bump into",@(1,6);_
            "one by accident, so please be careful.",@(1,7);_
            "Move by using the mouse or the 10-key pad.",@(1,9);_
            "Up is 8,down is 2, left is 4, right is 6, etc.",@(1,11);_
            "You can forfeit a game by using the F option.",@(1,12);_
            "If your boss walks in, press B (immediately).",@(1,13);_
            "Good luck and happy bat hunting...",@(1,14);_
            "Click the mouse or press any key to continue.",@(1,17)



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!     CODE SECTION                                                            !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

TOP:
      ESCAPESUB GOTCHA                                          ! SET F3 TRAP
      CLEAR                                                     ! CLEAR ALL VARIABLES

      Print (AutoMetaFile=0)                                    ! Turn off GDI cache

      Print (Screen=81,28)                                      ! Set screen size
      PRINT (CS)                                                ! CLEAR THE SCREEN
      PRINT (Easy Scan)                                         ! Set Easy Scan mode

      Print (HideCursor)                                        ! Hide the cursor

      C = RND(1)                                                ! RANDOMIZE

      ! Determine the path for the WAN files and graphics files

      PATH$ = STRIP(PATH(SUB(FSTAT("BATS"),3,3))) + "MEDIA\"

      VAMPIRE$ = "^" + CHR(64) + "^"                            ! FLYING BAT CHARACTER  ^@^

      ! Display the outer border of fences
```

```
      LNN = 1                                                  ! SET LINE TO 1
      PRINT (0,BORDER)                                         ! PRINT TOP LINE OF FENCES

      FOR LN = 2 TO 19                                         ! LOOP FROM LINES 2 TO 19
        LNN = LN                                               ! SET LINE VALUE
        COLL = 1                                               ! SET COLUMN VALUE
        PRINT (0,FENCE)                                        ! DISPLAY A FENCE
        COLL = 58                                              ! SET COLUMN VALUE
        PRINT (0,FENCE)                                        ! DISPLAY A FENCE
      NEXT LN                                                  ! CONTINUE LINE LOOP

      LNN = 20                                                 ! SET LINE TO 20
      PRINT (0,BORDER)                                         ! PRINT BOTTOM LINE OF FENCES

      Print (Draw box=62,1,80,20)                              ! DRAW A BOX AROUND THE OPTIONS

      PRINT (0,OPTIONS)                                        ! PRINT MOVE SCREEN
      Print (0,ClickOptions)                                   ! Options matrix


! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-

START:      CLEAR LOCAL                                        ! CLEAR THE LOCAL FIELDS

      GOSUB STATS                                              ! COMPUTE GAME STATISTICS
      GAMENUMBER = TOTAL + 1                                   ! COMPUTE GAME NUMBER

      Print (PlaySound="",64)                                  ! 64 = purge any active sounds

      ! THE FOLLOWING LINES SET UP THE FENCES AROUND THE BORDER AREA
      ! NOTE: THE CODE VALUE FOR FENCES IS 1

      FOR LN = 1 TO 20
        FOR COL = 1 TO 20
```

```
    IF LN=1 OR LN=20 OR COL=1 OR COL=20 THEN GRID(LN,COL)=1
  NEXT COL
NEXT LN

! THE FOLLOWING LINES SET UP THE RANDOM LOCATIONS FOR THE BATS
! NOTE:  THE CODE NUMBER FOR BATS IS 2

FOR LOC = 1 TO 12

OOPS1:
  LN = 20 * RND(0) + 1                          ! CREATE RANDOM LINE NUMBER
  COL = 20 * RND(0)+ 1                          ! CREATE A RANDOM COLUMN NUMBER
    IF GRID(LN,COL) NE 0 THEN GOTO OOPS1        ! IS SOMETHING THERE?
  GRID(LN,COL) = 2                              ! PUT A BAT THERE (CODE = 2)
  LINE(LOC) = LN                                ! SAVE THE LINE LOCATION
  COLUMN(LOC) = COL                             ! SAVE THE COLUMN LOCATION

NEXT LOC

! THE FOLLOWING LINES PUT THE PLAYER IN A RANDOM LOCATION
! NOTE:  THE CODE FOR THE PLAYER IS 3

OOPS2:
  LN = 20 * RND(0) + 1                          ! CREATE A RANDOM LINE NUMBER
  COL = 20 * RND(0) + 1                         ! CREATE A RANDOM COLUMN NUMBER
    IF GRID(LN,COL) NE 0 THEN GOTO OOPS2        ! IS SOMETHING THERE?
  GRID(LN,COL) = 3                              ! PUT THE PLAYER THERE (3)
  LINE(13) = LN                                 ! RECORD THE LINE LOCATION
  COLUMN(13) = COL                              ! RECORD THE COLUMN LOCATION

! THE FOLLOWING LINES PLACE THE FENCES IN THE GRID
! THE NUMBER OF FENCES IS BETWEEN 35 AND 50 (RANDOM)
! NOTE:  THE CODE FOR A FENCE IS 1

FENCE = 35 * RND(0) + 15                        ! FIGURE OUT HOW MANY FENCES
```

```
      FOR I = 1 TO FENCE

      OOPS3:
        LN = 20 * RND(0) + 1                            ! CREATE A RANDOM LINE NUMBER
        COL = 20 * RND(0) + 1                           ! CREATE A RANDOM COLUMN NUMBER
          IF GRID(LN,COL) NE 0 THEN GOTO OOPS3          ! IS SOMETHING THERE?
        GRID(LN,COL) = 1                                ! SET THE FENCE CODE = 1
        LINE(I+13) = LN                                 ! RECORD THE LINE LOCATION
        COLUMN(I+13) = COL                              ! RECORD THE COLUMN LOCATION

      NEXT I

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-

GO:
      ! THE FOLLOWING LINES DISPLAY THE INNER PLAYING SCREEN

      Print (0,ClearTheField1)                          ! First, clear the inner lines
      Print (0,ClearTheField2)                          ! (in two passes, because of the limited buffer size)

      FOR LN = 2 TO 19                                  ! LOOP FROM LINES 2 TO 19
      LNN = LN                                          ! SET LINE VALUE
            FOR COL = 2 TO 19                           ! LOOP FROM COLUMN 2 TO 19
            COLL = ((COL-1)*3)+1                        ! SET COLUMN VALUE

            ON GRID(LN,COL) GOTO SKIP,SKIP,FENCE,BAT,YOU

      FENCE:     PRINT (0,FENCE)   & GOTO SKIP          ! DISPLAY A FENCE
      BAT:       PRINT (0,VAMPIRE) & GOTO SKIP          ! DISPLAY A BAT
      YOU:       PRINT (0,YOU)                          ! DISPLAY "YOU"

      SKIP:  NEXT COL                                   ! CONTINUE COLUMN LOOP
      NEXT LN                                           ! CONTINUE LINE LOOP
```

```
! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-

MAKEMOVE:

      ALIVE = 12 - DEAD                                  ! NUMBER OF BATS STILL ALIVE
      LN = LINE(13)                                      ! GET PLAYER'S LOCATION
      COL = COLUMN(13)                                   ! GET PLAYER'S LOCATION

      PRINT (0,STATS)                                    ! PRINT STATISTICS SCREEN

      Print " ",@(76,18);"@00@"                          ! Clear user's previous move option

      ! Determine which sound file to play, based on the game number

      GameNumberMOD = GAMENUMBER MOD 28

            SELECT CASE GameNumberMOD
            CASE 1
            AUDIOFILE$ = PATH$ + "GAME1.WAV"
            CASE 2
            AUDIOFILE$ = PATH$ + "GAME2.WAV"
            CASE 3
            AUDIOFILE$ = PATH$ + "GAME3.WAV"
            CASE 4
            AUDIOFILE$ = PATH$ + "GAME4.WAV"
            CASE 5
            AUDIOFILE$ = PATH$ + "GAME5.WAV"
            CASE 6
            AUDIOFILE$ = PATH$ + "GAME6.WAV"
            CASE 7
            AUDIOFILE$ = PATH$ + "GAME7.WAV"
            CASE 8
            AUDIOFILE$ = PATH$ + "GAME8.WAV"
            CASE 9
            AUDIOFILE$ = PATH$ + "GAME9.WAV"
```

```
CASE 10
AUDIOFILE$ = PATH$ + "GAME10.WAV"
CASE 11
AUDIOFILE$ = PATH$ + "GAME11.WAV"
CASE 12
AUDIOFILE$ = PATH$ + "GAME12.WAV"
CASE 13
AUDIOFILE$ = PATH$ + "GAME13.WAV"
CASE 14
AUDIOFILE$ = PATH$ + "GAME14.WAV"
CASE 15
AUDIOFILE$ = PATH$ + "GAME15.WAV"
CASE 16
AUDIOFILE$ = PATH$ + "GAME16.WAV"
CASE 17
AUDIOFILE$ = PATH$ + "GAME17.WAV"
CASE 18
AUDIOFILE$ = PATH$ + "GAME18.WAV"
CASE 19
AUDIOFILE$ = PATH$ + "GAME19.WAV"
CASE 20
AUDIOFILE$ = PATH$ + "GAME20.WAV"
CASE 21
AUDIOFILE$ = PATH$ + "GAME21.WAV"
CASE 22
AUDIOFILE$ = PATH$ + "GAME22.WAV"
CASE 23
AUDIOFILE$ = PATH$ + "GAME23.WAV"
CASE 24
AUDIOFILE$ = PATH$ + "GAME24.WAV"
CASE 25
AUDIOFILE$ = PATH$ + "GAME25.WAV"
CASE 26
AUDIOFILE$ = PATH$ + "GAME26.WAV"
CASE 27
AUDIOFILE$ = PATH$ + "GAME27.WAV"
```

```
      CASE 0
      AUDIOFILE$ = PATH$ + "GAME28.WAV"
      ENDSELECT

Print (PlaySound=AUDIOFILE$,19)                 ! Play sound
                                                ! 19 = 1 + 2 + 16
                                                !  1 = play asynchronously
                                                !  2 = silence if not found
                                                ! 16 = don't stop if other sound is playing
                                                !      (which keeps file playing during game)

PRINT (0,SKT)                                   ! SET "SINGLE KEY TRANSMIT"
INPUT @(76,18),MOVE$                            ! ENTER THE MOVE OR OPTION
MOVE$ = UCASE(MOVE$)                            ! CONVERT TO UPPER CASE

! Check to see if the mouse was clicked. If so, determine where and set the option

If ASC(SUB(STS(0),8,1)) = 6 THEN

      MouseRow = Asc(Sub(Sts(0),15, 1))              ! Get the row
      MouseColumn = Asc(Sub(Sts(0), 16, 1))          ! Get the column

      If MouseRow=8 AND MouseColumn=67 THEN MOVE$="1" & GOTO SelectMove
      If MouseRow=8 AND MouseColumn=69 THEN MOVE$="2" & GOTO SelectMove
      If MouseRow=8 AND MouseColumn=71 THEN MOVE$="3" & GOTO SelectMove
      If MouseRow=7 AND MouseColumn=67 THEN MOVE$="4" & GOTO SelectMove
      If MouseRow=7 AND MouseColumn=69 THEN MOVE$="5" & GOTO SelectMove
      If MouseRow=7 AND MouseColumn=71 THEN MOVE$="6" & GOTO SelectMove
      If MouseRow=6 AND MouseColumn=67 THEN MOVE$="7" & GOTO SelectMove
      If MouseRow=6 AND MouseColumn=69 THEN MOVE$="8" & GOTO SelectMove
      If MouseRow=6 AND MouseColumn=71 THEN MOVE$="9" & GOTO SelectMove
      If MouseRow=13 AND MouseColumn=63 THEN MOVE$ = "H" & GOTO SelectMove
      If MouseRow=14 AND MouseColumn=63 THEN MOVE$ = "B" & GOTO SelectMove
      If MouseRow=15 AND MouseColumn=63 THEN MOVE$ = "F" & GOTO SelectMove
      If MouseRow=16 AND MouseColumn=63 THEN MOVE$ = "Q" & GOTO SelectMove
```

```
            GOTO MAKEMOVE

      Endif

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-

! THE FOLLOWING STRUCTURE EVALUATES THE PLAYER'S MOVE OR OTHER CHOICE

SelectMove:

       SELECT CASE MOVE$                                  ! START OF CASE STRUCTURE

       CASE ""                                            ! IF MOVE$ = NULL
          GOTO MAKEMOVE                                   ! GO BACK AND RE-INPUT

      CASE "H"                                            ! HELP MESSAGE

            GameNumberMOD2 = GameNumber MOD 2             ! Choose sound file

            IF GameNumberMOD2 = 1 THEN
              AUDIOFILE$ = PATH$ + "HELP1.WAV"
            ELSE
              AUDIOFILE$ = PATH$ + "HELP2.WAV"
            ENDIF

            Print (PlaySound=AUDIOFILE$,11)               ! Play sound file
                                                          ! 11 = 1 + 2 + 8
                                                          !  1 = play asynch
                                                          !  2 = silence if no sound file
                                                          !  8 = loop until the next sound is played


            Print (Create Window EX=5,3,50,21,64,1,"How to play the game of BATS")
            Input Handle$
```

```
            WALLPAPER$ = PATH$ + "HELP.JPG"                 ! Put wallpaper in window
            Print (WallpaperBitmap=WALLPAPER$)

            PRINT (0,HELP)                                  ! Display help message


            Print (0,SKT)                                   ! Single key transmit
            Input @(47,17),MOVE$                            ! Dummy input (to hold the message on the
screen)

            Print (Delete Window EX=Handle$)                ! Delete the window

            GOTO MAKEMOVE                                   ! GO BACK

      CASE "J"                                              ! "J" IS A HIDDEN OPTION
            TOTAL = WINS + LOSSES                           ! CALCULATE TOTAL GAMES
            GOTO JUMP                                       ! GO

      CASE "B"                                              ! BOSS WALKS IN
            PRINT (LAUNCH="explorer.exe")                   ! LAUNCH EXPLORER

            AUDIOFILE$ = PATH$ + "SILENCE.WAV"              ! Play silence
            Print (PlaySound=AUDIOFILE$,3)

            GOTO MAKEMOVE                                   ! JUMP BACK TO DISPLAY

      CASE "F"                                              ! FORFEIT A GAME

      AUDIOFILE$ = PATH$ + "RICOCHET.WAV"                   ! Play sound file
            Print (PlaySound=AUDIOFILE$,3)                  ! 3 = 1 (asynch) + 2 (silence if not found)

            PAUSE(18)                                       ! Wait for 1 second (to play sound)

            LOSSES = LOSSES + 1                             ! INCREMENT NUMBER OF LOSSES

            GOTO START                                      ! BRANCH TO START OF GAME
```

```
CASE "Q"                                    ! QUIT THE GAME
      GOTO GameOver

CASE "1"                                    ! PLAYER SELECTED "1"
    LN = LN + 1                             ! MOVE DOWN
    COL = COL - 1                           ! MOVE LEFT

CASE "2"                                    ! PLAYER SELECTED "2"
    LN = LN + 1                             ! MOVE DOWN

CASE "3"                                    ! PLAYER SELECTED "3"
    LN = LN + 1                             ! MOVE DOWN
    COL = COL + 1                           ! MOVE RIGHT

CASE "4"                                    ! PLAYER SELECTED "4"
    COL = COL - 1                           ! MOVE LEFT

CASE "5"                                    ! PLAYER SELECTED "5"
                                            ! DON'T MOVE

CASE "6"                                    ! PLAYER SELECTED "6"
    COL = COL + 1                           ! MOVE RIGHT

CASE "7"                                    ! PLAYER SELECTED "7"
    LN = LN - 1                             ! MOVE UP
    COL = COL - 1                           ! MOVE LEFT

 CASE "8"                                   ! PLAYER SELECTED "8"
    LN = LN - 1                             ! MOVE UP

 CASE "9"                                   ! PLAYER SELECTED "9"
    LN = LN - 1                             ! MOVE UP
    COL = COL + 1                           ! MOVE RIGHT

 CASE ELSE                                  ! ANY OTHER VALUE?
```

```
          GOTO MAKEMOVE                                    ! JUMP BACK TO MAKEMOVE

        ENDSELECT                                          ! END OF CASE STRUCTURE

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-

!    THE FOLLOWING LINES CLEAR THE PLAYER'S LOCATION AND MOVE THE
!    PLAYER TO A NEW LOCATION

     GRID(LINE(13),COLUMN(13)) = 0      ! ZERO OUT OLD LOCATION
     LNN = LINE(13)                     ! GET SCREEN COORDINATES
     COLL = ((COLUMN(13)-1)*3)+1        !   FOR ROW AND COLUMN
     PRINT (0,BLANKS)                   ! BLANK OUT OLD LOCATION

     IF GRID(LN,COL) = 0 THEN           ! IF NEW LOCATION IS EMPTY
        GRID(LN,COL) = 3                ! SAVE NEW PLAYER LOCATION
        LINE(13) = LN                   ! SAVE NEW COORDINATES
        COLUMN(13) = COL                !   FOR ROW AND COLUMN
        LNN = LINE(13)                  ! SET NEW SCREEN COORIDATES
        COLL = ((COLUMN(13)-1)*3)+1     !   FOR ROW AND COLUMN
        PRINT (0,YOU)                   ! DISPLAY NEW PLAYER LOCATION
        GOTO BATLOOP
     ENDIF

     IF GRID(LN,COL) = 1 OR _           ! HIT A FENCE, OR
        GRID(LN,COL) = 2 THEN _         ! HIT A BAT
        GOTO LOSER                      ! JUMP TO LOSER ROUTINE

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

!  THE FOLLOWING LOOP MOVES ALL 12 BATS

BATLOOP:                                ! LOOP FOR ALL 12 BATS
     FOR LOC = 1 TO 12
```

```
     IF GRID(LINE(LOC),COLUMN(LOC)) NE 2 THEN GOTO BOTTOM

     GRID(LINE(LOC),COLUMN(LOC)) = 0    ! ZERO OUT PREVIOUS LOCATION
     LNN = LINE(LOC)                    ! DETERMINE SCREEN COORDINATES
     COLL = ((COLUMN(LOC)-1)*3)+1       !   FOR ROW AND COLUMN
     PRINT (0,BLANKS)                   ! BLANK OUT OLD LOCATION

MOVEBAT:
     IF LINE(LOC) >= LN THEN GOTO UP
     LINE(LOC) = LINE(LOC) + 1                  ! MOVE DOWN

UP:
     IF LINE(LOC) <= LN THEN GOTO RIGHT
     LINE(LOC) = LINE(LOC) - 1                  ! MOVE UP

RIGHT:
     IF COLUMN(LOC) >= COL THEN GOTO LEFT
     COLUMN(LOC) = COLUMN(LOC) + 1              ! MOVE RIGHT
     IF COLUMN(LOC) <= COL THEN GOTO CHECK
LEFT:
     IF COLUMN(LOC) <= COL THEN GOTO CHECK
     COLUMN(LOC) = COLUMN(LOC) - 1              ! MOVE LEFT

CHECK:
     IF GRID(LINE(LOC),COLUMN(LOC)) NE 1 AND _
        GRID(LINE(LOC),COLUMN(LOC)) NE 2 THEN GOTO SHOWBAT

     DEAD = DEAD + 1                    ! INCREMENT THE DEAD COUNTER
     IF DEAD = 12 THEN GOTO WINNER      ! ALL BATS ARE DEAD, WIN GAME

     GOTO BOTTOM                        ! BRANCH TO BOTTOM OF LOOP

SHOWBAT:
     LNN = LINE(LOC)                    ! DETERMINE SCREEN COORDINATES
     COLL = ((COLUMN(LOC)-1)*3)+1       !   FOR ROW AND COLUMN
     PRINT (0,VAMPIRE)                  ! DISPLAY BAT AT NEW LOCATION
```

```
     IF GRID(LINE(LOC),COLUMN(LOC)) = 3 THEN   ! THE BAT HIT "YOU"
        FLAG$ = "OUCH"                         ! SET FLAG
        GOTO LOSER                             ! GO TO LOSER ROUTINE
     ENDIF

     GRID(LINE(LOC),COLUMN(LOC)) = 2    ! SAVE THE BAT'S NEW LOCATION

BOTTOM:
     NEXT LOC                           ! BOTTOM OF LOOP
     GOTO MAKEMOVE                      ! BRANCH TO MAVEMOVE ROUTINE

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

!  THE FOLLOWING ROUTINE IS EXECUTED WHEN THE PLAYER WINS A GAME

WINNER:
        WINS = WINS + 1                 ! INCREMENT THE WIN COUNTER
        ALIVE = 0                       ! ZERO OUT THE ALIVE COUNTER
        GOSUB STATS                     ! DETERMINE GAME STATISTICS
        PRINT (0,STATS)                 ! DISPLAY STATISTICS

      ! Determine which sound file to play, based on the game number

      GameNumberMOD4 = GAMENUMBER MOD 4

      IF GameNumberMOD4 = 1 THEN AUDIOFILE$ = PATH$ + "win1.wav"
      IF GameNumberMOD4 = 2 THEN AUDIOFILE$ = PATH$ + "win2.wav"
      IF GameNumberMOD4 = 3 THEN AUDIOFILE$ = PATH$ + "win3.wav"
      IF GameNumberMOD4 = 0 THEN AUDIOFILE$ = PATH$ + "win4.wav"

      Print (PlaySound=AUDIOFILE$,3)

      Print (CREATE WINDOW EX=45,19,30,8,64,1,"Game Over")
      Input Handle$
```

```
      WALLPAPER$ = PATH$ + "GRAY.JPG"
      Print (WallpaperBitmap=WALLPAPER$)
      Print "I don't believe it.",@(1,1);_
                  "You actually won a game...",@(1,2);_
                  "Click the mouse or press",@(1,4);_
                  "any key to continue.",@(1,5)

      GOTO ContinueTheGame

!  THE FOLLOWING ROUTINE IS EXECUTED WHEN THE PLAYER LOSES A GAME
!  62800

LOSER:

      LOSSES = LOSSES + 1             ! INCREMENT THE LOSS COUNTER
      GOSUB STATS                     ! DETERMINE GAME STATISTICS
      PRINT (0,STATS)                 ! DISPLAY STATISTICS


!     The player hit a fence

      IF GRID(LN,COL) = 1 THEN

            AUDIOFILE$ = PATH$ + "ZAP.WAV"
            Print (PlaySound=AUDIOFILE$,3)

            Print (CREATE WINDOW EX=45,19,30,8,64,1,"Game Over")
            Input Handle$

            WALLPAPER$ = PATH$ + "RED.JPG"
            Print (WallpaperBitmap=WALLPAPER$)
            Print "You hit the fence and had a ",@(1,1);_
                        "very SHOCKING experience.",@(1,2);_
                        "Click the mouse or press",@(1,4);_
                        "any key to continue.",@(1,5)
```

```
      ENDIF

!     The player flew into a bat

      IF GRID(LN,COL) = 2 THEN

            AUDIOFILE$ = PATH$ + "DUCK.WAV"
            Print (PlaySound=AUDIOFILE$,3)

            Print (CREATE WINDOW EX=45,19,30,8,64,1,"Game Over")
            Input Handle$

            WALLPAPER$ = PATH$ + "RED.JPG"
            Print (WallpaperBitmap=WALLPAPER$)
            Print "Hey dummy!",@(1,1);_
                        "You flew right into a bat!",@(1,2);_
                        "Click the mouse or press",@(1,4);_
                        "any key to continue.",@(1,5)

      ENDIF

! A bat flew into the player

      IF FLAG$ = "OUCH" THEN

            AUDIOFILE$ = PATH$ + "HITBAT.WAV"
            Print (PlaySound=AUDIOFILE$,3)

            Print (CREATE WINDOW EX=45,19,30,8,64,1,"Game Over")
            Input Handle$

            WALLPAPER$ = PATH$ + "RED.JPG"
            Print (WallpaperBitmap=WALLPAPER$)
            Print "Hold on to your neck.",@(1,1);_
                        "A bat just bit you!",@(1,2);_
                        "Click the mouse or press",@(1,4);_
```

```
                        "any key to continue.",@(1,5)

      ENDIF

ContinueTheGame:

      PRINT (0,SKT)                              ! SINGLE KEY TRANSMIT
      INPUT @ (32,6),MOVE$
      MOVE$ = UCASE(MOVE$)

      PRINT (DELETE WINDOW EX=Handle$)           ! DELETE THE WINDOW
      PRINT (WC);(SF);(BF)                       ! CLEAR THE MESSAGE LINE
      GOTO START                                 ! GO TO BEGINNING OF GAME


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!    S U B R O U T I N E S                                                     !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

STATS:                                       ! COMPUTE GAME STATISTICS
       TOTAL = WINS + LOSSES                 ! COMPUTE NEW TOTAL
       PCTG = (WINS * 1.000)/TOTAL           ! COMPUTE NEW PERCENTAGE
       RETURN

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

GOTCHA: ESCAPESUB GOTCHA                     ! RESET F3 TRAP
        RETURN                               ! RETURN TO PROGRAM

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

JUMP:                                      ! THIS IS A HIDDEN OPTION
                                           ! TO JUMP TO A NEW LOCATION
                                           ! IF YOU'RE CORNERED

     GRID(LINE(13),COLUMN(13)) = 0         ! CLEAR OLD PLAYER LOCATION
```

```
     LNN = LINE(13)                      ! GET SCREEN COORDINATE
     COLL = ((COLUMN(13)-1)*3)+1         ! GET SCREEN COORDINATE
     PRINT (0,BLANKS)                    ! BLANK OUT OLD LOCATION

!    THE FOLLOWING LINES PUT THE PLAYER IN A RANDOM LOCATION
!    NOTE:  THE CODE FOR THE PLAYER IS 3

OOPS9:
     LN = 20 * RND(0) + 1                     ! CREATE A RANDOM LINE NUMBER
     COL = 20 * RND(0) + 1                    ! CREATE A RANDOM COLUMN NUM.

     IF GRID(LN,COL) NE 0 THEN GOTO OOPS9    ! IS SOMETHING THERE?

     GRID(LN,COL) = 3                    ! PUT THE PLAYER THERE (3)
     LINE(13) = LN                       ! RECORD THE LINE LOCATION
     COLUMN(13) = COL                    ! RECORD THE COLUMN LOCATION
     LNN = LINE(13)                      ! GET SCREEN COORDINATE
     COLL = ((COLUMN(13)-1)*3)+1         ! GET SCREEN COORDINATE
     PRINT (0,YOU)                       ! PRINT "YOU" IN NEW PLACE

     GOTO MAKEMOVE                       ! BACK TO THE PROGRAM

! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-

GameOver:

      AUDIOFILE$ = PATH$ + "GOODBYE.WAV"                   ! Select the sound file
        Print (PlaySound=AUDIOFILE$,3)                     ! Play it

      PRINT (CS)                                           ! CLEAR THE SCREEN
      PRINT (SCAN CODES OFF)                               ! TURN SCAN CODES OFF
      PRINT (WC);(SF);(BF)                                 ! CLEAR THE MESSAGE LINE
      Print (ShowCursor)                                   ! Show the cursor
```

```
      ! Decide whether to STOP or EXIT, based on the value of ENTERLEVEL
      ! (i.e., this program can be RUN or ENTERed).

      IF ENTERLEVEL = 0                                    ! CHECK ENTERLEVEL VALUE
            STOP                                           ! STOP
      ELSE
            EXIT                                           ! EXIT TO CALLING PROGRAM
      ENDIF

END
```