

HP-71

Users' Library Solutions

Games





USERS' LIBRARY ORDER FORM

HP-71 Solutions Books are available at a cost of \$12.50 each. To order, please call toll free (800) 538-8787, contact your local dealer, or use the form provided below to order directly from the Users' Library. If ordering from the Users' Library, please include a handling charge of \$3.50 per order.

REMEMBER...

You can further customize your HP-71 with software available through the Users' Library.

A Users' Library Membership includes 2 free Solutions Books (a \$25.00 value) plus periodic updates containing:

- Product Information
- New Software Availability
- Contests
- Third-Party Products
- Discounts

YES! I want to subscribe to the Users' Library.

☐ \$20.00 U.S. and Canada (5851-1011)
(plus state and local taxes)

☐ \$35.00 OUTSIDE the U.S. and Canada (5851-1012)

Solution Books Requested

1. _____
Title _____ Part Number _____
2. _____
Title _____ Part Number _____

HP-71 Solutions Book Selection includes:

00071-90064/Math 00071-90066/General Utilities
00071-90065/Games

Please contact the Users' Library for further information regarding recorded material.

Solutions Book Request

Solutions Book Part Number	Title	Price
00071-900 _ _		
00071-900 _ _		
00071-900 _ _		
Users' Library Subscription		
State and Local Taxes		
Solutions Book handling charge (\$3.50)		
TOTAL		

All payments must be in U.S. dollars and drawn on a U.S. bank.

Send all orders (with payment) directly to: HEWLETT-PACKARD USERS' LIBRARY
Dept. 39UL
1000 N.E. Circle Blvd.
Corvallis, Oregon 97330

- ☐ Payment enclosed ☐ Master Card ☐ VISA ☐ Purchase order
(\$20.00 minimum)

Card Number/Company Purchase Order

Name

Company (if applicable)

Address

City

State/Country

Zip

CONTENTS

<u>Code Crack</u>	1
User Instructions	2
Listing	4
<u>Craps</u>	6
User Instructions	7
Listing	10
<u>Hangman</u>	12
User Instructions	13
Listing	15
<u>Blackjack</u>	16
User Instructions	17
Listing	21
<u>Hamurabi</u>	25
User Instructions	26
Listing	28
<u>Space War</u>	30
User Instructions	32
Listing	36

Code Crack

The object of this game is to guess the random sequence of colors the HP-71 has selected. Initially, the player must specify the number of elements in the sequence (4 to 9), the number of colors used to generate the sequence (4 to 9), and whether a color may appear more than once in the sequence. Note that if doubling of colors is not selected, the number of colors must at least equal the number of elements. In this case, the program will not allow inputs for the number of colors that are less than the number of elements. Note also that if doubling is not allowed and the number of elements selected is 9, the program assumes 9 colors and will not prompt for the input.

Once the sequence is generated, the player must enter a guess. The program accepts the guess one character at a time with no correction of errors. If the value entered is not what was intended, it can not be changed and the remainder of the guess must be entered. Once the last color is entered, the program compares the input with the stored sequence and tells you how close you were. The two numerals displayed after the player's guess are the number of elements of the right color which are in the right position and the number of elements which are of the right color but in the wrong position. For example, if the stored sequence is "VIBG" and the guess is "VBIY" the display would show "'VBIY' [1,2]", indicating 1 element in the right place ("V") and 2 elements of the right color in the wrong places ("I" and "B").

The program allows the player to review past guesses. There is no limit to the number of guesses you can make, except to the limit of memory used to store guesses.

The game uses the following color abbreviations: Violet - "V", Indigo - "I", Blue - "B", Green - "G", Yellow - "Y", Orange - "O", Red - "R", White - "W", and Purple - "P". If the player chooses a number of colors that is less than 9, the valid colors are the first "n" colors from the above sequence, where n is the number of colors. For example, if 4 colors are chosen, they will be V, I, B, and G. No other color inputs will be allowed. Of course, if doubling were allowed, all of these first four colors might not appear.

User Instructions

Comments	Input	Display
=====	=====	=====
1) Run the program.		Code Crack
2) This display identifies the program.		Doubling? (Y,N)
3) The program is asking whether more than one occurrence of the same color is to be allowed. Press [Y] for "yes" and [N] for "no".	[Y] or [N]	# of positions? (4-9)
4) The program is asking for the number of elements in the sequence. Valid entries are the single digits 4 through 9. (4-9)	4 through 9	# of colors?
5) The program is asking for the number of colors from which to choose in generating the sequence. If the "no doubling" option has been selected, the range for this prompt will be from N to 9 where N is the number of elements from step 4. As a convenience, if "no doubling" has been selected and the number of elements is 9, this prompt will be skipped since the only valid input must be 9.	N through 9	guess[1]: '####'
6) The player is prompted to make a guess as to the sequence. The number in brackets is the guess number. The number of #'s represent the number of elements in the sequence and a color guess for each element must be entered. Possible colors include: "V" - violet, "I" - indigo, "B" - blue, "G" - green, "Y" - yellow, "O" - orange, "R" - red, "W" - white, and "P" - purple.		

"Q" may also be entered at this time to terminate the program. In all cases, the program continues after the required inputs are entered. Pressing [END LINE] is not required.

colors

'AAAA' [a,b]

If "Q" was chosen, goto step 9.

The sequence was

If the following display is seen, goto step 8.

Correct in 'n' tries.

- 7) This display shows how close your guess was to the correct solution. The sequence is displayed followed by two numerals in square brackets. "a" is the number of elements that are in the correct position. "b" is the number of correct color guesses that are not in the proper location.

At this time, the user may scroll through all previous guesses by using the [v], [^], [g][v] and [g][^] keys. The program will beep when you reach the extremes of the guess list.

To continue guessing, press [END LINE] and go to step 6 of these instructions.

- 8) You have guessed the sequence in 'n' attempts.
- 9) The correct sequence is displayed and you are asked if you would like to play again.
- 10) If you would like to play again, press [Y] and go to step 2. Otherwise, press [N].

The sequence was

Play again?

Game Over.

Listing

```
10 RANDOMIZE
20 INTEGER C,G,D,P
30 DIM D0$[2],D1$[4],W$[2]
40 D0$=PEEK$("2F6DC",2)
50 D1$=PEEK$("2F946",4)
60 W$=PEEK$("2F94F",2)
70 'START': STD @ DELAY 0,0 @ OPTION BASE 1 @ WIDTH INF
80 DISP "Code Crack" @ WAIT 1
90 D=POS("NY",FNK$("Doubling? (Y,N)","YN",1))-1
100 P=VAL(FNK$("# of positions? (4-9)","456789",1))
110 IF D THEN C=VAL(FNK$("# of colors? (4-9)","456789",1))
120 IF NOT D AND P#9 THEN C=VAL(FNK$("# of colors? ("&STR$(P)&"-9)","1
23456789"[P],1))
130 IF NOT D AND P=9 THEN C=9
140 DISP "working";
150 DIM A$[P],C$[9],G$[P],S$[P]
160 C$="VIBGYORPW"
170 FOR I=1 TO P
180 'AGAIN': A$[I]=C$[IP(C*RND)+1][1,1]
190 IF POS(A$[1,I-1],A$[I]) AND NOT D THEN 'AGAIN'
200 DISP ".";
210 NEXT I
220 DISP
230 G=0
240 'GUESS': G$="#####"[1,P]
250 G=G+1
260 FOR I=1 TO P
270 G$[I,I]=FNK$(CHR$(13)&"guess["&STR$(G)&"]": '"&G$&"',C$[1,C]&"Q",0
)
280 IF G$[I,I]="Q" THEN 'QUIT'
290 NEXT I
300 DISP CHR$(13)&"guess["&STR$(G)&"]": '"&G$&"'
310 IF G$=A$ THEN 'CORRECT'
320 O=0 @ C0=0
330 DIM O$(G)[P+8]
340 O$(G)="'"&G$&"' [" @ S$=A$
350 FOR I=1 TO P
360 IF G$[I,I]=S$[I,I] THEN O=O+1 @ G$[I,I]=" " @ S$[I,I]="*"
370 NEXT I
380 FOR I=1 TO P
390 IF POS(S$,G$[I,I]) THEN C0=C0+1 @ S$[POS(S$,G$[I,I]),POS(S$,G$[I,I
])]="*" @ G$[I,I]=" "
400 NEXT I
410 O$(G)=O$(G)&STR$(O)&","&STR$(C0)&"]"
420 I=G
430 'REVIEW': K0$=FNK1$(O$(I),"#38#50#51#162#163")
440 IF K0$="#38" THEN 'GUESS'
450 IF K0$="#50" THEN I=I-1
460 IF K0$="#51" THEN I=I+1
470 IF K0$="#162" THEN I=1
480 IF K0$="#163" THEN I=G
490 IF I<1 THEN I=1 @ BEEP 3000,.05
500 IF I>G THEN I=G @ BEEP 3000,.05
510 GOTO 'REVIEW'
```

```

520 'CORRECT': DISP "Correct in "&STR$(G)&" tries." @ WAIT 2
530 'QUIT': DISP @ DISP "The sequence was" @ WAIT 2 @ DISP "'&A$&'"
@ WAIT P/2
540 IF POS("YN",FNK$("Play again?","YN",1))=1 THEN 'START'
550 DISP "Game over." @ WAIT 1
560 POKE "2F6DC",D0$
570 POKE "2F946",D1$
580 POKE "2F94F",W$
590 PUT "#38"
600 !
610 DEF FNK$(D$,K$,E)
620 DISP D$; @ IF E THEN DISP
630 'LOOP': K1$=KEY$
640 IF NOT POS(K$,K1$) THEN 'LOOP'
650 FNK$=K1$
660 END DEF
670 !
680 DEF FNK1$(D$,K$)
690 DISP D$
700 'LOOP1': K1$=KEY$
710 IF LEN(K1$)<2 OR NOT POS(K$,K1$) THEN 'LOOP1'
720 FNK1$=K1$
730 END DEF

```

Craps

In this game of craps, the HP-71 is the 'house', using simplified casino rules. The game begins with the player specifying his bank limit, a value that will be rounded to the nearest dollar amount. If the bank specified by the player is any value less than or equal to 0, the bank will be set to \$100. Next the player must place his bet, which may be as much or all of his bank as he wishes (to end game enter 0). On the players first roll after a bet, the dice are automatically rolled, and the value of each, along with their total, is displayed by the HP-71. If the total of the dice equals 7 or 11 on the first roll, the player wins and his bet is added to his bank. If the total of the dice equals 2,3 or 12 on the first roll, the amount of his bet is subtracted from his bank. Any other total on the first roll becomes the players "point". The player then continues to roll until the total of the dice equals his "point" (the player wins) or the total equals 7 or 11 (the player loses).

If a 'point' has been established the HP-71 displays "More bets? (S,H,N)". The player now has the option to press:

S - SIDE BET. The bet is that the player will make his point. The House pays 1:1 if the point is 6 or 8; 3:2 if the point is 5 or 9; and 2:1 if the point is 4 or 10.

H - HARD WAY BET. If the point is 4, 6, 8, or 10 then the player can bet he will make his point in the form of doubles. The House pays as follows: 8:1 if the point is 4 or 10; 10:1 if the point is 6 or 8.

N - NO BET. Roll again, no additional betting is desired.

User Instructions

Comments	Input	Display
1) Run the program.		What's your bank? \$_
2) Enter your starting Bank. If the number entered here is less than or equal to 0, the program will set the bank to \$100.	0<bank<99999	You have \$b.
3) The program displays your current bank.		Place your bet. \$_
4) Place your bet. If the bet exceeds your current bank value, the program cycles to step 2 of these instructions.	0<=bet<=bank	Your roll is d,d=t
If the player's bet is 0 then the game is ended.		Game Over.
5) The dice are rolled. If the total is 7 or 11: If the total is 2, 3 or 12: For either of these occurrences, the program cycles to step 2 of these instructions.		You won!! Sorry, you lost.
If the total is not one of the above:		Your point is p More bets? (S,H,N)
6) Your point is fixed and you are now trying to match it. You have the option at this point of placing a side or hard bet. These bets are described on page 6 of this book. Note that your bets are subtracted from your bank as they are made. Your total accumulated bet is kept track of. The program will not allow you to bet more than your bank.		
Press [S] to make a side bet and go to step 7 of these instructions.		Amount of bet? b

Press [H] to make a hard bet and go to step 8 of these instructions.

Amount of bet? b

Press [N] for no bets and go to step 9 of these instructions.

Your roll is d,d= t

- 7) Having elected to place a side bet, the program asks for the amount of the bet.

$0 \leq \text{bet} \leq \text{bank}$

If the bet is larger than your bank, the program displays:
and execution continues with step 6 of these instructions.

You only have \$b.

After entering the bet, the program cycles to step 6.

More bets? (S,H,N)

- 8) Having elected to place a hard bet, the program asks for the amount of the bet.

$0 \leq \text{bet} \leq \text{bank}$

If the bet is larger than your bank, the program displays:
and execution continues with step 6 of these instructions.

You only have \$b.

If the point is not even, the program displays:

Point is odd.
More bets? (S,H,N)

and execution continues with step 6 of these instructions.

After entering the bet, the program cycles to step 6.

More bets? (S,H,N)

- 9) The dice are rolled.
If the total is 7 or 11: is displayed and the program cycles to step 4 of these instructions.

Sorry, you lost.

Place your bet. \$b

If you've lost, and your bank is 0: is displayed and the program terminates.

You're broke! Go home!

If the total is your point:
is displayed and the program
cycles to step 4 of these
instructions.

If the total is not one of
the above:

and the program cycles to
step 6 of these instructions.

You won!!

Place your bet. \$b

Your point is p
More bets? (S,H,N)

Listing

```
10 DEF FNK$(D0$,K0$)
20 DISP D0$
30 'KEY': K1$=KEY$
40 IF NOT POS(K0$,K1$) THEN "KEY"
50 FNK$=K1$
60 END DEF
70 !
80 DEF FNF$(A)
90 D$=PEEK$("2F6DC",2) @ STD
100 FNF$=STR$(A)
110 POKE "2F6DC",D$
120 END DEF
130 !
140 INTEGER B,B1,D1,D2,F,H,P,R,S
150 DIM D$(2),D1$(4),K$(1),K1$(3)
160 D1$=PEEK$("2F946",4)
170 DELAY 2
180 INPUT "What's your bank? $";B
190 IF B<=0 THEN B=100
200 RANDOMIZE
210 !
220 'ROLL1': DISP "You have $";FNF$(B);"."
230 INPUT "Place your bet. $";FNF$(F);F
240 IF F<0 OR F>B THEN "ROLL1"
250 IF NOT F THEN "END"
260 S=0 @ H=0
270 GOSUB "ROLL"
280 IF R=7 OR R=11 THEN "NATURAL"
290 IF R=2 OR R=3 OR R=12 THEN "CRAPS"
300 P=R
310 DISP "Your point is "&FNF$(P)
320 'ROLLPT': GOSUB "MOREBETS"
330 GOSUB "ROLL"
340 IF R=7 OR R=11 THEN "CRAPS"
350 IF R#P THEN "ROLLPT"
360 IF D1#D2 THEN F=F-H @ H=0
370 IF P=5 OR P=9 THEN S=IP(1.5*S)
380 IF P=4 OR P=10 THEN S=2*S @ H=8*H
390 IF P=6 OR P=8 THEN H=10*H
400 'NATURAL': DISP "You won!!"
410 B=B+F+S+H
420 GOTO "ROLL1"
430 !
440 'MOREBETS': K$=FNK$("More bets? (S,H,N)","SHN")
450 IF K$="N" THEN RETURN
460 IF K$="H" AND MOD(P,2) THEN DISP "Point is odd." @ GOTO 'MOREBETS'
470 INPUT "Amount of bet? ";FNF$(B1);B1
480 B1=ABS(B1)
490 IF B1>B-F-S-H THEN DISP "You have $";FNF$(B-F-S-H);"." @ GOTO 'MOREBETS'
500 IF K$="H" THEN H=H+B1 ELSE S=S+B1
510 GOTO 'MOREBETS'
520 !
530 'ROLL': D1=IP(RND*6)+1
```

```
540 D2=IP(RND*6)+1
550 R=D1+D2
560 DISP "Your roll is ";FNF$(D1);", ";FNF$(D2);"= ";FNF$(R)
570 RETURN
580 !
590 'CRAPS': DISP "Sorry, you lost."
600 B=B-F-S-H
610 IF B THEN "ROLL1"
620 DISP "You're broke! Go home!"
630 'END': IF B THEN DISP "Game Over."
640 POKE "2F946",D1$
650 END
```

Hangman

Hangman is a word guessing game. This game is a version of the popular word game "hangman". The first player selects a word or phrase that is as many as 19 characters in length and keys it into the computer. The second player guesses various characters until he completes the word or gets hanged.

This version allows spaces, numerals, and other special characters in the guess word though they will not be hidden. For example, the phrase "The HP-71B" would be displayed for guessing as "### #-71#". There are three different skill levels, Expert, Intermediate, and Novice, which offer different numbers of guesses per phrase. Expert allows 1 guess per letter in the word, Intermediate allows 50% more guesses than letters, and Novice allows 2 guesses per letter in the word.

User Instructions

Comments	Input	Display
=====	=====	=====
1) Run the program.		Word: _
2) The program is asking the first player to input the word or phrase that the second player must guess.	up to 19 chars.	
If the input was greater than 19 or less than 2, the program displays:		1<word length<20 Word: _
and step 2 of these instructions is repeated.		
Otherwise:		Skill level (N,I,E) ?
3) The program is prompting player 2 for his skill level: Novice, Intermediate, or Expert.	N, I, or E	G:####
4) The display shows the number of guesses left and a # for each alphabetic character in the word or phrase. The player must enter either a single character or make an attempt to guess the whole word or phrase.		
To guess a letter, simply press the corresponding key. The guess counter will decrement and if the character exists in the phrase, it will be displayed.		
To attempt the whole phrase, press [END LINE] and see:		Guess: _
5) Enter your guess of the complete phrase and press [END LINE]. If the guess is incorrect, see:		Sorry. G:####
and the program continues at step 4 of these instructions.		
6) If all of the letters of the word are displayed, or your guess at the phrase is		

correct, see:

To start an new game, press
[Y] and go to step 1 of these
instructions.

To quit the game, press [N]
and see:
and the program halts.

- 7) If you run out of guesses,
see:

To start an new game, press
[Y] and go to step 1 of these
instructions.

To quit the game, press [N]
and see:
and the program halts.

Correct in G tries.
The word was: word
Play again (Y,N)?

Game over.

Sorry.
You lose.
The word was: word
Play again (Y,N)?

Game over.

Listing

```
10 DEF FNF$(N)
20 D9$=PEEK$("2F6DC",2) @ STD @ FNF$=STR$(N) @ POKE "2F6DC",D9$
30 END DEF
40 !
50 DEF FNK$(D9$,K9$)
60 DISP D9$
70 'RPT': K8$=KEY$ @ IF K8$="#38" THEN K8$="" @ GOTO "OUT"
80 IF NOT POS(K9$,K8$) THEN "RPT"
90 'OUT': FNK$=K8$
100 END DEF
110 !
120 INTEGER G,G1,I,L
130 DIM A$[26],K$[1],K8$[4],W$[90]
140 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
150 'START': LINPUT "Word: ";W$
160 L=LEN(W$)
170 W$=UPRC$(W$)
180 DIM D$[L],W2$[L]
190 D$=""
200 FOR I=1 TO L
210 IF NOT POS(A$,W$[I,I]) THEN D$=D$&W$[I,I] @ L=L-1 ELSE D$=D$&"#"
220 NEXT I
230 IF L>19 OR L<2 THEN DISP "1<word length<20" @ WAIT 1 @ GOTO "START"
240 DISP CHR$(27)&"H"&CHR$(27)&"J";
250 K$=FNK$("Skill level (N,I,E) ?","NIE")
260 G=L
270 IF K$="N" THEN G=2*L
280 IF K$="I" THEN G=4*L DIV 3
290 G1=G
300 'IN': IF NOT G OR NOT POS(D$,"#") THEN 'END' ELSE K$=FNK$(FNF$(G)&
": "&D$,A$)
310 IF K$="" THEN "GUESS"
320 P=POS(W$,K$)
330 'FIND': IF NOT P THEN G=G-1 @ GOTO 'IN'
340 D$[P,P]=W$[P,P]
350 P=POS(W$,K$,P+1)
360 GOTO 'FIND'
370 'GUESS': LINPUT "Guess: ";W2$ @ W2$=UPRC$(W2$) @ G=G-1 @ GOTO 'G2'
380 'END': W2$=D$
390 'G2': IF W2$=W$ THEN DISP "Correct in ";FNF$(G1-G);" tries." ELSE
DISP "Sorry."
400 WAIT 1
410 IF G AND W$#W2$ THEN 'IN'
420 IF W$#W2$ THEN DISP "You lose." @ WAIT 1
430 DISP "The word was: ";W$ @ WAIT 1
440 IF FNK$("Play again (Y,N)?","YN")="Y" THEN 'START'
450 DISP "Game over." @ WAIT 1
460 PUT "#38"
470 END
```

Blackjack

In this card game, the HP-71 is the dealer and up to eight people may play. Each player begins with \$200 and may bet as much or all of it as he wishes up to \$200. Before the cards are dealt, each player enters his bet (to leave the game, enter \$0 for a bet). Then the dealer deals the cards from a deck of 104 (double deck). If the dealer is holding a natural, he immediately collects the bets of all players who do not have naturals. (A natural is an ace and any face card or ten, giving a count of 21 in two cards. A natural is also called BLACKJACK.) If the dealer and any other player both have naturals, the bet is a standoff and the player keeps his money. If the dealer does not have a natural, the game continues with each player deciding what he will do with his hand. Aces may be counted as 1 or 11 as the player wishes. All face cards count as 10 (K, Q, J) and any other cards count as their pip value (numerical value), i.e., "T" counts as 10, 9 as 9, 8 as 8 and so on.

Each player has the option to hit (H), double (D), split (S), or stand (S). A hit means he would like another card to add to the ones he already has. A double (down-for-double) means he wants only one more card and he is doubling his original bet. This is only done if the player has a total of 11 with his first two cards, taking a chance on getting a 10 or a face card to total 21. A split means the player has two identical cards, i.e., two queens, two fives, etc., and would like to separate them into two different hands drawing two more cards to go with the two like cards. The bet that the player originally made is then placed also on the second hand. In other words, he is now playing for twice the money. The player must play each hand separately, finishing his first hand before going on to the second. A stand means that the player is satisfied with the cards that he has.

When all players are done, the dealer settles his hand. If his total is 17 or more, he must stand. If his total is less, he must take a card and must continue to take cards until his total is 17 or more, at which point he must stand. If the dealer has an ace, and counting it as 11 would bring his total to 17 or more (but not over 21), he must count the ace as 11 and stand.

When the dealer is done, all bets are settled. Any player with a natural wins 1.5 times his bet. Any player whose total went over 21 loses his bet. If the dealer went over 21, he pays to each player under 21 the amount of his bet. If the dealer stands at 21 or less, he pays the bet of any player having a higher total (not over 21), collects the bet of any player with a lower total, and is at a standoff ("PUSH") with any player who has the same total.

User Instructions

Comments	Input	Display
=====	=====	=====
1) Run the program.		BLACKJACK Number of players? _
2) Enter the number of players. 1-8 [RTN]		Shuffling Place your bets!
3) If one player: If more than one player: Bets must be greater than \$10 and less than \$200. If a player enters 0 then his total winnings are displayed: and he is effectively out of the game. If all players enter 0, then the game ends (go to step 11 of these instructions).		bet \$0 Player n bet \$0 Total \$(bank)
If the bet exceeds the \$200 limit or is less than \$10: and step 3 of these instructions is repeated.		10<=Bet<=200
If the player bets more than his bank: and step 3 of these instructions is repeated.		You only have \$(bank)
The program procedes to deal the cards. If the dealer has a natural then settle all bets immediately - go to step 10 of these instructions.		DEALER NATURAL: c c
Otherwise:		Player n d{} c c :_
4) The program displays the dealer's first card face up and second card face down, followed by the player's cards face up.		
The player's options are: for Hit, Double, Stand and Split, respectively.	H, D, S or / [END LINE]	
If "H" then go to step 5.		

If "D" then go to step 6.
If "/" then go to step 7.
If "S" then go to step 8.

This step is repeated until
all players have played
their hands. When all hands
are played, go to step 9.

- 5) HIT. Take another card
from the deck. If the card
total is over 21, see:
otherwise go to step 4.

d{} c c c BUST !
d{} c c c :

- 6) DOUBLE. If the hand value
is 11, the player may opt
to take one card and double
the amount of the original
bet.

If the player does not have
enough money to double the
bet, see:
and go to step 4.

Not enough money.

If the hand value is not 11,
or the number of cards is
greater than 2, or the hand
being played is part of a split,
see:
and go to step 4.

CANNOT DOUBLE

If the player can double:
This player's hand is
complete. Go to step 4
for the next player.

d{} c c c

- 7) SPLIT. If the player's first
two cards are identical, he
may choose to play each card
as a separate hand.

If the player does not have
enough money to double the
original bet, see:
and go to step 4.

Not enough money.

If the cards are not identical,
or the number of cards is
greater than 2, or the hand
being played is already part of
a split, see:
and go to step 4.

CANNOT SPLIT

If the player can split:
Play the 1st hand using one
of the identical cards. Use

Play hand 1

the same options as in step
4 except that you cannot
split again or double.
Then play the 2nd hand.
Go to step 4 for the next
player.

Play hand 2

8) STAND. Take no more cards.
If all players have played,
goto step 9. Otherwise,
goto step 3 for the next player.

9) After all players have played,
the dealer plays his hand. The
program displays the dealer's
cards and his total score.
If the dealer's hand is greater
than 21, see:

dd.. =tt

dd.. BUST !

10) SETTLE BETS. If a player has
busted, then he loses.

Player n LOSES \$(bet)

If he has a total greater
than the dealer's and less
than or equal to 21, then he
wins.

Player n WINS \$(bet)

If he has a natural, he wins
1.5 times his bet.

Player n has BLACKJACK
WINS \$(1.5*bet)

If the player's total is less
than the dealer's, he loses.

Player n LOSES \$(bet)

If the player and dealer tie
at or below 21, the player
keeps his bet.

Player n PUSH \$(total)

If the player's hand was split,
his winnings (or losses) are
combined. Should he end up
even:

Player n EVEN \$(total)

After the outcome, the player's
bank is displayed:

Bank \$(bank)

If a player loses all of his
money (bank < \$10), then he
must quit.

Player n is out.

If all players' banks are
less than \$10, go to step 11.

After settling bets, goto
step 3.

11) END OF GAME. You have the

Play again? _

option to run the program
again.

To restart the game, press
"Y" and go to step 1. Y [END LINE]

Otherwise, press "N" to end
the game. N [END LINE] End of Game

Listing

```
10 ! BLACKJACK
20 INTEGER M,C,I,L,P,C0,A,J,Q,K,T,A0,B0,B2,S
30 DIM D$(104),D1$(4),O$(3),Q$(3)
40 O$=PEEK$("2F6DB",3) @ D1$=PEEK$("2F946",4)
50 OPTION ROUND NEAR
60 OPTION BASE 0
70 A=11 @ T=10 @ J=T @ Q=T @ K=T
80 STD
90 DELAY 2,0
100 !
110 DISP "BLACKJACK"
120 'PLYR': INPUT "Number of players? ";M
130 IF M<1 OR M>8 THEN DISP '1<=Players<=8' @ GOTO 'PLYR'
140 !
150 CALL SHUFFLE(D$,C)
160 !
170 INTEGER B(M)
180 FOR I=0 TO M @ B(I)=200 @ NEXT I
190 !
200 'NEXTHAND':
210 !
220 B2=0
230 FOR I=1 TO M @ B2=B2+B(I) @ NEXT I
240 IF NOT B2 THEN 'ENDGAME'
250 !
260 DESTROY H$,N,H,B1
270 DIM H$(2*M)[12]
280 INTEGER N(M),H(2*M),B1(M)
290 !
300 DISP 'Place your bets!'
310 B2=0
320 FOR I=1 TO M
330 IF B(I)=0 THEN 'NXTBET'
340 'BET': IF M#1 THEN DISP "Player";I;
350 INPUT "bet $","0";B1(I)
360 IF B1(I)=0 THEN DISP "Total $"&STR$(B(I)) @ B(I)=0 @ GOTO 'NXTBET'
370 IF B1(I)<10 OR B1(I)>200 THEN DISP "10<=Bet<=200" @ GOTO 'BET'
380 IF B1(I)>B(I) THEN DISP "You only have $"&STR$(B(I)) @ GOTO 'BET'
390 B2=B2+B1(I)
400 'NXTBET': NEXT I
410 IF NOT B2 THEN 'ENDGAME'
420 !
430 FOR I=0 TO M
440 IF NOT B(I) THEN 'NXTHND'
450 FOR L=1 TO 2
460 IF C=104 THEN CALL SHUFFLE(D$,C)
470 C=C+1
480 H$(I)[L]=D$(C,C)
490 H(I)=H(I)+VAL(H$(I)[L])
500 NEXT L
510 IF H(I)=21 THEN N(I)=1
520 'NXTHND': NEXT I
530 !
540 IF N(0) THEN GOSUB 'DLRNTRL' @ GOTO 'HANDOVER'
```

```

550 FOR P=1 TO M
560 I=P @ S=0
570 IF NOT B(I) THEN 'NXTP'
580 IF M#1 THEN DISP 'Player';I
590 C0=2
600 'SHOW': GOSUB 'DISPHAND'
610 'CARD': GOSUB 'IPT'
620 ON POS('SHD/',Q$)+1 GOSUB 'SHOW','STAND','HIT','DOUBLE','SPLIT'
630 IF S THEN RETURN
640 'NXTP': NEXT P
650 GOTO 960
660 !
670 'STAND': H(I)=0 @ A0=0
680 FOR L=1 TO C0
690 H(I)=H(I)+VAL(H$(I)[L,L])
700 IF H$(I)[L,L]='A' THEN A0=A0+1
710 NEXT L
720 IF A0>1 THEN H(I)=H(I)-10*A0
730 IF A0=1 AND H(I)>21 THEN H(I)=H(I)-10
740 IF H(I)>21 THEN DISP 'BUST !'
750 RETURN
760 !
770 'HIT': GOSUB 'CHECK' @ IF H(I)>21 THEN RETURN ELSE POP @ GOTO 610
780 !
790 'DOUBLE':
800 IF S OR C0#2 OR VAL(H$(I)[1,1])+VAL(H$(I)[2,2])#11 THEN DISP "CANN
OT DOUBLE" @ GOTO 'SHOW'
810 IF B1(I-M*(I>M))*2>B(I-M*(I>M)) THEN DISP "Not enough money." @ PO
P @ GOTO 'SHOW'
820 B1(I-M*(I>M))=B1(I-M*(I>M))*2 @ GOSUB 'CHECK' @ DISP @ RETURN
830 !
840 'SPLIT':
850 IF H$(I)[1,1]#H$(I)[2,2] OR S OR C0>2 THEN DISP "CANNOT SPLIT" @ P
OP @ GOTO 'SHOW'
860 IF B1(I-M*(I>M))*2>B(I-M*(I>M)) THEN DISP "Not enough money." @ PO
P @ GOTO 'SHOW'
870 S=1 @ H$(M+I)=H$(I)[2]
880 FOR X=0 TO 1
890 I=X*M+I @ C0=1
900 DISP "Play hand";X+1
910 GOSUB 'CHECK' @ GOSUB 'CARD'
920 NEXT X
930 S=0
940 RETURN
950 !
960 B2=0
970 FOR I=1 TO 2*M @ B2=B2+H(I)*(H(I)<22) @ NEXT I
980 IF NOT B2 THEN H(0)=21 @ GOTO 'HANDOVER'
990 !
1000 C0=2 @ I=0 @ GOSUB 'STAND'
1010 IF H(0)>16 THEN GOSUB 'DISPHAND' @ GOTO 'DDISP'
1020 FOR X=1 TO 10
1030 GOSUB 'CHECK'
1040 IF H(0)>16 THEN 'DDISP'
1050 NEXT X
1060 'DDISP': IF H(0)<=21 THEN DISP ' =';H(0)
1070 !

```



```

1080 'HANDOVER': FOR I=1 TO M
1090 IF NOT B(I) THEN 'NEXTP'
1100 IF N(I) AND NOT N(0) AND H(0)#21 THEN B0=1.5*B1(I) @ GOTO 'BJ'
1110 B0=0
1120 FOR L=0 TO 1-NOT H(M+I)
1130 IF H(I+M*L)>21 THEN B0=B0-B1(I) @ GOTO 'NXT'
1140 IF H(0)>21 THEN B0=B0+B1(I) @ GOTO 'NXT'
1150 IF H(I+M*L)<H(0) THEN B0=B0-B1(I) @ GOTO 'NXT'
1160 IF H(I+M*L)>H(0) THEN B0=B0+B1(I)
1170 'NXT': NEXT L
1180 'BJ': DISP 'Player';I;
1190 IF N(I) THEN DISP "has BLACKJACK"
1200 IF NOT N(I) AND N(0) THEN DISP ", "&H$(I)[1,1]&" "&H$(I)[2,2]
1210 IF B0>0 THEN DISP 'WINS $'&STR$(B0) @ GOTO 'BANK'
1220 IF B0<0 THEN DISP 'LOSES $'&STR$(-B0) @ GOTO 'BANK'
1230 IF H(M+I) THEN DISP 'EVEN' @ GOTO 'BANK'
1240 DISP 'PUSH'
1250 'BANK': B(I)=B(I)+B0
1260 DISP 'Bank $'&STR$(B(I))
1270 IF B(I)<10 THEN B(I)=0 @ DISP 'Player';I;'is out.'
1280 'NEXTP': NEXT I
1290 GOTO 'NEXTHAND'
1300 'DISPHAND': IF I THEN DISP H$(0)[1,1];CHR$(31);' '; ELSE DISP
1310 FOR J1=1 TO C0
1320 DISP H$(I)[J1,J1];' ';
1330 NEXT J1
1340 RETURN
1350 !
1360 'IPT': INPUT ':';Q$ @ Q$=UPRC$(Q$[1,1]) @ RETURN
1370 !
1380 'CHECK': C0=C0+1
1390 IF NOT S THEN N(I)=0
1400 IF C=104 THEN CALL SHUFFLE(D$,C)
1410 C=C+1
1420 H$(I)[C0]=D$[C,C]
1430 GOSUB 'DISPHAND'
1440 GOSUB 'STAND'
1450 RETURN
1460 !
1470 'DLRNTRL':
1480 FOR I=1 TO 3 @ BEEP 500 @ BEEP 600 @ NEXT I
1490 DISP 'DEALER NATURAL: '&H$(0)[1,1]&' '&H$(0)[2,2]
1500 RETURN
1510 !
1520 'ENDGAME': ON ERROR GOTO 1530
1530 'PA': INPUT 'Play again? ';Q$
1540 ON POS('YN',UPRC$(Q$[1,1]))+1 GOTO 'PA','PLYR','EOG'
1550 'EOG': DISP 'End of Game'
1560 POKE "2F6DB",O$ @ POKE "2F946",D1$
1570 PUT "#38"
1580 END
1590 !
1600 SUB SHUFFLE(D$,C)
1610 DELAY 0
1620 DISP 'Shuffling'
1630 D$='A23456789TJQK'
1640 D$=D$&D$&D$&D$&D$&D$&D$&D$

```

```
1650 RANDOMIZE
1660 FOR Z=1 TO 104
1670 Z1=IP(104*RND)+1
1680 D1$=D$[Z,Z]
1690 D$[Z,Z]=D$[Z1,Z1]
1700 D$[Z1,Z1]=D1$
1710 NEXT Z
1720 DELAY 2,0
1730 C=0
1740 END SUB
```

Hamurabi

This game allows a player to control a country's economy through the buying and selling of land. The more efficiently he uses the land, the better a governor he is.

The game begins with a report of the economy and the population. The player then has the opportunity to buy and sell land, allocate food, and plant a number of acres for harvest. The player must deal with plagues, starvation, rats, and the rise and fall of the land market.

If the player rules unsuccessfully, he will be warned and asked if he wishes to continue. He may resign at any time by selling his land.

The object of the game is to determine how it works and find the best set of circumstances for a growing economy.

Note that the HP-71 DELAY setting is not altered by the program. If the displays proceed too slowly or rapidly, stop the program and reset the DELAY to something more appropriate. A DELAY of 3 may be satisfactory.

User Instructions

Comments	Input	Display
=====	=====	=====
1) Run the program. 1a)		HAMURABI Last year: n people starved, n people immigrated. Population is nnn. nnnn bushel harvest, at n bushels/acre. Rats ate nnn bushels. nnnn bushels stored. You own nnnn acres, worth nn bushels/acre. Hamurabi ... Acres to buy? _
1b)		
2) The program is asking how many acres of land to buy at the current rate in bushels per acre. If 0 acres are bought, go to step 3, otherwise go to step 4.	# of acres [END LINE]	Acres to sell? _
3) The program is asking how many acres of land to sell at the current rate in bushels per acre.	# of acres [END LINE]	Bushels for food? _
4) The program is asking how many bushels of food to distribute to the people over the year.	# of bshls [END LINE]	Acres to plant? _
5) How many acres do you want to plant, given the bushels in storage, the number of people you have, the number of acres you own, and your future food needs?	# of acres [END LINE]	
If at any time in steps 2 through 4 the program deems one of your inputs inappropriate, you would see:		Think again, you have: nnn people, nnnn acres, and nnnn bushels stored.

and the prompt would be repeated.

Steps 1a through 5 are repeated until some outstanding event occurs.

- 6) Occasionally, your people will be hit with a plague that will sharply reduce their numbers. When it happens, this display will appear immediately before the report of your total population.

Plague killed nnn.

- 7) If things look especially bleak one year, after the state of the people report you will see this message: and the program will go to step 8 of these instructions.

Things look grim!

- 8) Every fifth year and on especially bad years you are given the option to quit.

You've ruled n years.
Continue? _

Responding with "Y" continues your current reign. Go to step 1b. Otherwise select "N" and go to step 9. "Y" or "N"
[END LINE]

Start over? _

- 9) You can restart the game, if you desire, by pressing "Y" [END LINE] and going to step 1. Pressing "N" [END LINE] terminates the game.

"Y" or "N"
[END LINE]

END OF GAME

Listing

```
10 ! HAMURABI
20 INTEGER A1,A2,A3,B1,B2,B3,B4,C1,I,J,Z
30 'START': DISP 'HAMURABI' @ STD
40 A1=100 @ A2=5 @ A3=0 @ B1=2800 @ B2=200 @ B3=3 @ B4=3000 @ C1=1000
   @ J=1 @ Z=0
50 !
60 'NEXT': BEEP 600,.15 @ BEEP 400,.1 @ BEEP 600,.15
70 Z=Z+1
80 DISP 'Last year:'
90 DISP STR$(A3)&' people starved,'
100 DISP STR$(A2)&' people immigrated.'
110 IF J<=0 THEN DISP 'Plague killed '&STR$(A1-A1 DIV 2)&'.' @ A1=A1 D
   IV 2
120 DISP 'Population is '&STR$(A1)&'.'
130 DISP STR$(B4)&' bushel harvest,'
140 DISP 'at '&STR$(B3)&' bushels/acre.'
150 DISP 'Rats ate '&STR$(B2)&' bushels.'
160 DISP STR$(B1)&' bushels stored.'
170 DISP 'You own '&STR$(C1)&' acres,'
180 C2=17+IP(6*RND)
190 DISP 'worth '&STR$(C2)&' bushels/acre.'
200 BEEP 600
210 DISP 'Hamurabi ...'
220 IF Z>1 AND A3>(A1+A3)*.45 THEN DISP "Things look grim!" @ GOSUB 'Y
   RS'
230 IF NOT RMD(Z,5) THEN GOSUB 'YRS'
240 !
250 'BUYLAND': INPUT 'Acres to buy? ';I
260 IF I<0 THEN 'BUYLAND'
270 IF I=0 THEN 'SELLLAND'
280 J=I*C2
290 IF J>B1 THEN GOSUB 'ERR' @ GOTO 'BUYLAND'
300 B1=B1-J
310 C1=C1+I
320 GOTO 'FOOD'
330 !
340 'SELLLAND': INPUT 'Acres to sell? ';I
350 IF I<0 THEN 'SELLLAND'
360 IF I=0 THEN 'FOOD'
370 IF I>C1 THEN GOSUB 'ERR' @ GOTO 'SELLLAND'
380 IF I=C1 THEN 'RESTART'
390 C1=C1-I
400 B1=B1+C2*I
410 !
420 'FOOD': INPUT 'Bushels for food? ';I
430 IF I<0 THEN 'FOOD'
440 IF I>B1 THEN GOSUB 'ERR' @ GOTO 'FOOD'
450 B1=B1-I
460 A3=A1-I DIV 20
470 A2=0
480 IF A3<0 THEN A2=-A3/2 @ A3=0
490 !
500 'PLANT': INPUT 'Acres to plant? ';I
510 IF I<0 THEN 'PLANT'
```

```

520 J=I DIV 2
530 IF I>C1 OR J>B1 OR I>10*A1 THEN GOSUB 'ERR' @ GOTO 'PLANT'
540 B1=B1-J
550 !
560 B3=IP(5*RND)+1
570 B4=B3*I
580 B2=IP((B1+B4)*.07*RND)
590 B1=B1-B2+B4
600 J=IP(10*RND)
610 A2=IP(A2+(5-B3)*B1/600+1)
620 IF A2>50 THEN A2=50
630 A1=A1+A2-A3
640 GOTO 'NEXT'
650 !
660 'ERR': BEEP 100,.2
670 DISP 'Think again, you have:'
680 DISP STR$(A1)&' people,'
690 DISP STR$(C1)&' acres, and'
700 DISP STR$(B1)&' bushels stored.'
710 RETURN
720 !
730 'YRS': DISP "You've ruled";Z;"years."
740 INPUT 'Continue? ';Q$
750 Q$=UPRC$(Q$[1,1])
760 IF Q$='N' THEN POP @ GOTO 'RESTART'
770 IF Q$# 'Y' THEN 'YRS'
780 RETURN
790 !
800 'RESTART': INPUT 'Start over? ';Q$ @ Q$=UPRC$(Q$[1,1])
810 IF Q$='Y' THEN 'START'
820 IF Q$# 'N' THEN 'RESTART'
830 DISP 'END OF GAME' @ PUT "#38"
840 END

```

Space War

"Space War" sets up a two-dimensional playing area where you do battle with from 16 to 30 enemy ships. Your ship is equipped with long- and short-range sensors, shields, and 20 torpedoes. You also have 5 starbases at which you may effect repairs to your ship. The object of the game is to destroy all of the enemy ships before they destroy you.

The playing field is a two dimensional array, (0,0) to (9,9), of quadrants. Each quadrant is itself a two-dimensional array, (0,0) to (9,9), of sectors. You may "jump" from location to location on the playing field but are not permitted to leave the field nor to jump on top of and enemy ship or starbase.

										9
										8
										7
										6
										5
										4
										3
										2
										1
										0
0	1	2	3	4	5	6	7	8	9	

The grid above may be used to describe the quadrants in the playing field, or the sectors within a quadrant. Locations are displayed by the program as [c,r] where c is the column and r is the row in the above grid.

Short range scans are used to examine the sectors in a quadrant. Long range scans are used to examin both the quadrant you are in and the eight adjacent quadrants. For example, if your location is displayed as "Quad. [3,4] Sect. [5,9]", a short range scan would reveal all objects in quadrant [3,4] (column 3, row 4), and a long range scan would show all objects in quadrants [2,3], [2,4], [2,5], [3,3], [3,4], [3,5], [4,3], [4,4], and [4,5].

Torpedoes may be fired at any time and will follow the course that you have previously set for the ship. In other words, the "torpedo" command launches a torpedo in the direction set by the "course" command. A torpedo will follow its course until it either meets an

enemy ship or meets the boundary of your quadrant. A torpedo can not be fired into a neighboring quadrant. Torpedos cannot destroy starbases.

You may also be fired upon from time to time by enemy ships. Only those ships in your quadrant may fire upon you. Enemy fire may miss. If it doesn't, your ship has sustained damage. The damage may be either in the form of the weakening of your shields, the damaging of your ship, or both. You may find that you can survive attacks even without shields, but if in the process your sustained damage reaches 100%, the game is over.

You may check your current status at any time by using the "status" command. This command displays your current location, number of torpedoes, per cent effectiveness of your shields, warp (jump distance) setting, heading, sustained damage and remaining enemy ships.

Docking may only be performed when your ship is in a sector adjacent to a starbase. Upon docking, your stock of torpedoes will be increased to 20, your shields increased to 100% efficiency, and your damaged decreased to 0%.

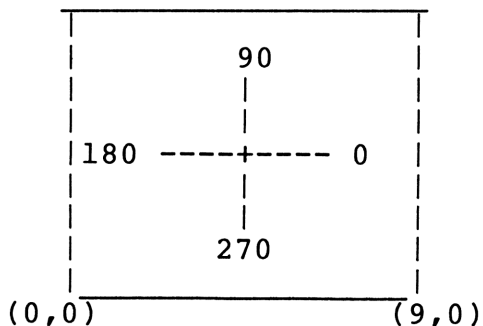
Note that the program does not alter the machine's DELAY setting. If displayed information passes too quickly or too slowly, halt the program and alter the DELAY to your liking. Likewise, the program does not alter the state of the tone generator. If you execute BEEP ON or BEEP OFF, the program will either beep or not. If you execute SFLAG -25 or CFLAG -25, the beeps will either be loud or soft, as you prefer.

User Instructions

Comments	Input	Display
1) Run the program.		Skill level (0-9)?
2) Enter your skill level. 0 is no experience, 9 is expert.	0 - 9	Setting up Cmnd: C,D,J,L,Q,S,T,?
3) You are at the command level of the program. At this point you can exercise several different options. The letters in the display correspond to these options as follows: C - COURSE : step 4. D - Dock : step 5. J - Jump : step 6. L - Long range scan : step 7. Q - Quit : step 8. S - Short range scan : step 9. T - Torpedo : step 10. ? - Status : step 11. Select an option and go to the appropriate step.		
4) COURSE. Enter an angle and a distance. The angle is your intended direction of travel. Given the 9 x 9 quadrant layout of the playing area, the course angle moves your ship in the direction indicated below.		COURSE, speed: 0,0

(0,9)

(9,9)



Note that you may use
either radians or degrees

depending on your current
OPTION ANGLE setting.

The distance input is the
number of quadrants to be
jumped. The fractional
portion of the number is the
number of sectors to jump.
The COURSE command will not
allow inputs for distance of
greater than 9: the prompt
will be repeated.

The program returns to the
command level (step 3).

Cmnd: C,D,J,L,Q,S,T,?

5) DOCK.

If your ship is in a sector
that is immediately adjacent
to a starbase, you may dock:
Docking restores your
torpedoes to 20, your shields
to 100%, and reduces your
damage to 0%.

Docked

If you are not close enough
to the starbase, you cannot
dock:

Cannot Dock

The program returns to the
command level (step 3).

Cmnd: C,D,J,L,Q,S,T,?

6) JUMP.

The ship moves to a new
quadrant and/or sector
based on the information
set by the COURSE command.

Jumps that would place your
ship out of the playing field,
or on top of another ship or
starbase are not allowed:

Bad Jump!

Successful jumps result in the
displaying of your new
coordinates:

Quad [q,q] Sect [s,s]

The program returns to the
command level (step 3).

Cmnd: C,D,J,L,Q,S,T,?

7) LONG RANGE SCAN.

This command searches your
present quadrant and all
adjacent quadrants for starbases
and enemy ships. It first
reports your present location,
and then lists what it finds.

Quad [q,q] Sect [s,s]

If no objects are found by the scan, the program will display:

The program returns to the command level (step 3).

8) QUIT.

The program terminates.
No warnings or options to recover are given.

9) SHORT RANGE SCAN.

This command searches your present sector for starbases and enemy ships. It first reports your present location, and then lists what it finds.

If no objects are found by the scan, the program will display:

The program returns to the command level (step 3).

10) TORPEDOES.

Torpedoes are launched in the direction specified in the COURSE command. The distance parameter is irrelevant. Once launched, torpedoes will not travel from your current quadrant to another quadrant. Only enemy ships within your quadrant may be fired upon.

A successful torpedo attack will result in the displays:

An unsuccessful attack displays:

An attempt to fire a torpedo when none are left will have no result. No message

Enemy Ship Quad. [q,q]
.
.
Star Base Quad. [q,q]

Clear

Cmnd: C,D,J,L,Q,S,T,?

Quad [q,q] Sect [s,s]
Enemy Ship Sect. [s,s]
.
.
Star Base Sect. [s,s]

Clear

Cmnd: C,D,J,L,Q,S,T,?

Enemy Ship [s,s]

Torpedo missed

will be displayed.

The program returns to the command level (step 3).

Cmnd: C,D,J,L,Q,S,T,?

11) STATUS.

This command displays a summary of the current status of the game.

Quad [q,q] Sect [s,s]
Torpedoes : nn
Shields : nnn%
Warp : n.n
Heading : nnn.nnnn
Damage : nnn%
Enemy Ships : nn

The program returns to the command level (step 3).

Cmnd: C,D,J,L,Q,S,T,?

NOTE: After the execution of any command, you may be attacked by an enemy ship, if one exists in your quadrant. The attack may or may not inflict damage upon your ship.

A non-injurious assault will be displayed as:

Attacked: Missed

A damaging attack will be displayed as:

Attacked: Sect. [s,s]

If more than one enemy ship exists in your quadrant, you may be attacked more than once. Attacks will inflict varying degrees of damage.

12) The game ends when either all of the enemy ships have been destroyed:

Glorious win Captain.
The enemy is defeated.

or your damage reaches 100%:

Your ship is dead!

Listing

```
10 'INIT': GOSUB 'CLRDSF' @ STD
20 INTEGER C,D,E0,E1,I,O,S,S1,T
30 REAL T9,D9
40 O=FLAG(-16,1)
50 INTEGER B(5),E(50)
60 DIM C0,K$(1),K1$(4),W
70 O=FLAG(-16,O)
80 S1=VAL(FNK0$("Skill level (0-9)?","0123456789"))
90 DISP "Setting up"
100 RANDOMIZE
110 C=IP(FNR(0,10000))
120 !
130 E1=FNR(31-30/(S1+1),31)
140 FOR I=1 TO E1
150 'ERPT': E(I)=IP(FNR(0,10000))
160 IF C=IP(E(I)) THEN 'ERPT'
170 NEXT I
180 !
190 FOR I=1 TO 5
200 'BRPT': B(I)=FNR(0,10000)
210 IF C=B(I) THEN 'BRPT'
220 NEXT I
230 !
240 E0=E1 @ T=20 @ S=100 @ W=0 @ D=0
250 !
260 RESTORE 'CHARGE' @ GOSUB 'TUNE'
270 'COMMANDS': K$=FNK0$("Cmnd: C,D,J,L,Q,S,T,?","CDJLQST?")
280 BEEP 4000,.04 @ BEEP 3000,.04 @ GOSUB 'CLRDSF'
290 IF K$="?" THEN GOSUB 'STS' ELSE GOSUB K$
300 CALL ENEMY(C,E1,E,S,D,S1)
310 IF D>=100 THEN 'DEAD'
320 IF E0=0 THEN 'VICTORY'
330 DISP @ BEEP 3000,.04 @ BEEP 4000,.04 @ GOTO 'COMMANDS'
340 'Q': PUT "#38"
350 END
360 !
370 'CHARGE': DATA 440,.05,554.37,.05,659.26,.05,880,.1,1,.1,659.26,.0
5,880,.9,0,0
380 'TAPS': DATA 440,.8,440,.8,440,.1,440,.7,523.25,.8,493.88,.2,493.8
8,.6
390 DATA 440,.2,440,.6,415.3,.2,440,.8,0,0
400 'TUNE': READ T9,D9 @ BEEP T9,D9 @ IF T9=0 THEN RETURN ELSE 'TUNE'
410 'CLRDSF': DISP CHR$(27)&"H"&CHR$(27)&"J"; @ RETURN
420 'D': CALL DOCK(C,B,T,S,D) @ RETURN
430 'DEAD': RESTORE 'TAPS' @ GOSUB 'TUNE'
440 DISP "Your ship is dead!" @ END
450 'VICTORY': RESTORE 'CHARGE' @ GOSUB 'TUNE' @ DISP "Glorious win Ca
ptain."
460 RESTORE 'CHARGE' @ GOSUB 'TUNE' @ DISP "The enemy is defeated." @
END
470 'L': GOSUB 'W' @ CALL LONGSCAN(C,E1,E,B) @ RETURN
480 'S': GOSUB 'W' @ CALL SHORTSCN(C,E1,E,B) @ RETURN
490 'T': CALL FIRETORP(C,C0,E1,E,E0,T) @ RETURN
500 'J': CALL JUMP(C,C0,E1,E,B,W)
```

```

510 'W': DISP "Quad ["&STR$(C DIV 1000)&","&STR$(RMD(C DIV 100,10))&"]
";
520 DISP "Sect ["&STR$(RMD(C DIV 10,10))&","&STR$(RMD(C,10))&"]"
530 RETURN
540 'C': INPUT "Course, speed: ",STR$(C0)&","&STR$(W);C0,W
550 IF W>10 OR W<0 THEN 'C' ELSE RETURN
560 'STS': GOSUB 'W'
570 I$="12A,':',3DZ" @ I1$=I$&","&'%'
580 DISP USING I$;"Torpedoes",T
590 DISP USING I1$;"Shields",S
600 DISP USING I$&".D";"Warp",W
610 DISP USING I$&".4D";"Heading",C0
620 DISP USING I1$;"Damage",D
630 DISP USING I$;"Enemy Ships",E0
640 RETURN
650 !
660 DEF FNR(L0,H0)=RND*(H0-L0)+L0
670 !
680 DEF FNK0$(D0$,K0$)
690 DISP D0$
700 'IDLE': K1$=KEY$
710 IF NOT POS(K0$,K1$) THEN 'IDLE'
720 FNK0$=K1$
730 END DEF
740 !
750 SUB LONGSCAN(C,E1,E(),B())
760 INTEGER F,I,X,Y,X1,Y1
770 DIM S$[10]
780 X=C DIV 1000 @ Y=RMD(C DIV 100,10)
790 S$="Enemy Ship"
800 FOR I=1 TO E1
810 IF E(I)=INF THEN 'NXT'
820 X1=E(I) DIV 1000 @ Y1=RMD(E(I) DIV 100,10)
830 IF ABS(X1-X)<=1 AND ABS(Y1-Y)<=1 THEN GOSUB 'CONTACT'
840 'NXT': NEXT I
850 S$="Star Base"
860 FOR I=1 TO 5
870 X1=B(I) DIV 1000 @ Y1=RMD(B(I) DIV 100,10)
880 IF ABS(X1-X)<=1 AND ABS(Y1-Y)<=1 THEN GOSUB 'CONTACT'
890 NEXT I
900 IF NOT F THEN BEEP 3000,.05 @ DISP "Clear"
910 END
920 !
930 'CONTACT': F=1 @ BEEP 3000,.05
940 DISP S$&" Quad. ["&STR$(X1)&","&STR$(Y1)&"]" @ RETURN
950 END SUB
960 !
970 SUB SHORTSCN(C,E1,E(),B())
980 INTEGER F,X1,Y1
990 DIM S$[10]
1000 S$="Enemy Ship"
1010 FOR I=1 TO E1
1020 IF E(I)=INF THEN 'NXT'
1030 X1=RMD(E(I),100) DIV 10 @ Y1=RMD(E(I),10)
1040 IF C DIV 100=E(I) DIV 100 THEN GOSUB 'CONTACT'
1050 'NXT': NEXT I
1060 S$="Star Base"

```

```

1070 FOR I=1 TO 5
1080 X1=RMD(B(I),100) DIV 10 @ Y1=RMD(B(I),10)
1090 IF C DIV 100=B(I) DIV 100 THEN GOSUB 'CONTACT'
1100 NEXT I
1110 IF NOT F THEN BEEP 3000,.05 @ DISP "Clear"
1120 END
1130 !
1140 'CONTACT': F=1 @ BEEP 3000,.05
1150 DISP S$&" Sect. ["&STR$(X1)&","&STR$(Y1)&"]" @ RETURN
1160 END SUB
1170 !
1180 SUB JUMP(C,C0,E1,E(),B(),W)
1190 INTEGER X0,Y0,X1,Y1,I
1200 Y=W*SIN(C0) @ X=W*COS(C0)
1210 Y0=IP(Y)+RMD(C DIV 100,10) @ X0=IP(X)+C DIV 1000
1220 Y1=FP(Y)*10+RMD(C,10) @ X1=FP(X)*10+RMD(C DIV 10,10)
1230 IF X1>=10 THEN X0=X0+X1 DIV 10 @ X1=MOD(X1,10)
1240 IF X1<0 THEN X0=X0-1 @ X1=MOD(X1,10)
1250 IF Y1>=10 THEN Y0=Y0+Y1 DIV 10 @ Y1=MOD(Y1,10)
1260 IF Y1<0 THEN Y0=Y0-1 @ Y1=MOD(Y1,10)
1270 IF X0>9 OR X0<0 OR Y0>9 OR Y0<0 THEN 'BLORPED'
1280 C1=X0*1000+Y0*100+X1*10+Y1
1290 FOR I=1 TO E1
1300 IF C1=IP(E(I)) THEN 'BLORPED'
1310 NEXT I
1320 FOR I=1 TO 5
1330 IF C1=B(I) THEN 'BLORPED'
1340 NEXT I
1350 C=C1
1360 END
1370 'BLORPED': BEEP 100
1380 DISP "Bad Jump!"
1390 END SUB
1400 !
1410 SUB FIRETORP(C,C0,E1,E(),E0,T)
1420 IF T=0 THEN END ELSE T=T-1
1430 X=RMD(C DIV 10,10) @ Y=RMD(C,10)
1440 FOR I=1 TO E1
1450 IF E(I)=INF OR C DIV 100#E(I) DIV 100 THEN 'NXT'
1460 X1=RMD(E(I) DIV 10,10) @ Y1=RMD(IP(E(I)),10)
1470 IF ANGLE(X1-X,Y1-Y)=C0 THEN 'GOTHIM'
1480 'NXT': NEXT I
1490 BEEP 100 @ DISP "Torpedo missed"
1500 END
1510 !
1520 'GOTHIM': BEEP 3000,.05 @ BEEP 3000,.05 @ E(I)=INF @ E0=E0-1
1530 DISP "Enemy Ship ["&STR$(X1)&","&STR$(Y1)&"]"&CHR$(13);
1540 FOR I=1 TO 16 @ BEEP 2000,.05 @ DISP "*"; @ NEXT I @ DISP
1550 END SUB
1560 !
1570 SUB DOCK(C,B(),T,S,D)
1580 FOR I=1 TO 5
1590 IF C DIV 100<>B(I) DIV 100 THEN 'NXT'
1600 IF ABS(RMD(C DIV 10,10)-RMD(B(I) DIV 10,10))>1 THEN 'NXT'
1610 IF ABS(RMD(C,10)-RMD(B(I),10))<=1 THEN 'OK'
1620 'NXT': NEXT I
1630 BEEP 100 @ DISP "Cannot Dock" @ END

```



```

1640 'OK': T=20 @ S=100 @ D=0 @ BEEP 3000,.04 @ BEEP 3000,.04 @ DISP "
Docked"
1650 END SUB
1660 !
1670 SUB ENEMY(C,E1,E(),S,D,S1)
1680 FOR I=1 TO E1
1690 IF E(I)=INF THEN 'NXT'
1700 IF E(I) DIV 100#C DIV 100 THEN 'NXT'
1710 IF RND<=S1/10 THEN GOSUB 'ATTACK'
1720 'NXT': NEXT I
1730 END
1740 'ATTACK': DISP "Attacked: "; @ R=RND
1750 IF RND>(S1+1)/10 THEN DISP "Missed" @ RETURN
1760 DISP "Sect. ["&STR$(RMD(E(I) DIV 10,10))&","&STR$(RMD(E(I),10))&
]"
1770 D=D+100*(S1*RND/10)/(S+(S=0)) @ IF D>100 THEN D=100
1780 S=S-100*S1*RND/10 @ S=S*(S>0) @ RETURN
1790 END SUB

```

NOTES

NOTES

NOTES

NOTES

NOTES

