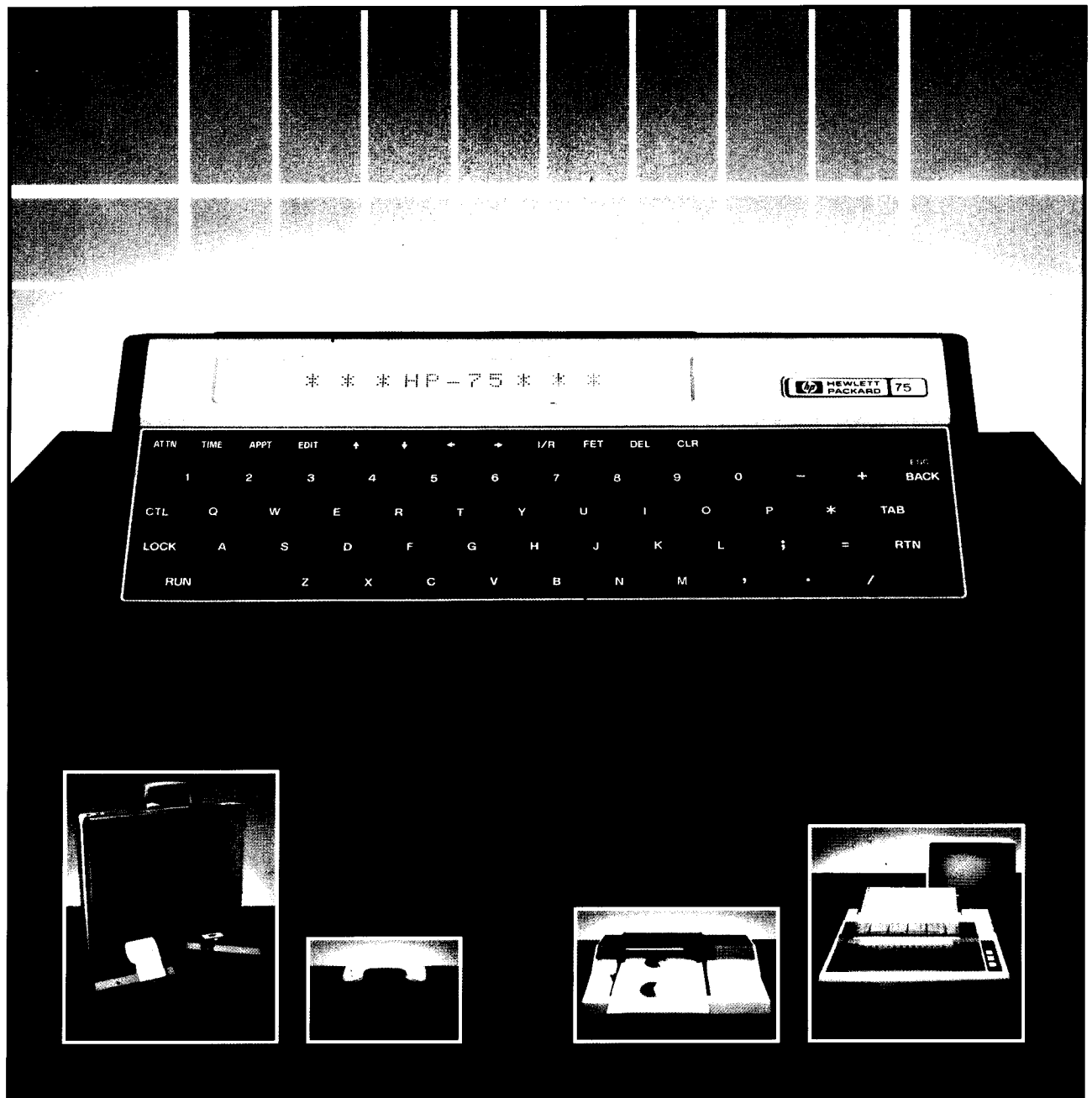


# HEWLETT-PACKARD

## HP-75

### USERS' LIBRARY SOLUTIONS

# Games II



## **NOTICE**

**The program material contained herein is supplied without representation or warranty of any kind. Hewlett-Packard Company therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.**

## TABLE OF CONTENTS

1. FOOTBALL . . . . . by Dennis Corliss 1  
Play the role of quarterback and defensive signal caller in a football game.
2. GOLF . . . . . by Dennis Corliss 17  
Play an 18-hole championship golf course.
3. HAMURABI . . . . . by Dennis Corliss 27  
Play the role of a dictator governing the economy of a small country.
4. REVERSE . . . . . by Dennis Corliss 34  
Order a series of random numbers by reversing the order of subsets of the numbers.
5. SLOT MACHINE . . . . . by Dennis Corliss 39  
Play the slot machines in Las Vegas! You may choose machines that accept everything from nickels to dollars.
6. BREAKOUT . . . . . by Ed Groth 45  
Use a ball and paddle to knock bricks out of a wall. Requires the HP 62163A Video Interface.

# PROGRAM DESCRIPTION

## FOOTBALL

This is pro-style football. The player assumes the role of quarterback and defensive signal-caller for the team. The opponent is the HP-75. There are 14 available offensive plays (8 runs, 6 passes) and 4 defensive alignments, along with field goal, punt, and quick kick capabilities.

There is a two-minute warning before the end of each half. Five to seven plays are left in the half at that point. At halftime, the player may view cumulative statistics for their team and the opponent.

Call all plays by their corresponding number:

OFFENSE	Runs:	1 = Dive	2 = Off Tackle
		3 = Scissors	4 = Trap
		5 = Sweep	6 = Option
		7 = Reverse	8 = Draw
	Passes:	9 = Sideline	10 = Look-in
		11 = Rollout	12 = Screen
		13 = 'Fly'	14 = 'Post'
		Kicks:	15 = Field Goal
	17 = Quick Kick		
	DEFENSE	1 = 'Pass Rush' (4-3)	2 = 'Okie' (5-2)
		3 = 'Short-Ydg'	4 = 'Prevent'

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		FOOTBALL	
1	Option to view play codes If 'Y' then display codes	Do you need a list of plays?N	[RTN] or Y [RTN]
2	Enter team name (<=10 chars.)	Name of your team?	aaaa [RTN]
3	Opponent	Your opponent is the HP-75	
4	Coin toss. If HP-75 wins: If player wins toss:	I won the toss and will receive aaaa won the toss	
	Option to kick or receive ball:	Kick of Receive?	K or R [RTN]
5	Kickoff	Get ready for the kickoff!	
5a		nn yard kickoff! nn yard runback	
6	Ball is on HP-75 or aaaa yard line	Ball on <span style="border: 1px solid black; padding: 2px;">HP-75</span> <span style="border: 1px solid black; padding: 2px;">aaaa</span> nn yard line	
7	If player is offense: Enter offensive play code: Goto step 8	X down aaaa X to go Your play (1-17)	1-17 [RTN]
7a	If HP-75 is offense: Enter defensive play code:	X down HP-75 X to go Enter defense (1-4):	1-4 [RTN]
8	Display plays chosen:	aaaa = Play HP-75 = Play	
9	Display play results: Display current yard line, possession, down: Goto step 7	play results Ball on <span style="border: 1px solid black; padding: 2px;">HP-75</span> <span style="border: 1px solid black; padding: 2px;">aaaa</span> nn yard line 1st down nn to go	
10	Touchdowns. Add 6 points to	Touchdown <span style="border: 1px solid black; padding: 2px;">HP-75</span> <span style="border: 1px solid black; padding: 2px;">aaaa</span> !	

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	appropriate score, attempt	Kicking extra point	
	extra point (if good, add to		
	score) and goto step 5.		
11	Safety. Add 2 points to		
	appropriate score.		
	Kickoff from 20 yard line and		
	goto step 5a.	Kickoff from <span style="border: 1px solid black; padding: 2px;">HP-75 aaaa</span> 20 yd line	
12	Penalties. If holding:	HOLDING	
	If offside:	OFFSIDE	
		Penalty on <span style="border: 1px solid black; padding: 2px;">HP-75 aaaa</span>	
12a	If penalty on aaaa:	Penalty accepted: nn yards	
	If penalty on HP-75:	Accept the penalty (Y/N)	Y or N [RTN]
	If 'Y' then step 12a		
13	At halftime and end of game,	<span style="border: 1px solid black; padding: 2px;">*** HALFTIME *** *** END OF GAME ***</span>	
	give option to view statistics:	View statistics?	Y or N [RTN]
	If 'Y' then:	Statistics for aaaa	
		⋮	
		Statistics for HP-75	
		⋮	
14	After end-of-game statistics,		
	give option to play another		
	game:	Another game?Y	[RTN] or N [RTN]
	If 'Y' then step 4 else stop		

# VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
B( )	Statistics	L( ),G( )	Storage for L, G
S8	Indicates safety	U	Constant for Y
R2	Indicates extra point	M	Fudge factor for Y
P8	Penalty yards	D9	Defensive play
Z9	Flag to show game over	I	Index and looping
D	Counts downs	W	Same as Q
T	Timer	Q	Indicates team in possession
S1	Player's score	X	Yard line
S2	HP-75's score	Y	Yards gained, lost
P1	Previous play	R	Yards runback
P	Offensive play	P9	Extra points
S	Yards gained before 1st down	T5	Indicates possession in 2nd half
G1	Indicates = 10 yards to goal	E\$	Offensive plays
T1,Z	Indicates second half	D\$	Downs
H9,E9	Time constants	F\$	Statistics
F1,F2	Figure distance to goal	R\$	Players team name
L,G	Help determine yards gained or lost for a play	Q\$	User Interaction
E2\$	Defensive plays		

# NOTES AND REFERENCES

- Notes:
1. There is a slight pause for initialization before the program begins.
  2. To omit play list from program: Omit step 1 from User Instructions. Delete lines 140-280. Delete lines 5020-5190.

Reference: "FOOTBALL", HP-85 Basic, Hewlett-Packard.



# PROGRAM LISTING

```

10 ! FOOTBALL
20 !
30 ! Revision 11/01/82
40 !
50 DELAY 3
60 RANDOMIZE
70 INTEGER B(15),S8,R2,P8,Z9,D,T,S1,S2
   ,P1,P,S,G1,T1,Z
80 INTEGER H9,E9,R,F1,L,G,D9,I,W
90 INTEGER L(15),G(15)
100 DIM E#[187],D#[12],F#[55],R#[10],Q#
   [4],E2#[52]
110 DEF FNF(X) = SIN(PI*RND)

120 DISP TAB(12);'FOOTBALL'
130 S8,R2,P8,Z9,D,T,S1,S2,P1,P,S,G1,T1,
   Z=0
150 FOR I=1 TO 15
160 B(I)=0
170 NEXT I
180 RESTORE
190 FOR I=1 TO 5
200 READ F#[11*I-10,11*I]
210 NEXT I
220 DATA POINTS,FIRST DOWNS,YDS-RUSH,YD
   S-PASS,YDS-PENALTY
240 ON ERROR GOTO 250
250 INPUT 'Do you need a list of plays?
   ','N';Q#
260 Q#=UPRC$(Q#[1,1]) @ IF Q##'Y' AND Q
   ##'N' THEN 250
270 OFF ERROR
280 IF Q#='Y' THEN GOSUB 5020
290 ON ERROR GOTO 300
300 INPUT 'Name of your team? ',CHR$(95)
   ;R#
310 IF R#=CHR$(95) THEN 300
320 IF LEN(R#)>10 THEN R#=UPRC$(R#[1,10
   ]) ELSE R#=UPRC$(R#)
330 OFF ERROR
340 DISP ' Your opponent is the HP-75
   ,
350 DISP
360 FOR I=1 TO 17
370 READ E#[11*I-10,11*I]
380 NEXT I
390 DATA DIVE,OFF TACKLE,SCISSORS,TRAP
400 DATA SWEEP,OPTION,REVERSE,DRAW,SIDE
   LINE
410 DATA LOOK-IN,ROLLOUT,SCREEN,FLY,POS
   T
420 DATA FIELD GOAL,PUNT,QUICK KICK
430 FOR I=1 TO 4
440 READ E2#[I*13-12,I*13]

```

-Function for generating random number

# PROGRAM LISTING

```

450 NEXT I
460 DATA PR,OKIE,SHORT YARDAGE,PREVENT
470 FOR I=1 TO 4
480 READ D#[I*3-2,I*3]
490 NEXT I
500 DATA '1st','2nd','3rd','4th'
510 FOR I=1 TO 14
520 READ L(I),G(I)
530 NEXT I
540 DATA 2,7,3,10,5,15,5,20,3,10,7,15,9
      ,25
550 DATA 5,15,5,10,5,10,7,15,10,20,10,5
      0,10,40
560 H9=25
570 E9=50
580 T5,Q=1-2*IP(RND*2)
590 IF Q>0 THEN 620
600 DISP 'I won the toss and will receive.'
610 GOTO 690
620 DISP R#;' won the toss.'
630 ON ERROR GOTO 640
640 DISP CHR$(203);'ick or ';CHR$(210);
650 INPUT 'eceive?',CHR$(95);Q#
660 Q#=UPRC$(Q#[1,1])
670 OFF ERROR
680 ON POS('KR',Q#)+1 GOTO 630,710,690
690 Q=-Q
700 IF Z9>0 THEN 4920
710 X=50-10*Q
720 DISP 'Get ready for the kickoff !!'
730 FOR N=1 TO 20
740 Y=30+IP(40*FNF(1))
750 BEEP @ BEEP @ BEEP
760 DISP Y;'yard kickoff!'
770 X=X+Q*Y
780 IF Q=1 THEN 810
790 IF X>0 THEN 900
800 GOTO 820
810 IF X<100 THEN 900
820 DISP TAB(6);'* * * TOUCHBACK * * *'

830 D=0
840 IF P#15 THEN 870
850 X=X-Q*Y
860 IF ABS(50-X)<30 THEN 880
870 X=50+30*Q
880 Q=-Q
890 GOTO 950
900 IF P>14 THEN 920
910 R=IP(40*FNF(1))
920 Q=-Q
930 DISP R;'yard runback.'

```

-Yardage loss/gain data for  
offensive plays

-Touchback- ball will be  
brought out to the 20 yard  
line

# PROGRAM LISTING

```

940 X=X+Q*R
950 R,P1=0
960 GOSUB 980
970 GOTO 1030
980 IF X>50 THEN 1010
990 DISP 'Ball on ',R#;X;'yd line'
1000 GOTO 1020
1010 DISP 'Ball on HP-75';ABS(X-100);'yd
    line'
1020 RETURN
1030 D=D+1
1040 GOTO 1090
1050 F1=ABS(X-(Q+1)/2*100)

1060 RETURN
1070 F2=ABS(X-(Q-1)/2*100)
1080 RETURN
1090 IF D#1 THEN 1200
1100 IF P#0 THEN 1120
1110 S=0
1120 GOSUB 1050
1130 IF F1<=10 THEN 1190
1140 IF Q=-1 THEN 1170
1150 DISP '1st down ',R#;' ';ABS(10-S);'
    to go'
1160 GOTO 1230
1170 DISP '1st down HP-75 ';ABS(10-S);'t
    o go'
1180 GOTO 1230
1190 G1=1
1200 DISP D#ID*3-2,D*31;' down '
1210 IF Q=-1 THEN DISP 'HP-75 ';ELSE DIS
    P R#;' ';
1220 IF G1<=0 THEN DISP ABS(10-S);'to go
    ' ELSE DISP ' goal to go'
1230 P#0
1240 P1=P @ P=0
1250 GOSUB 4440
1260 IF Z>0 THEN 4340
1280 IF Q>0 THEN 1640
1290 GOTO 2260
1300 IF D>1 THEN 1330

1310 P=1+IP(3.5*RND)*4
1320 GOTO 1720
1330 IF D>2 THEN 1420
1340 IF ABS(10-S)>3 THEN 1370
1350 IF P1>12 THEN 1310
1355 P=1+IP(5*RND)
1360 GOTO 1720
1370 IF P>12 THEN 1310
1380 IF P1>8 THEN 1410
1390 P=P+1
1400 GOTO 1720

```

-Aid in determining distances  
to goals

-Begin analyzing defense  
(player) to choose offense

# PROGRAM LISTING

```

1410 ON IP(2*RND)+1 GOTO 1310,1390
1420 IF D>3 THEN 1540
1430 IF ABS(10-S)>3 THEN 1460
1440 P=1+4*IP(2*RND)+IP(2*RND)
1450 GOTO 1720
1460 IF ABS(10-S)>6 THEN 1480
1470 GOTO 1390
1480 IF ABS(10-S)>11 THEN 1510
1490 P=P+1
1500 GOTO 1390
1510 IF ABS(10-S)<25 THEN 1350
1520 P=17
1530 GOTO 1720
1540 GOSUB 1050
1550 IF F1>40 THEN 1620
1560 IF ABS(10-S)>3 THEN 1580
1570 GOTO 1440
1580 GOSUB 1050
1590 IF F1>25 THEN 1620
1600 P=15
1610 GOTO 1720
1620 P=16
1630 GOTO 1720
1640 ON ERROR GOTO 1650
1650 INPUT 'Your play (1-17):',CHR$(95);
P
1660 IF P>17 THEN 1700
1670 IF P<1 THEN 1700
1680 OFF ERROR
1690 GOTO 1720
1700 DISP 'Try again.'
1710 GOTO 1640
1720 IF Q=1 THEN DISP R#;': ';ELSE DISP
'HP-75: ';
1730 DISP E#[P*11-10,P*11]
1740 IF P<15 THEN 1760
1750 ON P-14 GOTO 1800,2040,2100
1760 L=L(P)
1770 G=G(P)
1780 IF Q<0 THEN 2325
1790 GOTO 2160
1800 ! FIELD GOAL ROUTINE.
1810 Y=IP(FNF(1)*50)
1820 GOSUB 1050
1830 IF Y>.6*F1 THEN 1860
1840 BEEP @ DISP ' Field goal is block
ed !!!'
1850 GOTO 1930
1860 X=X+Q*Y
1870 IF Y<F1+10 THEN 1920
1880 IF .05+10/F1<RND THEN 1920
1890 BEEP @ DISP TAB(6);'Field goal is g
ood !!!'
1900 P9=3

```

-Player is offense - must enter  
play number

# PROGRAM LISTING

```

1910 GOTO 4200
1920 BEEP @ DISP TAB(7);'Field goal fail
    ed.'
1930 D,S=0
1940 GOSUB 1050
1950 IF Y>F1 THEN 820
1960 R=IP(FNF(1)*Y)
1970 GOTO 2150
1980 IF RND>.2 THEN 2020
1990 DISP TAB(7);'Extra point failed.'
2000 P9=0
2010 GOTO 4340
2020 DISP TAB(6);'Extra point is good !!
    !'
2030 GOTO 4190
2040 !
2050 Y=20+IP(30*FNF(1))
2060 DISP Y;'yard punt.'
2070 X=X+Q*Y
2080 R=IP(35*FNF(1))
2090 GOTO 2150
2100 !
2110 Y=30+IP(20*FNF(1))
2120 DISP Y;'yard kick.'
2130 X=X+Q*Y
2140 R=IP(10*FNF(1))
2150 GOTO 3040
2160 IF Q<0 THEN 2260
2170 GOSUB 1050
2180 IF F1>20 THEN 2210
2190 D9=3
2200 GOTO 2300
2210 IF (D=2 OR D=3) AND 10-S>5 THEN D9=
    4 ELSE D9=1+IP(2*RND)
2230 GOTO 2300
2260 ON ERROR GOTO 2270
2270 INPUT 'Enter defense (1-4):',CHR$(9
    5);D9
2280 IF D9>4 OR D9<1 THEN 2270
2290 OFF ERROR
2300 IF Q=-1 THEN DISP R#;' : ';ELSE DISP
    'HP-75: ';
2320 DISP E2#(D9*13-12,D9*13)
2325 IF P#0 THEN 2340
2330 IF Q=-1 THEN 1300
2340 IF P>8 THEN 2470
2350 IF P>4 THEN 2370
2360 ON P GOTO 2470,2380,2400,2400
2370 ON P-4 GOTO 2380,2420,2420,2440
2380 IF P1#1 THEN 2470
2390 GOTO 2450
2400 IF P1#2 THEN 2470
2410 GOTO 2450
2420 IF P1#5 THEN 2470

```

-Punts

-Kickoffs

-Choose defense for 75. If 2nd  
or 3d w/>5 to go PREVENT

-Player is defense

# PROGRAM LISTING

```

2430 GOTO 2450
2440 IF P1#9 THEN 2470
2450 U=1
2460 GOTO 2480
2470 U=1.25
2480 IF D9#1 THEN 2510
2485 IF D9#1 THEN 2510
2490 IF P>5 THEN 2630
2500 ON P GOTO 2650,2630,2610,2630,2650
2510 IF D9#2 THEN 2540
2520 IF P>5 THEN 2630
2530 ON P GOTO 2630,2630,2610,2630,2650
2540 IF D9#3 THEN 2580
2550 IF P>5 THEN 2570
2560 ON P GOTO 2650,2650,2630,2630,2650
2570 IF P#9 THEN 2630
2580 IF P<13 THEN 2600
2590 GOTO 2650
2600 IF P#8 THEN 2630
2610 M=1.25
2620 GOTO 2660
2630 M=1
2640 GOTO 2660
2650 M=.8
2660 Y=IP((G*RND-L*RND)*(U*M))
2670 IF R2#0 THEN 4130
2680 IF RND<.98 THEN 2720
2690 GOSUB 1050
2700 Y=F1+1
2710 GOTO 3630
2720 IF P>8 THEN 2900
2730 IF RND<.93 THEN 3070

2740 Y=IP(.5*FNF(1)*Y)
2750 DISP 'Fumble after';
2760 IF Y<0 THEN 2790
2770 DISP Y;'yard gain.'
2780 GOTO 2830
2790 IF Y#0 THEN 2820
2800 DISP ' no gain.'
2810 GOTO 2830
2820 DISP ABS(Y);'yard loss.'
2830 IF 1-IP(2*RND)=0 THEN 2860
2840 DISP '      Fumble recovered !'
2850 GOTO 3070
2860 DISP '      Fumble lost !'
2870 X=X+Q*Y
2880 Q=-Q
2890 GOTO 4280
2900 IF RND<.07 THEN 3010
2910 IF Y>0 THEN 2940
2920 DISP 'QB tackled for';
2930 GOTO 3630
2940 IF ABS(RND-.5)<.23 THEN 3630

```

-Determine yardage

-Determine chances of fumble  
for running plays

-Determine outcome of fumble

# PROGRAM LISTING

```

2950 IF 1P(RND*2)=0 THEN 2980
2960 DISP '          Batted down.'
2970 GOTO 2990
2980 DISP '          Incomplete.'
2990 Y=0
3000 GOTO 3070
3010 DISP '          * * * INTERCEPTION * * *'
3020 X=X+Q*Y
3030 R=IP(15*FNF(1))
3040 P=18
3050 D,S,G1=0
3060 GOTO 3630
3070 IF ABS(RND-.5)>.04 THEN 3630
3080 P8=1
3090 GOTO 3750
3100 IF 1P(2*RND)=0 THEN 3170
3110 W=-1
3120 IF 1P(2*RND)=0 THEN 3150
3130 P8=15
3140 GOTO 3190
3150 P8=5
3160 GOTO 3210
3170 W=1
3180 GOTO 3120
3190 DISP '          HOLDING'
3200 GOTO 3220
3210 DISP '          OFFSIDE'
3220 DISP 'Penalty on ';
3230 IF W=1 THEN DISP R#;'.' ELSE DISP '
      HF-75.'
3240 IF W<0 THEN 3400
3250 IF Q<0 THEN 3350
3260 IF P8=15 THEN 3280
3270 IF Y<0 THEN 3370
3280 GOSUB 3470
3290 Y=P8*W
3300 DISP 'Penalty accepted: ';P8;'yards
      '
3310 D=D-1
3320 X=X+Q*Y
3330 B(14-W)=B(14-W)+P8
3340 GOTO 4240
3350 IF S+Y>ABS(10-S) THEN 3370
3360 GOTO 3280
3370 P8=0
3380 GOTO 3630
3390 IF W>0 THEN 3260
3400 ON ERROR GOTO 3410
3410 INPUT 'Accept the penalty (Y/N)',CH
      R#(95);Q#
3420 Q#=UPRC$(Q#[1,1])
3430 IF Q#='N' THEN 3370
3440 IF Q#='Y' THEN 3410
3450 OFF ERROR

```

-Passing errors

-Penalties

	PROGRAM LISTING	
--	-----------------	--

```

3460 GOTO 3280
3470 IF Q>0 THEN 3570
3480 IF W>0 THEN 3530
3490 GOSUB 1070
3500 IF 2*P8<F2 THEN 3620
3510 P8=IP(F2/2)
3520 GOTO 3620
3530 GOSUB 1050
3540 IF 2*P8<F1 THEN 3620
3550 P8=IP(F1/2)
3560 GOTO 3620
3570 IF W>0 THEN 3600
3580 W=1
3590 GOTO 3530
3600 W=-1
3610 GOTO 3490
3620 RETURN
3630 IF P>14 THEN 3720
3640 X=X+Q*Y
3650 GOSUB 4650
3660 IF Q>0 THEN 3700
3670 IF X<1 THEN 3930
3680 IF X>99 THEN 3850
3690 GOTO 3720
3700 IF X>99 THEN 3930
3710 IF X<1 THEN 3930
3720 IF P>14 THEN 4310
3730 IF R=0 THEN 3750
3740 GOTO 930
3750 IF Y<=0 THEN 3790
3770 DISP Y; 'yard gain.'
3780 GOTO 3830
3790 IF Y#0 THEN 3820
3800 DISP ' No gain.'
3810 GOTO 3830
3820 DISP ABS(Y); 'yard loss.'
3830 IF P8#0 THEN 3100
3840 GOTO 4240
3850 DISP 'Safety vs ';
3860 IF Q=1 THEN DISP R#; '!' ELSE DISP '
HP-75.'
3870 P9=2
3880 Q=-Q
3890 S8=1
3900 GOSUB 4590
3910 Q=-Q
3920 GOTO 4340
3930 DISP ' Touchdown ';
3940 IF Q=1 THEN DISP R#; '!' ELSE DISP '
HP-75.'
3950 P9=6
3960 GOSUB 4590
3970 IF Q>0 THEN 4030
3980 IF S2+1=S1 THEN 1980

```

-Safety



# PROGRAM LISTING

```

3990 IF S2+2=S1 THEN 4010
4000 GOTO 1980
4010 P=4
4020 GOTO 4100
4030 DISP '      Kicking extra point.'
4040 GOTO 1980
4050 ON ERROR GOTO 4060
4060 INPUT 'Enter play (1-14):',CHR$(95)
      ;P
4070 IF P>14 OR P<1 THEN 4060
4080 OFF ERROR
4090 IF P>12 THEN 4220
4100 R2=1
4110 D9=2
4120 GOTO 2340
4130 R2=0
4140 IF Y<4.1 THEN 4220
4150 P9=2
4160 DISP '      Extra point is good !!'
4170 GOSUB 4590
4180 GOTO 4340
4190 P9=1
4200 GOSUB 4590
4210 GOTO 4340
4220 DISP '      Extra point attempt failed.'
      ,
4230 GOTO 4340
4240 S=S+Y
4250 IF S>9 THEN 4280
4260 IF D<4 THEN 4310
4270 Q=-Q
4280 D,S,G1=0
4290 IF P>14 THEN 4310
4300 B(S+Q)=B(S+Q)+1
4310 IF P=18 THEN 780
4320 GOSUB 980
4330 GOTO 1030
4340 DISP R$;':';S1;'      HP-75:';S2
4350 D,S,Z,P1,G1,P,P9=0
4360 IF S8#0 THEN 4390
4370 Q=-Q
4380 GOTO 690
4390 X=50-30*Q
4400 DISP 'Kickoff from';
4410 IF Q=1 THEN DISP R$;' 20 yard' ELSE
      DISP 'HP-75 20 yard'
4420 S8=0
4430 GOTO 740
4440 T=T+1
4450 IF T=H9-5 THEN 4540
4460 IF T=E9-5 THEN 4540
4470 IF T>H9 THEN 4490
4480 GOTO 4580
4490 IF T1>0 THEN 4560

```

-Scoreboard

# PROGRAM LISTING

```

4500 IF RND<.35 THEN 4580
4510 GOSUB 4700
4520 Q=T5
4530 GOTO 4340
4540 BEEP @ DISP '> > > TWO-MINUTE WARNI
      NG < < <'
4550 GOTO 4580
4560 IF T<E9 THEN 4580
4570 IF RND>.5 THEN 4720
4580 RETURN
4590 B(2+Q)=B(2+Q)+P9
4600 IF Q>0 THEN 4630
4610 S2=S2+P9
4620 GOTO 4640
4630 S1=S1+P9
4640 RETURN
4650 IF P>8 THEN 4680
4660 B(8+Q)=B(8+Q)+Y
4670 GOTO 4690
4680 B(11+Q)=B(11+Q)+Y
4690 RETURN
4700 DISP '          * * * HALFTIME * * *'
4710 GOTO 4740
4720 DISP '          * * * END OF GAME * * *'
4730 Z9=1
4740 ON ERROR GOTO 4750
4750 INPUT 'View statistics?'; Q$ @ Q$=U
      PRC$(Q$[1,1])
4760 OFF ERROR
4770 IF Q$='Y' THEN 4800
4780 IF Q$='N' THEN 4750
4790 GOTO 4890
4800 DISP 'STATISTICS FOR ';R$
4810 IMAGE 11A,3X,4D
4820 FOR I=0 TO 4
4830 DISP USING 4810 ; F$(I+1)*11-10,(I
      +1)*111,B(3+I*3)
4840 NEXT I
4850 DISP 'STATISTICS FOR HP-75'
4860 FOR I=0 TO 4
4870 DISP USING 4810 ; F$(I+1)*11-10,(I
      +1)*111,B(1+I*3)
4880 NEXT I
4890 Z,T1=1
4900 T=H9
4910 RETURN
4920 ON ERROR GOTO 4930
4930 INPUT 'Another game?','Y';Q$
4940 Q$=UPRC$(Q$[1,1])
4950 OFF ERROR
4960 IF Q$='N' THEN 5010
4970 IF Q$='Y' THEN 4920
4980 Z9,D,T,S1,S2,P1,P,S,G1,T1,Z=0
4995 FOR I=1 TO 15 @ B(1)=0 @ NEXT I

```

-Statistics review for halftime  
and end of game

# PROGRAM LISTING

```

5000 GOTO 560
5010 STOP
5020 !
5030 DISP 'Call plays as follows ...'
5040 DISP '      * * * RUNS * * *'
5050 DISP '1 = DIVE' @ DISP '2 = OFF TAC
      KLE'
5060 DISP '3 = SCISSORS' @ DISP '4 = TRA
      P'
5070 DISP '5 = SWEEP' @ DISP '6 = OPTION
      '
5080 DISP '7 = REVERSE' @ DISP '8 = DRAW
      '
5090 DISP '      * * * PASSES * * *'
5100 DISP '9 = SIDELINE' @ DISP '10 = LO
      OK-IN'
5110 DISP '11 = ROLLOUT' @ DISP '12 = SC
      REEN'
5120 DISP "13 = 'FLY'" @ DISP "14 = 'POS
      T'"
5130 DISP '      * * * KICKS * * *'
5140 DISP '15 = FIELD GOAL' @ DISP '16 =
      PUNT'
5150 DISP '17 = QUICK KICK'
5160 DISP 'Call defenses as follows ...'
5170 DISP "1 = 'PR' (4-3)" @ DISP "2 = '
      OKIE' (5-2)"
5180 DISP "3 = 'SHORT YARDAGE'" @ DISP "
      4 = 'PREVENT'"
5190 RETURN

```

-Key to plays

# PROGRAM DESCRIPTION

## GOLF

This game is a simulation of an 18-hole golf course. The player is supplied with 12 clubs (a driver, 2 woods, 8 irons, and putter) to tackle the course. In addition, he may choose to use a partial swing (i.e. a percent of his full swing) on any of the irons (see club list below).

Distances for each hole range from 180 yards to 560 yards, with pars of 3, 4 or 5 strokes. Par for the course is 72. Hazards include trees, sand-traps and water.

To begin the game the player must enter his handicap (0-30) and indicate his worst difficulty at golf: hook, slice, poor distance, trapshot, or putt. Then the distance to the hole, the par for that hole, and the conditions of the right and left sides of the fairway are displayed. The player then proceeds as on a golf course, selecting appropriate clubs for particular shots. If he hits a ball into the water or out of bounds he must hit again from the previous location and a penalty stroke is assessed. If the player selects a partial swing club, he must enter the percent of a full swing that he desires (1-99%).

When the player reaches the green, he must enter a putt potency number (>0). For example, a distance of 3 feet to the pin (hole) suggests a putt potency number of around 1.

CLUBS:	WOODS	IRONS (FULL)		IRONS (PARTIAL)	
	1) Driver	12) 2 Iron	16) 6 Iron	22) 2 Iron	26) 6 Iron
	2) 2 Wood	13) 3 Iron	17) 7 Iron	23) 3 Iron	27) 7 Iron
	3) 3 Wood	14) 4 Iron	18) 8 Iron	24) 4 Iron	28) 8 Iron
		15) 5 Iron	19) 9 Iron	25) 5 Iron	29) 9 Iron

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		WELCOME TO THE GOLF COURSE	
1	Enter handicap (0-30)	What is your handicap?	0-30 [RTN]
2	Enter worst difficulty at golf	Choose your worst difficulty ...	
		Hook, Slice, Dist., Trap or Putt?	H, S, D, T, or P [RTN]
3	Display conditions of hole e.g.:	You are on tee of hole 1	
		Distance 361 yards PAR 4	
		On right is adjacent fairway	
		On left is rough	
4	Enter club: woods 1-3,	What club do you want?	1-3, 12-19, or 22-29 [RTN]
	irons 12-19,		
	or if partial swing 22-29:	Percent of full swing (1-99)?	1-99 [RTN]
5	If ball landed on green		
	then step 6		
5a	Display flight and distance		
	of ball and new location e.g.:	You sliced	
		Shot went 228 yards ...	
		133 yards from the hole	
		Ball is 11 yards off line ...	
		... in adjacent fairway	
	Goto step 4		
6	Ball on green	On green # feet from pin	
7	Enter strength of putt	Putt potency number?	p [RTN]
	must be greater than 0		
7a	If ball went in hole then		
	step 8 else	Passed by cup	

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	Goto step 6		
8		You holed it!!!	
	Display number of strokes, and	Score on hole # was # #	
	total strokes so far:		
	IF PAR:	A Par. Nice going!	
	if PAR - 1:	A Birdie. Very good!	
	if PAR - 2:	An Eagle! Excellent!!!	
	if 1 stroke:	*** A HOLE IN ONE ***	
	if PAR + 1 or + 2:	Keep your head down	
9	To continue game goto step 3		
	for next hole and new condi-		
	tions. If hole 18 has been		
	played, then goto step 10.		
10	Display total score:	Total score for 18 holes was ##	
	if PAR:	Par game!	
	if <PAR:	## under par!!!	
	if >PAR:	## over par	
11	Option to play again:	Would you like to play again?Y	[RTN] or N [RTN]

## VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
L(1)	Location of ball	T	Difficulty at golf
L(2)	Terrain on right side of fairway	T5	Total par for course
L(3)	Terrain on left side of fairway	Z	Indicates use of subroutine
W	Percent of full swing on irons	H	Handicap
S1	Score on a particular hole	Q	Aid in determining dubbed shot
S2	Accumulated score	X	Index for terrain subroutine
F	Number of hole	J	Determines penalty stroke
D	Distance from tee to hole	K	Counts putts
P	Par for a hole	N	Random number used to find probability of getting out of trap
D2	Current distance to hole		
D1	Distance of shot	B( )	Temp. storage for D
O	Yards off line	I	Putt strength
C	Club number	T\$	Golf difficulty
		Q\$	User interaction

## NOTES AND REFERENCES

Notes: 1. To omit club list from program change:

580 if C >= 1 and C <= 29 then 600

630!

Delete 2220-2490

Reference: 1. "GOLF", HP-85, Hewlett-Packard.

# PROGRAM LISTING

```
10 ! GOLF
20 !
30 ! Revision 11/01/82
40 INTEGER L(11),J,K,Q,X,B,H,Z,S1,S2,F
   ,D,P,D2,D1,C,T,T5
50 REAL N,O,W,I
60 DIM T$(4),Q$(4)
70 RANDOMIZE
80 DELAY 2.5
90 DISP ' WELCOME TO THE GOLF COURSE.
   '

100 X=3 @ N=.8 @ F=1
110 S1,S2,T5=0
120 ON ERROR GOTO 130
130 INPUT 'What is your handicap?';H
140 IF H>30 OR H<0 THEN 130
150 ON ERROR GOTO 170
160 DISP 'Choose your worst difficulty.
   '
170 DISP CHR$(200);'ook ';CHR$(211);'li
   ce ';CHR$(196);'ist. ';CHR$(212);'r
   ap or ';CHR$(208);
180 INPUT 'utt?';CHR$(95);T$
190 T$=UPRC$(T$(1,1))
200 ON POS('HSDTP',T$)+1 GOTO 170,210,2
   20,230,240,250
210 T=1 @ GOTO 260
220 T=2 @ GOTO 260
230 T=3 @ GOTO 260
240 T=4 @ GOTO 260
250 T=5
260 OFF ERROR
270 !
280 J,Q,K,L(1)=0
290 S2=S2+S1
300 IF F=1 THEN 460
310 DISP 'Score on hole';F-1;'was';S1;T
   AB(29);S2
315 ON ERROR GOTO 330
320 ON P-S1+3 GOTO 450,450,350,370,390

330 DISP 'Keep your head down.'
340 GOTO 450
350 DISP 'A Par. Nice going!'
360 GOTO 450
370 DISP 'A Birdie. Very good !!'
380 GOTO 450
390 IF P=3 THEN 420
400 DISP 'An Eagle! Excellent!!!'
410 GOTO 450
420 DISP ' * * * A HOLE IN ONE * * *'
430 BEEP @ BEEP @ BEEP @ BEEP
450 OFF ERROR
455 IF F=19 THEN 1970
```

-PGA rules - handicap 0-30

-Determines comparative skill  
on hole

-Check for end of game



# PROGRAM LISTING

```

460 S1=0
470 IF S1=0 THEN 1800
480 IF L(1)<1 THEN 1330
490 X=1
500 IF L(1)>5 THEN 1370
510 DISP 'Shot went';D1;'yards...'
520 DISP D2;'yards from the hole.'
530 DISP 'Ball is';IP(0);'yards off lin
e...'
540 Z=1
550 GOSUB 2060
560 ON ERROR GOTO 570
570 INPUT 'What club do you want?';C
580 IF C<1 OR C>29 THEN GOSUB 2220 ELSE
600
590 GOTO 560
600 OFF ERROR
610 IF C>3 THEN 650
620 IF L(1)<5 OR C=14 OR C=23 THEN 680
630 GOSUB 2220
640 GOTO 560
650 IF C<12 THEN 630
660 C=C-6
670 GOTO 620
680 S1=S1+1
690 W=1
700 IF C>13 THEN 1010
710 IF F/3=IP(F/3) THEN 970
720 IF C<4 THEN 740
730 GOTO 750
740 IF L(1)=2 THEN 830
750 IF S1<7 THEN 860
760 D1=IP(((30-H)*2.5+187-((30-H)*.25+15
)*C/2+25* $\text{RND}$ ))
770 D1=IP(D1*W)
780 IF 1=2 THEN 1350
790 O= $\text{RND}/.8*(2*H+16)*\text{ABS}(\text{TAN}(D1*.0035))$ 
)
800 D2=IP( $\text{SQR}(O^2+\text{ABS}(D-D1)^2)$ )
810 IF D-D1<0 THEN 880
820 GOTO 900
830 DISP 'You dubbed it.'
840 D1=35
850 GOTO 790
860 IF D<200 THEN 1480
870 GOTO 760
880 IF D2<20 THEN 900
890 DISP 'Too much club...passed the ho
le.'
900 B=D
910 D=D2
920 IF D2>27 THEN 1150
930 IF D2>20 THEN 1280
940 IF D2>.5 THEN 1300

```

-Determine penalty shot

-Check for legal club selection

# PROGRAM LISTING

```

950 L(1)=9
960 GOTO 1690
970 IF (72+(H+1)/.85)/18(S2+Q+10*(F-1)/
18 THEN 720
980 Q=Q+1
990 IF S1/2#IP(S1/2) THEN 1100
1000 GOTO 830
1010 ON ERROR GOTO 1020
1020 INPUT 'Percent of full swing (1-99)
: ';W
1030 W=W/100
1040 IF W(<=0 OR W>=1) THEN 1020
1050 OFF ERROR
1060 IF L(1)=5 THEN 1470
1070 IF C=14 THEN 750
1080 C=C-10
1090 GOTO 750
1100 IF D<95 THEN 830
1110 DISP 'Ball hit tree...went into rou
gh'
1120 DISP D-75;'yards from hole.'
1130 D=D-75
1140 GOTO 570
1150 IF O>30 OR J>0 THEN 1330
1160 IF I>0 THEN 1220
1170 IF (S2+1)/15=IP((S2+1)/15) THEN 123
0
1180 DISP 'You hooked ';
1190 L(1)=L(3)
1200 IF O>45 THEN 1260 ELSE DISP
1210 GOTO 470
1220 IF (S2+1)/15=IP((S2+1)/15) THEN 118
0
1230 DISP 'You sliced ';
1240 L(1)=L(2)
1250 GOTO 1200
1260 DISP 'badly.'
1270 GOTO 470
1280 L(1)=5
1290 GOTO 470
1300 L(1)=8
1310 D2=IP(D2*3)
1320 GOTO 1560
1330 L(1)=1
1340 GOTO 470
1350 D1=IP(.85*D1)
1360 GOTO 790
1370 IF L(1)>6 THEN 1450
1380 DISP 'Your shot went into the water
.'
1390 S1=S1+1
1400 DISP 'Penalty stroke assessed.' @ D
ISP 'Hit from previous location.'
1410 J=J+1

```

# PROGRAM LISTING

```

1420 L(1)=1
1430 D=B
1440 GOTO 560
1450 DISP 'Your shot went out of bounds.
      '
1460 GOTO 1390
1470 !
1480 D2=1+3*IP(80/(40-H)*RND)
1490 GOTO 1560
1500 IF RND>N THEN 1540
1510 N=N*.2
1520 DISP 'Shot dubbed...still in trap.'
1530 GOTO 560
1540 N=.8
1550 GOTO 1480
1560 DISP 'On green';D2;'feet from pin.'
1570 ON ERROR GOTO 1580
1580 INPUT 'Putt potency number?';I
1590 OFF ERROR
1600 IF I<=0 THEN 1570
1610 S1=S1+1
1620 IF S1+1-P>H*.072+2 THEN 1690
1630 IF K>2 THEN 1690
1640 K=K+1
1650 IF T=4 THEN 1750
1660 D2=D2-I*(4+2*RND)+1.5
1670 IF D2<-2 THEN 1770
1680 IF D2>2 THEN 1720
1690 DISP 'You holed it!!!'
1700 F=F+1
1710 GOTO 280
1720 DISP 'Putt short.'
1730 D2=IP(D2)
1740 GOTO 1560
1750 D2=D2-I*(4+RND)+1
1760 GOTO 1670
1770 DISP 'Passed by cup.'
1780 D2=-D2
1790 GOTO 1730
1800 READ D,P,L(2),L(3)

1810 IF F-1>18 THEN 1970
1820 T5=T5+P
1830 DISP 'You are on tee of hole';F
1840 DISP 'Distance';D;'yards  PAR';P
1850 DISP 'On right is ';
1860 Z=0
1870 X=2
1880 GOSUB 2060
1890 DISP 'On left is ';
1900 X=3
1910 GOSUB 2060
1920 DISP
1930 GOTO 560

```

-Read in distance for hole, the  
par and fairway condition

# PROGRAM LISTING

```
1940 DATA 361,4,4,2,389,4,3,3,206,3,4,2,
      500,5,7,2,408,4,2,4,359,4,6,4
1950 DATA 424,4,4,2,388,4,4,4,196,3,7,2,
      400,4,7,2,560,5,7,2,132,3,2,2
1960 DATA 357,4,4,4,294,4,2,4,475,5,2,3,
      375,4,4,2,180,3,6,2,550,5,6,6
1970 DISP 'Total score for';F-1;'holes w
      as';S2
1980 IF S2-T5 THEN 2010
1990 DISP 'Par game !'
2000 GOTO 2500
2010 IF S2-T5>0 THEN 2040
2020 DISP -(S2-T5);'under par!!!'
2030 GOTO 2500
2040 DISP S2-T5;'over par.'
2050 GOTO 2500
2060 IF Z=1 AND L(X)#7 THEN DISP '    ...
      in ';
2070 ON L(X) GOTO 2100,2120,2140,2160,21
      80,2200,2080
2080 DISP 'out of bounds.'
2090 RETURN
2100 DISP 'fairway.'
2110 RETURN
2120 DISP 'rough.'
2130 RETURN
2140 DISP 'trees.'
2150 RETURN
2160 DISP 'adjacent fairway.'
2170 RETURN
2180 DISP 'trap.'
2190 RETURN
2200 DISP 'water.'
2210 RETURN
2220 ON ERROR GOTO 2230

2230 INPUT 'Need a list of clubs?','Y';Q
      #
2240 Q#=UPRC$(Q#[1,1])
2250 IF Q##'Y' THEN 2490
2260 DISP '    WOODS *** Full swing only'
2270 DISP '    1) DRIVER'
2280 DISP '    2) 2 WOOD'
2290 DISP '    3) 3 WOOD'
2300 DISP '    IRONS *** Full swing only'
2310 DISP '    12) 2 IRON'
2320 DISP '    13) 3 IRON'
2330 DISP '    14) 4 IRON'
2340 DISP '    15) 5 IRON'
2350 DISP '    16) 6 IRON'
2360 DISP '    17) 7 IRON'
2370 DISP '    18) 8 IRON'
2380 DISP '    19) 9 IRON'
2390 DISP '    IRONS *** Partial swing only
```

-Data for course

-Check for use of preposition  
in subroutine

-Subroutine to display club  
choices

# PROGRAM LISTING

```
2400 DISP ' 22) 2 iron'  
2410 DISP ' 23) 3 iron'  
2420 DISP ' 24) 4 iron'  
2430 DISP ' 25) 5 iron'  
2440 DISP ' 26) 6 iron'  
2450 DISP ' 27) 7 iron'  
2460 DISP ' 28) 8 iron'  
2470 DISP ' 29) 9 iron'  
2480 OFF ERROR  
2490 RETURN  
2500 ON ERROR GOTO 2510  
2510 INPUT 'Would you like to play again  
?','Y';Q$  
2520 Q$=UPRC$(Q$[1,1])  
2530 IF Q$='Y' THEN 100  
2540 IF Q$# 'N' THEN 2510  
2550 OFF ERROR  
2560 DELAY 1  
2570 DISP TAB(9);'END OF GAME' @ DISP  
2580 STOP
```

# PROGRAM DESCRIPTION

## 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 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 all his land.

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

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		HAMURABI	
1	Report of economy	I beg to report that last year:	
		x people starved . . .	
		x people came to the city:	
1a	A plague has a probability of		
	1:10. If there was one:	The plague killed $\frac{1}{2}$ the people	
1b	Population	Population now xx	
1c	Harvest	We harvested xx bushels . . .	
		at xx bushels per acre	
1d	Rats	Rats destroyed xx bushels	
1e	Total food	xxx bushels in storehouses	
1f	Land	The city owns xxx acres . . .	
	Land will sell at . . .	Worth xx bushels/acre	
2	Decisions must be made	Hamurabi . . .	
		Buy how many acres?	xx [RTN]
	If insufficient bushels to		
	buy land then display stats		
		Sell how many acres?	xx [RTN]
	If insufficient acres to sell		
	then display stats		
	If all land sold goto step 5		
		How many bushels for food?	xx [RTN]
	If insufficient bushels in		
	store then display stats		
		How many acres to plant?	xx [RTN]

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	If insufficient acres to plant		
	then display statistics		
	If many people died then:	Things aren't looking too well!	
	and goto step 4 else step 1		
3	STATS = Review economic status	Think again ... you only have:	
		xx people ...	
		xxx acres and ...	
		xxx bushels in storehouses	
4	Option to continue governing	You have ruled for x years	
	under current conditions:	Do you want to continue?	Y [RTN] or N [RTN]
	If 'Y' then continue		
	If 'N' then step 5		
5	Options to begin again:	Start over?	Y [RTN] or N [RTN]
	If 'Y' then step 1		
	If 'N' then end	END OF GAME	



## VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A1	Population	B4	Total harvest
A2	Immigrants	C1	Acres
A3	Deaths	J	Temporary storage for comparison
B1	Bushels in storehouses	I	Input variable
B2	Bushels destroyed by rats	Z	Number of years
B3	Harvest per acre	C2	Value of an acre
		Q\$	User interaction

## NOTES AND REFERENCES

Reference: "HAMURABI", HP-2000 Basic, Hewlett-Packard

# PROGRAM LISTING

```

10 ! HAMURABI
20 !
30 ! Revision 11/01/82
40 !
50 INTEGER A1,A2,A3,B1,B2,B3,B4,C1,J,Z
60 DELAY 5
70 DISP TAB(12);'HAMURABI'
80 A1=100 @ A2=5 @ A3=0
90 B1=2800 @ B2=200 @ B3=3 @ B4=3000
100 C1=1000 @ J=1 @ Z=0
110 ! REPORT TO HAMURABI.
120 IF Z=1 THEN 140
130 IF A3>(A1+A3)*.45 THEN DISP "Things
    aren't looking too well!" @ GOSUB
    1010
140 BEEP 600,.15 @ BEEP 400,.1 @ BEEP 6
    00,.15
150 Z=Z+1
160 IF Z/5=IP(Z/5) THEN GOSUB 1010

170 DISP 'I beg to report that last yea
    r:'
180 DISP A3;'people starved and ...'
190 DISP A2;'people came to the city.'
200 IF J>0 THEN 230

210 A1=A1-IP(A1/2)
220 DISP 'The plague killed 1/2 the peo
    ple'
230 DISP 'Population now';A1;'.'
240 DISP 'We harvested';B4;'bushels...'
250 DISP 'at';B3;'bushels per acre.'
260 DISP 'Rats destroyed';B2;'bushels.'
270 DISP B1;'bushels in storehouses.'
280 DISP 'The city owns';C1;'acres...'
290 C2=17+IP(6*RND)

300 DISP 'worth';C2;'bushels/acre.'
310 BEEP 600
320 DISP 'Hamurabi ...'
330 ! BUY LAND?
340 ON ERROR GOTO 350
350 INPUT 'Buy how many acres?';CHR$(95
    );I
360 I=IP(ABS(I))
370 OFF ERROR
380 IF I=0 THEN 450
390 J=I*C2
400 IF J<=B1 THEN 430
410 GOSUB 940
420 GOTO 340
430 B1=B1-J
440 C1=C1+I
450 ! SELL LAND?

```

-Set up initial conditions

-If more than 45% of the people died, option to restart

-If 5 years have passed give option to quit or continue

-Plagues have a probability of 1 in 10

-Status report

-Acreage ranges in value from 17 to 22 bushels per acre

-Option to buy land

# PROGRAM LISTING

```

460 ON ERROR GOTO 470
470 INPUT 'Sell how many acres?',CHR$(9
      5);I
480 I=IP(ABS(I))
490 OFF ERROR
500 IF I=0 THEN 570
510 IF I<C1 THEN 550
520 IF I=C1 THEN 1090

530 GOSUB 940
540 GOTO 460
550 C1=C1-I
560 B1=B1+C2*I
570 ! BUY FOOD?
580 ON ERROR GOTO 590
590 INPUT 'How many bushels for food?',
      CHR$(95);I
600 I=IP(ABS(I))
610 OFF ERROR
620 IF I<=B1 THEN 650
630 GOSUB 940
640 GOTO 580
650 B1=B1-I
660 A3=A1-IP(I/20)
670 A2=0
680 IF A3>=0 THEN 710
690 A2=-A3/2
700 A3=0
710 ! PLANT ACRES?
720 ON ERROR GOTO 730
730 INPUT 'How many acres to plant?',CH
      R$(95);I
740 I=IP(ABS(I))
750 OFF ERROR
760 IF I>C1 THEN 790
770 J=IP(I/2)
780 IF J<=B1 THEN 810
790 GOSUB 940
800 GOTO 720

810 IF I>10*A1 THEN 790
820 B1=B1-J
830 ! HARVEST,RATS,POPULATION.
840 B3=IP(5*RND)+1
850 B4=B3*I
860 B2=IP((B1+B4)*.07*RND)
870 B1=B1-B2+B4
880 J=IP(10*RND)
890 A2=IP(A2+(5-B3)*B1/600+1)
900 IF A2<=50 THEN 920
910 A2=50
920 A1=A1+A2-A3
930 GOTO 110
940 ! ERROR ROUTINE.

```

-Check to see if selling all  
land - option to restart

-Food allocation

-Plagues have a probability of  
1 in 10

# PROGRAM LISTING

```
950 BEEP 100,.2
960 DISP 'Think again ... you only have
      : '
970 DISP A1;'people ...'
980 DISP C1;'acres and ...'
990 DISP B1;'bushels in storehouses.'
1000 RETURN
1010 DISP 'You have ruled for';Z;'years.
      '

1020 ON ERROR GOTO 1030
1030 INPUT 'Do you want to continue?';Q$
1040 Q$=UPRC$(Q#[1,1])
1050 OFF ERROR
1060 IF Q$='N' THEN POP @ GOTO 1090
1070 IF Q##'Y' THEN 1020
1080 RETURN
1090 ON ERROR GOTO 1100
1100 INPUT 'Start over?'; Q$ @ Q$=UPRC$(
      Q#[1,1])
1110 OFF ERROR
1120 IF Q$='Y' THEN 80
1130 IF Q##'N' THEN 1090
1140 DELAY 1
1150 DISP TAB(10);'END OF GAME' @ DISP
1160 STOP
```

# PROGRAM DESCRIPTION

## REVERSE

The object of the game is to order a series of numbers from lowest to highest by reversing the order of a subset of numbers. For example, a series of five numbers, such as 25314, might be ordered by the following moves:

Start : 25314

Reverse 2 : 52314 (reverses the two leftmost numbers)

Reverse 5 : 41325

Reverse 4 : 23145

Reverse 2 : 32145

Reverse 3 : 12345 and the numbers are in order.

In this example, it took 5 reversals to win. For  $n$  numbers, a solution can always be found in  $n+1$  moves or less. The program allows the player to make any number of moves necessary.

The game allows a series of 3 to 9 numbers to be used in the game.

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		*** REVERSE ***	
1	Choose size of game (3-9)	How many numbers to reverse? 5	3-9 [RTN]
2	Display n numbers in random order. Allow player to reverse the order of the first r numbers. *	$n_1 \ n_2 \ n_3 \ . \ . \ . \ n_n \ \#$	2-n [RTN]
3	Display n numbers in their new order. Allow player to reverse the order of the first r numbers. *	$n_1 \ n_2 \ n_3 \ . \ . \ . \ n_n \ \#$	r [RTN]
4	Repeat step 3 until the numbers are in their correct order from smallest to largest.		
5	Display correct order	$n_1 \ n_2 \ n_3 \ . \ . \ . \ n_n$	
6	Display number of reversals taken: if $\leq n$ : if $= n+1$ : if $> n+1$ :	WOW! You won in m moves! You won in m moves! It took you m moves	
7	Option to play again	Run again, or End? R	E [RTN] or [RTN]
	If 'R' then step 1 else	END OF GAME	

**VARIABLE NAMES**

NAME	DESCRIPTION	NAME	DESCRIPTION
R	Number of digits to reverse	I	Index
N	Number of digits	K	Index for mixing numbers
A( )	Numbers in mixed order	J	Looping
T	Temp. storage of numbers	R\$	Number of digits to reverse
V	Number of reversals made	N\$	Number of digits
Q\$	Option to run again		

# PROGRAM LISTING

```
10 ! REVERSE - game to reverse
20 ! the order of a set of
30 ! numbers.
40 ! Revision 11/01/81
50 OPTION BASE 1
60 DELAY 1
70 RANDOMIZE
80 DIM R$(3),Q$(3),N$(3)
90 INTEGER A(9),N,I,T,V,R,K,J
100 DISP '      * * * REVERSE * * *' @
    WAIT 2
110 ON ERROR GOTO 120
120 INPUT 'How many numbers to reverse?
    ','5'; N$ @ N=VAL(N$)
130 IF N$(1,1)='0' THEN 590
140 IF N<=2 OR N>9 THEN 120
150 OFF ERROR
160 FOR I=1 TO N
170 A(I)=I
180 NEXT I
190 FOR I=N TO 2 STEP -1
200 K=IP(I*RND+1)

210 T=A(I)
220 A(I)=A(K)
230 A(K)=T
240 NEXT I
250 FOR I=1 TO N

260 IF A(I)#I THEN 290
270 NEXT I
280 GOTO 190
290 V=1
300 GOSUB 670
310 ON ERROR GOTO 300
320 INPUT '#',CHR$(95); R$ @ IF R$(1,1)
    =CHR$(95) THEN 590
330 R=VAL(R$)
340 IF R<=1 OR R>N THEN 300
350 OFF ERROR
360 V=V+1
370 FOR I=1 TO IP(R/2)

380 T=A(I)
390 A(I)=A(R-I+1)
400 A(R-I+1)=T
410 NEXT I
420 FOR I=1 TO N
430 IF A(I)#I THEN 300

440 NEXT I
450 DISP TAB(IP((32-N*3)/2));
460 FOR I=1 TO N
470 DISP A(I);
```

-Set up random index for mixing  
the order of the numbers

-Check to be sure that numbers  
are not still in order

-Enter the number of digits  
that you want to reverse

-Adjust the order of the  
numbers

-Check to see if solution has  
been found

-Display solution



# PROGRAM LISTING

```
480 NEXT I
490 FOR J=1 TO 3
500 FOR I=1 TO 5
510 BEEP 502.857+I*(220/7),.02
520 NEXT I
530 NEXT J
540 WAIT 1 @ DISP
550 IF V-1<=N THEN 580
560 IF V-2=N THEN DISP '  You won in';
      V-1;'moves !' ELSE DISP '  It took
      you';V-1;'moves.'
570 GOTO 590
580 DISP 'W O W !  You won in';V-1;'mov
      es!'
590 WAIT 2
600 ON ERROR GOTO 610
610 DISP CHR$(210);'un again, or ';CHR$(
      197);
620 INPUT 'nd? ', 'R'; Q$ @ Q$=UPRC$(Q$[
      1,1])
630 ON POS('RE',Q$)+1 GOTO 590,110,640
640 DISP '          END OF GAME' @ WAIT
      2 @ DISP
650 OFF ERROR
660 STOP
670 ! DISPLAY NUMBERS.
680 DISP TAB(IP((32-N*3)/2)-1);
690 FOR I=1 TO N
700 DISP A(I);
710 NEXT I
720 RETURN
```

# PROGRAM DESCRIPTION

## SLOT MACHINE

This program simulates the actions of a slot machine. The user may choose from five machines (coin denominations) to play. The coin denominations are (1) \$ .05, (2) \$ .10, (3) \$ .25, (4) \$ .50, (5) \$1.00. The higher the coin, the higher the payoff (or loss). Each machine uses the following payoff system:

⊖ cherry	⊖⊖⊖	Pays 8 times
○ orange	⊖⊖X	Pays 5 times
* lemon	○○○	Pays 10 times
◊ plum	***	Pays 15 times
# bell	σσσ	Pays 20 times
= bar	###	Pays 50 times
	===	Pays 100 times
	⊖XX	Pays 2 times
	##not=	Pays 3 times
	##=	Pays 5 times

Note: 'X' represents any symbol except ⊖. 'A' represents any symbol except ⊖ and is the same symbol for both occurrences of 'A' in a line.

At the end of each play, the user has the opportunity to "pull" again, or change machines, or stop.

# USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		SLOT MACHINE	
1	Select coin size	Choose slot machine (1-5): █	1-5 [RTN]
2	Assign the corresponding coin value to the machine.	You'll be using the .bb machine	
3	Pull the handle	※ ※ ※	
4	View roll, winnings or losses, and current cash total in hand (X represents a symbol)	X X X win .nn \$n.nn or X X X lose \$n.nn	
5	Option to pull again: If 'P' then goto step 3 If 'S' then goto step 6	Pull or Stop? P	[RTN] or S [RTN]
6	Option to change machines: If 'Y' then step 1 If 'N' then step 7	Try another machine? Y	[RTN] or N [RTN]
7	View total winnings or losses:	Total losses = n.nn or You broke even or Total winnings = n.nn	
8	End	END OF GAME	

# VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
X( )	Represents each symbol for a roll	K	"⌘" decimal code
I	Index and looping	B	Coin denomination
P	Payoff value	S	Total cash in hand
J	Tab values	Q\$	User interaction

# NOTES AND REFERENCES

Note: A players current cash total is displayed in the right most corner of the display window. If he changes machines this total stays with him.

# PROGRAM LISTING

```

10 ! SLOT - SLOT MACHINE.
20 !
30 ! Revision 11/01/82
40 !
50 DIM Q$(4)
60 INTEGER X(3),I,P,J,K
70 SHORT B,S
80 DELAY 2
90 RANDOMIZE
100 DISP '          SLOT MACHINE'
110 S=0
120 ON ERROR GOTO 130
130 INPUT 'Choose slot machine (1-5):',
    CHR$(95);Q$
140 Q$=UPRC$(Q$(1,1))
150 B=VAL(Q$)
160 DISP "You'll be using the ";
170 IF B=1 THEN B=.05 @ GOTO 230

180 IF B=2 THEN B=.1 @ GOTO 230
190 IF B=3 THEN B=.25 @ GOTO 230
200 IF B=4 THEN B=.5 @ GOTO 230
210 IF B=5 THEN B=1 @ GOTO 230
220 GOTO 120
230 DISP USING 240 ; B
240 IMAGE d.dd,' machine'
250 OFF ERROR
260 !
270 DELAY 0
280 K=31

290 FOR I=1 TO 5
300 DISP TAB(2);CHR$(K);
310 DISP TAB(5);CHR$(K);
320 DISP TAB(8);CHR$(K);
330 BEEP 141.429,.03
340 BEEP 172.857,.03
350 BEEP 204.286,.03
360 BEEP 235.714,.03
370 DISP CHR$(27)&'E';

380 NEXT I
390 J=2
400 DISP
410 DELAY 5
420 FOR I=1 TO 3
430 DISP TAB(J); @ BEEP 439.999
440 X(I)=IP(8*RND+1)

450 IF X(I)>5 THEN 650
460 IF X(I)>2 THEN 540
470 IF X(I)=2 THEN 510
480 DISP CHR$(18);
490 X(I)=1

```

-Establish coin size for  
machine selected

-Decimal code for upside down  
card

-Clear display and move to  
column 1

-Set up tab values

-Select a random number to  
correspond with a symbol

-cherry symbol

# PROGRAM LISTING

```
500 GOTO 760
510 DISP '0';
520 X(I)=2
530 GOTO 760
540 IF X(I)>3 THEN 580
550 DISP '*';
560 X(I)=3
570 GOTO 760
580 IF X(I)=5 THEN 620
590 DISP CHR$(9);
600 X(I)=4
610 GOTO 760
620 DISP CHR$(7);
630 X(I)=5
640 GOTO 760
650 IF X(I)>7 THEN 730

660 IF X(I)=7 THEN 710
670 IF I=1 AND RND<.8 THEN 510
680 DISP '=';
690 X(I)=6
700 GOTO 760
710 IF I=2 THEN 480
720 GOTO 550
730 IF I=1 THEN 620
740 IF I=3 AND X(1)<6 THEN 680
750 GOTO 590
760 J=J+3
770 NEXT I
780 DISP TAB(12);
790 P=0
800 IF X(1)#1 THEN 820

810 P=2
820 IF X(1)=X(2) THEN 880
830 IF P>0 THEN 1070
840 S=S-B
850 BEEP 220,.2 @ DISP USING 860 ; 'LOS
    E',S
860 IMAGE k,6x,' $',mdddd.dd
870 GOTO 1110
880 IF X(2)=X(3) THEN 930
890 IF X(1)=1 THEN 910
900 IF X(3)<6 THEN P=3 @ GOTO 1070
910 P=5
920 GOTO 1070
930 IF X(1)=1 THEN P=8 @ GOTO 1070
940 IF X(1)>2 THEN 970
950 P=10
960 GOTO 1070
970 IF X(1)>3 THEN 1000
980 P=15
990 GOTO 1070
1000 IF X(1)>4 THEN 1030
```

-Orange

-Lemon

-Plum

-Bell

-Weighting process for a bar.  
Chance of 3 bars is 1/2560

-Set up value of payoff  
-Check for first symbol being a  
cherry

# PROGRAM LISTING

```
1010 P=20
1020 GOTO 1070
1030 IF X(1)>5 THEN 1060
1040 P=50
1050 GOTO 1070
1060 P=100
1070 W=P*B
1080 S=S+W-R
1090 BEEP 660,.2 @ DISP USING 1100 ; 'WI
    N ',W,S
1100 IMAGE K,ddD.DD,' $',MDdDD.DD
1110 DELAY 2
1120 ON ERROR GOTO 1140
1130 DISP TAB(10);
1140 DISP ' ';CHR$(208);'ull or ';CHR$(2
    11);
1150 INPUT 'top? ', 'P';Q$
1160 Q#=UPRC$(Q#[1,1])
1170 IF Q#='P' THEN 250
1180 IF Q#='S' THEN 1190 ELSE 1110
1190 ON ERROR GOTO 1200
1200 DISP TAB(6);
1210 INPUT 'Try another machine?', 'Y';Q$
1220 Q#=UPRC$(Q#[1,1])
1230 IF Q#='Y' THEN 120
1240 IF Q#='N' THEN 1250 ELSE 1190
1250 DELAY 3
1260 DISP TAB(7);
1270 IF S<0 THEN DISP USING 'K,DDD.DD' ;
    'Total losses =',ABS(S)
1280 IF S=0 THEN DISP 'You broke even.'
1290 IF S>0 THEN DISP USING 'K,DDD.DD' ;
    'Total winnings =',S
1300 DISP TAB(11);
1310 DISP 'END OF GAME'
1320 DISP
1330 STOP
```

# PROGRAM DESCRIPTION

## BREAK-OUT

In this game the player uses a paddle and ball to knock out as many bricks as possible. The playing field is displayed on a video screen using the HP 82163A Video Interface. The paddle is controlled by using the "up" and "down" arrow keys. There are six layers of bricks. Bricks that are farther back are worth more points. There are five balls for each game. Bonuses are given for any leftover balls when all the bricks are gone.

There are two versions of the game, "Break-out" and "Break-thru". The difference between them is that in Break-out, the ball bounces back when it knocks out a brick, and in Break-thru the ball continues on its original path. In addition, there is an auto-play mode available, which will show the game running without user interface.

To play the game, assign the printer to the HP 82163A Video Interface and set the display device to the LCD. Run the program. The first prompt will be for the version of the game you wish to play: Break-out or Break-thru. Press "O" for Break-out, or "T" for Break-thru. Then press "A" for Auto-play mode, or "N" for regular play. Press any key to serve the ball, and use the "up" and "down" arrows to move the paddle. If you miss with the first ball and require a second, the serve will occur after you press any key.



# VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
T(,)	Target array: 0 if hit, 1 if not	M	Mode flag
X	Next X coordinate	X0	Current X coordinate
Y	Next Y coordinate	Y0	Current Y coordinate
I,J	Temporary counters	B	Current ball
S	Score	H	High score
U	Horizontal vector	V	Vertical vector
P	Paddle position	T	Count of targets left
A	Auto mode	K\$	Current key hit
R	Ricochet	E\$	Escape
L\$	Linefeed	B\$	Backspace
S\$	Inverse space	C\$	Clear, cursor off
A\$	Cursor address	A0\$	Address column 0
A3\$	Address column 3	A5\$	Address column 5
U\$	Paddle up	D\$	Paddle down

# NOTES AND REFERENCES

Thanks to Ed Groth for the original program, Jack Applin IV and Raan Young of CVD for their contributions.

# PROGRAM LISTING

```

10 ! Breakout- video game
20 ! Requires 82163A
30 !
40 ! Revision 11/01/82
50 !
60 OPTION BASE 1
70 DIM E$(11),L$(11),B$(11),S$(11),C$(14),A
  $(2),A0$(3),A3$(3),A5$(3),K$(11),U$(
  6),D$(6)
80 INTEGER T(6,13),X,Y,X0,Y0,I,J,S,B,M
  ,U,V,P,T,H
90 E%=CHR$(27) @ L%=CHR$(10) @ B%=CHR$(
  8) @ S%=CHR$(160)
100 C%=E%&'E'&E%&'<'
110 A%=E%&'Z' @ A0%=A%&CHR$(0) @ A3%=A%
  &CHR$(3) @ A5%=A%&CHR$(5)
120 U%=S%&L%&L%&L%&B%&' '
130 D%= ' '&L%&L%&L%&B%&S%
140 H=0
150 RANDOMIZE
160 PRINTER IS ":TV" @ PWIDTH INF @ PRI
  NT C%
170 PRINT 'Do you wish to play Break-Ou
  t or Break-Thru? Break-';
180 K%=UPRC$(KEY%)
190 IF K%="Q" THEN M=-1 @ PRINT "Out" @
  GOTO 220
200 IF K%="T" THEN M=1 @ PRINT "Thru" @
  GOTO 220
210 GOTO 180
220 PRINT
230 PRINT "Auto or Normal mode? ";
240 K%=UPRC$(KEY%)
250 IF K%="A" THEN A=1 @ PRINT "Auto" @
  GOTO 300
260 IF K%="N" THEN A=0 @ PRINT "Normal"
  @ GOTO 280
270 GOTO 240
280 PRINT
290 PRINT 'Use up/down arrows to move
  paddle; any key to serve ball'
300 ! Construct playing board
310 WAIT 5 @ PRINT C%
320 ! Build frame
330 FOR I=0 TO 31 @ PRINT A%;CHR$(I);CH
  R$(0);S%; @ NEXT I
340 FOR I=1 TO 13 @ PRINT A%;CHR$(31);C
  HR$(I);S%; @ NEXT I
350 FOR I=31 TO 0 STEP -1 @ PRINT A%;CH
  R$(I);CHR$(14);S%; @ NEXT I
360 ! Build targets
370 FOR J=1 TO 6 @ FOR I=1 TO 13
380 T(J,I)=1
390 PRINT A%;CHR$(2*J+16);CHR$(I*(1-MOD
  (J,2))+((13-MOD(J,2))*(I-1))*MOD(J,2)
  );S%;

```

-Clear, cursor off

# PROGRAM LISTING

```

400 NEXT I @ NEXT J
410 ! Initialize ball, score, paddle, t
    arget count
420 B,S=0 @ P=7 @ T=78
430 ! Serve ball: build paddle, get bal
    l & coordinates, show score, wait f
    or key
440 PRINT A0$;CHR$(P-1);S$;L$;R$;S$;L$;
    R$;S$;
450 R=0 @ B=B+1 @ IF B>5 THEN 960
460 IF A THEN GOTO 480
470 IF KEY$='' THEN GOTO 470
480 GOSUB 1150
490 X0=14 @ Y0=INT(26*RND)+2 @ U=-1 @ V
    =INT(5*RND)-2
500 ! Find next point; let paddle move

510 X=X0+U @ Y=Y0-V
520 GOSUB 1100
530 ! Did ball hit walls?
540 IF Y<2 OR Y>27 THEN V=-V @ GOTO 780
550 ! Did ball hit end?
560 IF X>30 THEN U=-U @ R=1 @ GOTO 780
570 ! Did ball miss paddle?
580 IF X<0 THEN GOTO 870
590 ! Did ball hit paddle?
600 IF X=0 AND (Y0\2)=P-1 AND Y0\2<=P+1
    OR Y\2=P-1 AND Y\2<=P+1 THEN GOT
    O 810
610 ! Ball in target range?
620 IF X>=18 AND X<=28 AND MOD(X,2)=0 T
    HEN GOTO 670
630 ! Move ball, go find next point
640 GOSUB 1050
650 GOTO 510
660 ! Did ball hit target?
670 I=(X-18)\2+1 @ J=Y\2
680 IF NOT T(I,J) THEN GOTO 640
690 ! Process target hit
700 BEEP X*40+500,.05 @ T(I,J)=0 @ S=S+
    (I+3)*2 @ T=T-1
710 U=U*M
720 IF NOT R AND M<0 THEN U=-ABS(U)
730 GOSUB 1050
740 GOSUB 1150
750 IF T=0 THEN 910
760 GOTO 510
770 ! Process wall or end hit
780 X=X0 @ Y=Y0 @ BEEP 400,.05
790 GOTO 640
800 ! Process paddle hit
810 IF V=0 THEN V=V+INT(3*RND)-1
820 U=-U @ Y=Y0 @ X=X0 @ BEEP 600,.05
830 IF Y0\2<=P-1 AND V<0 OR Y0\2>=P+1 A
    ND V>0 THEN V=-2*SGN(V)

```

# PROGRAM LISTING

```

840 V=V+INT(3*RND)-1 @ IF ABS(V)>2 THEN
    V=2*SGN(V)
850 GOTO 640
860 ! Process paddle miss
870 BEEP 20,.5
880 PRINT A0$;CHR$(Y0\2);' ';
890 GOTO 440
900 ! Cleared board: show bonus, score,
    restart
910 PRINT A$;CHR$(X0);CHR$(Y0\2);' ';
920 PRINT A3$;CHR$(3);'Bonus for';6-B;'
    balls left: ';150*(6-B);
930 S=S+150*(6-B)
940 GOSUB 1150
950 ! Out of balls if entered here
960 IF S>H THEN H=S
970 PRINT A3$;CHR$(5);'High: ';H;
980 PRINT A3$;CHR$(7);'Again? ';
990 K$=UPRC$(KEY$) @ IF A THEN K$="Y"
1000 IF K$='Y' THEN PRINT 'Yes' @ GOTO 3
    10
1010 IF K$# 'N' THEN GOTO 990
1020 PRINT C$
1030 END
1040 ! Move ball subroutine
1050 PRINT A$;CHR$(X0);CHR$(Y0\2);' ';
1060 PRINT A$;CHR$(X);CHR$(Y\2);'*';
1070 X0=X @ Y0=Y
1080 RETURN
1090 ! Move paddle subroutine
1100 K$=UPRC$(KEY$)
1110 IF (K$=' ' OR A AND Y0\2<P) AND P>2
    THEN P=P-1 @ PRINT A0$;CHR$(P-1);U
    $;
1120 IF (K$=' ' OR A AND Y0\2>P) AND P<1
    2 THEN PRINT A0$;CHR$(P-1);D$; @ P=
    P+1
1130 RETURN
1140 ! Show score subroutine
1150 PRINT A5$;CHR$(15);'Ball: ';B;' Sc
    ore: ';S;
1160 RETURN

```