DEVELOPMENT OF GAMES

The first type of game to be programmed on the 65 was the single-player chance-type, such as Blackjack. A random number generator is used to determine the results.

The next development was to include some criteria by which the player could increase his winnings with the application of some knowledge, such as in the Gunner and Tic-Tac-Toe games.

Shortly after the development of the timer potential, games of skill, such as Golf, were developed.

As more and more games became popular, and the amount of time spent playing games increased, the 2-player games were developed so both the owner and the person he was showing could take an active part. Football is one 2-player chance game, Rock-Paper-Scissors is a knowledge type game, and Hot Potato is a 2-player skill game.

With the three basic types developed, the next logical game was one which included all three facets (chance, knowledge, skill). Bagels was, of course, the most popular and has so many variations that it is a subject all by itself.

Learning games were the next type developed; these games would start with the highest priority on chance and then switch to skill as the game is played more and more. Mr. Ahl, in his book "101 Basic Games", states that this type of game and this type of programming are the thing that will be the most valuable in the future of computers. And I strongly agree.

The newest development is the most valuable (as usual). Some attempts at word games have been made from time to time. However, the translation tables were so hard to learn that most new users lost interest. When Mr. Rausch of Dayton invented and distributed his layover template, the whole world of word games exploded. A new set of people became interested. Those who don't like numbers and math will sit for hours trying to spell a word. The increased complexity even captured the interest of some of the people who had thought of the 65 as 'underpowered'.

While this is the end of this article, in a year it will take 2 or 3 more paragraphs to cover this subject.

(41) D. Lampman, Piqua, OH.

PROGRAM: FOOTBALL BY: DEAN LAMPMAN

Program Description, Equations, Variables:

This program pits two players against each other in a game of Football. The defense chooses his play first and enters it. The offense then executes his play, and the yard line the ball is on is displayed. The computer will keep track of whose ball it is and the number of yards needed for a first down, the score, and which down it is.

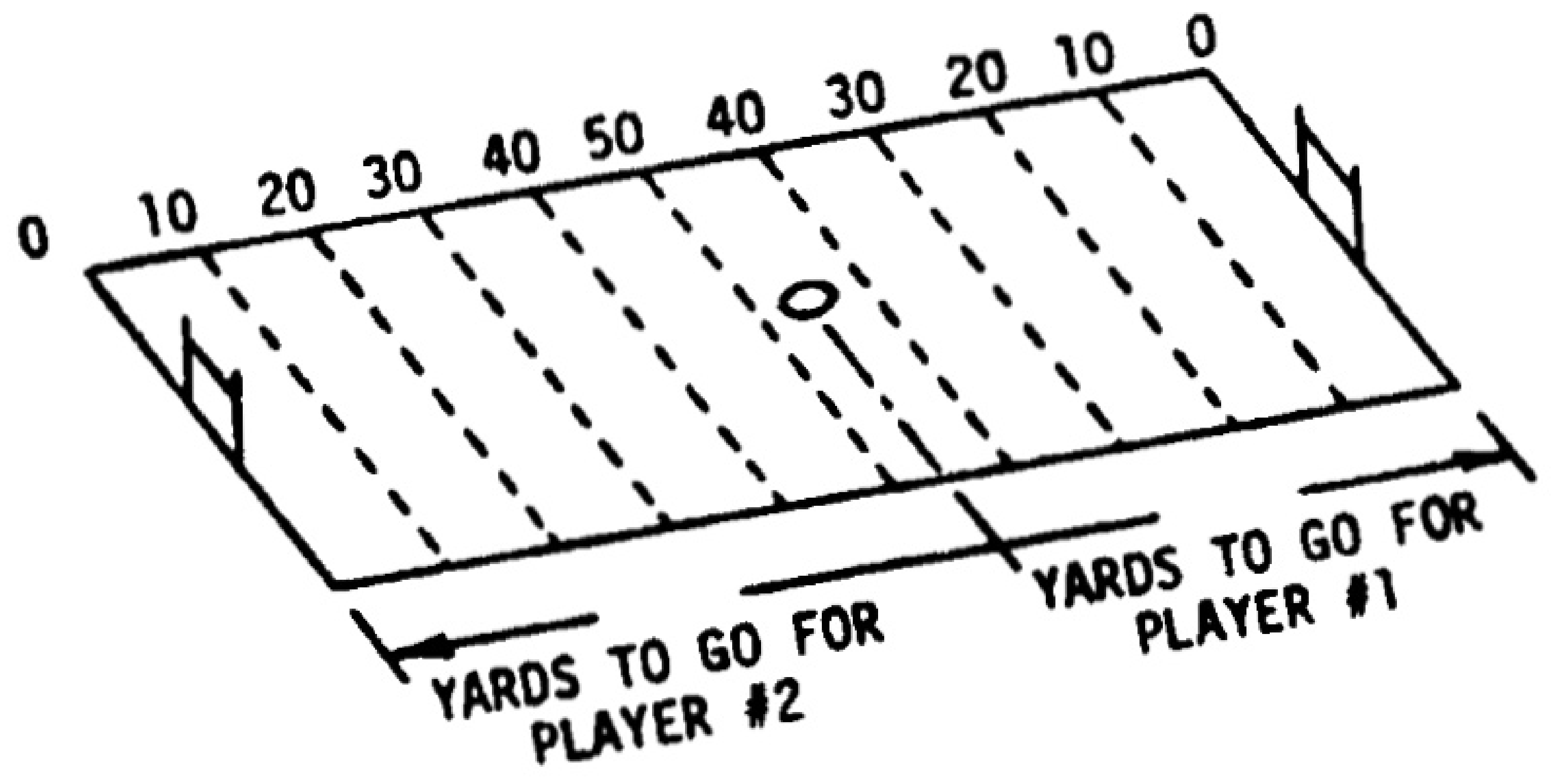
The plays are:
1) Quarterback sneak.
2) Line plunge.
3) End run.
4) Statue of Liberty.
5) Sideline pass.
6) Screen pass.
7) Criss-cross pattern pass.
8) Down and out pass.
9) The long bomb.

Yardage is harder to get near the goal line than at mid-field.

Operating Limits and Warnings:

The only score is a 7-point touchdown. The kickoff and run back always results with the ball on the 50-yard line, on a touchdown, or a new game. The time must be kept by the user -- for the half and end of game

Sketch:



Sample Problem:

How to beat your opponent; remembering that --- the higher the play number, the more yards you get; the closer your opponent defends your play, the less you get; the closer the goal, the less yards you get.

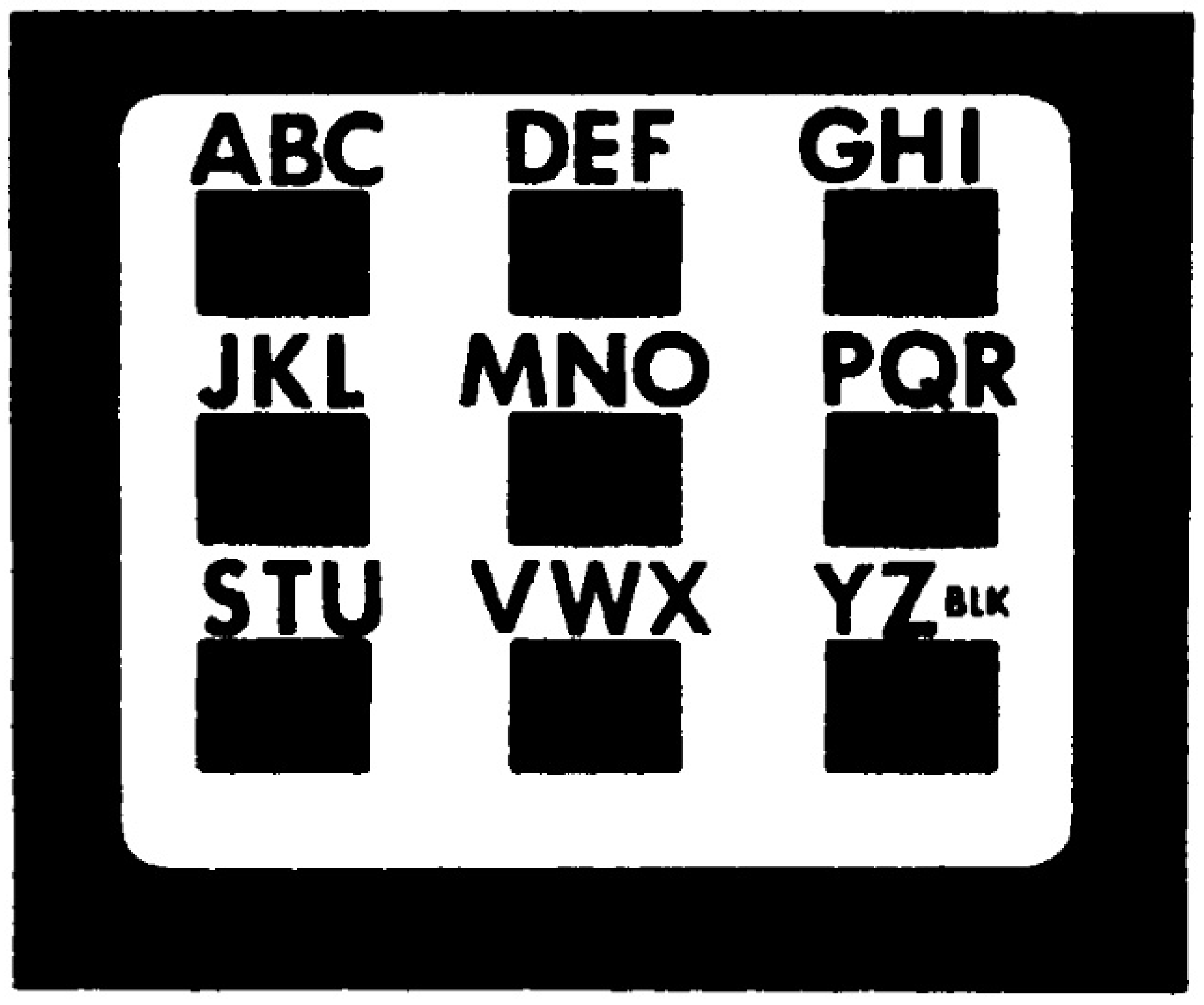
Solution:

Go for long yards on the early downs at mid-field, and go for short yards at the goal on 3rd and 4th downs.

KEYBOARD OVERLAY

The use of keyboard overlays is common practice with desk calculators, but it took John Rausch (88) to apply the overlay concept to the HP-65. His overlay design is reproduced below and is cut out of a yellow plastic IBM card. The idea is to use a digit entry coupled with a label key to encode or decode the alphabet plus a blank.

To make an overlay for your own use, cut out the overlay below - after covering one or both sides with Scotch Magic Mending Tape. An alternate approach is to use the drawing below as a template to cut one out of a more durable material. Press-on letters (18 point size is used below) make letter identification neat. An Xacto knife works well.



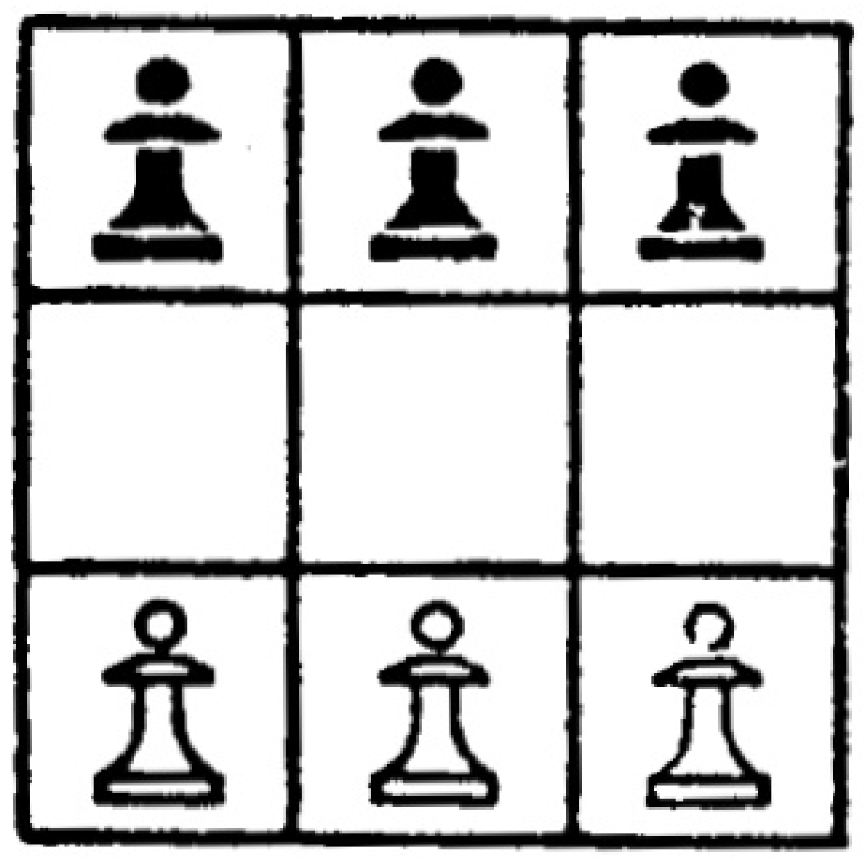
With the exception of some new word games that you will find elsewhere in this issue of 65 Notes, most games for the HP-65 have been numeric in nature. In addition, these games have used a fixed strategy for winning (NIMB) or they have used a random number generator in place of any strategy at all (CRAPS). This is usually true with computers of any size, not just the HP-65.

However, there is another way to program games for the HP-65. It can be programmed to learn the best strategy from its own mistakes! In fact, this allows you to program the HP-65 to play a game when you don't know the best strategy yourself. Cybernetics, or artificial intelligence, is the term given to this kind of programming.

One way to program the HP-65 to learn a game is to store all of the possible moves that it could be confronted with for a particular game configuration, select one of the moves at random when it is the HP-65's turn to play, and 'punish' it by removing any moves that cause it to lose. Eventually, the HP-65 will be able to play a perfect game. This is the easiest method to program, and it leads to the fastest education of the HP-65. There are several other methods of programming a computer to learn a game, but for the HP-65 this is the only one that is practical because of its size.

A game called HEXAPAWN (devised by Martin Gardner) can be used to demonstrate how the HP-65 can be programmed to learn the best strategy for a game. HEXAPAWN is played on a 3X3 board, with three pawns on each side as shown in Figure 1. Only two types of moves are allowed: (1) a pawn may advance straight forward one square; (2) a pawn may capture an enemy pawn by moving one square diagonally, left or right, to a square occupied by the enemy. The captured piece is removed from the board. The game is won in any of three ways: (1) by advancing a pawn to the third row; (2) by capturing all of the enemy pieces; (3) by achieving a position in which the enemy cannot move. You may play first or second, but once you choose when you want to play, you must continue to play the same way unless you completely start over. It is not immediately apparent whether the first or second player has the advantage. You are encouraged to NOT analyze the game but, instead, to learn along with the HP-65.

Figure 1.



Two program cards are required. One is used to prime the registers with the possible moves and set up a random number seed. The other is used to play the game. After entering and recording the programs in Figure 2, you will be ready to play.

(1) Enter program number 1. Press A if the HP-65 is to play first, or B if you are to play first. Let the calculator run a few seconds (10 or 15), then press R/S.

(2) Enter program number 2.

(3) Either make your move or tell the HP-65 that the board configuration is 1 so it can make its opening move. Whenever it is the HP-65's turn to move, you must tell it the board configuration. This is done by finding it in Figure 3, entering the number, and pressing A. When you move first, you have a choice between a center or an end opening, but only the left end is allowed because an opening on the right would obviously lead to identical lines of play (although mirror reflected). When you play second, you are only allowed to capture an opening move by the HP-65 to the center square by moving diagonally to the right for the same reason. After you press A, the HP-65 will respond with its move. If its move is zero, it forfeits the game because no acceptable move can be found. Otherwise, the number indicates which of the possible moves for the board configuration just entered that it wants to make. Continue playing until either you or the HP-65 wins. If you win, the HP-65 must be punished. This is done by pressing B. The HP-65 must be punished regardless of whether it loses by a forfeit or by your move.

(4) It is very interesting to chart the won-lost record of the HP-65. Try it once playing your best, then play dumb. You'll find that the better you play, the faster it learns.

If you have a friend with an HP-65, try setting them up to play against each other. Since either the first or second player has the advantage, eventually one HP-65 will forfeit right off the bat when confronted with the first configuration.

Interested readers are encouraged to read Chapter Eight of The Unexpected Hanging and Other Mathematical Diversions by Martin Gardner. It is published by Simon and Schuster ($2.95).

John Rausch

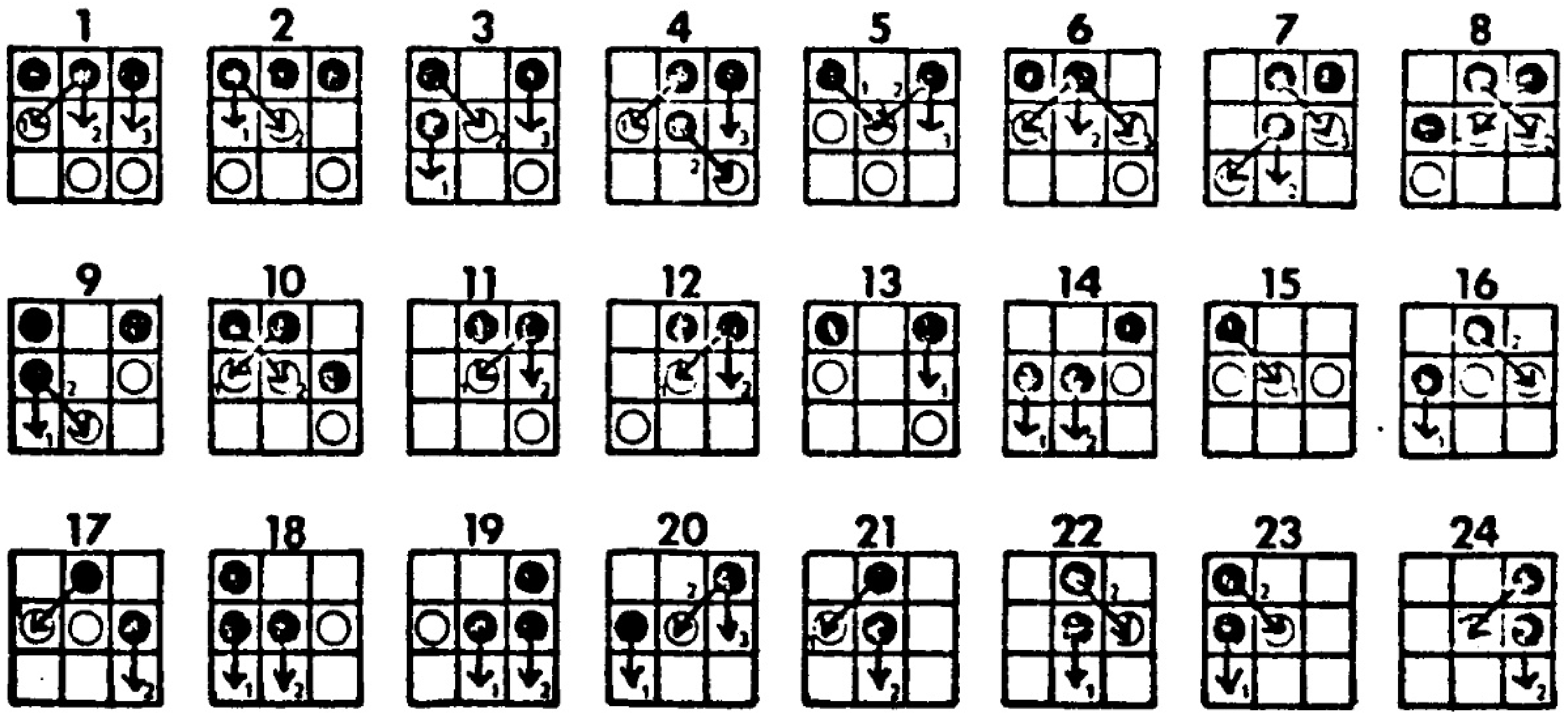
← Blank to allow you to cut out the keyboard overlay (templet)

HP-65 User Instructions

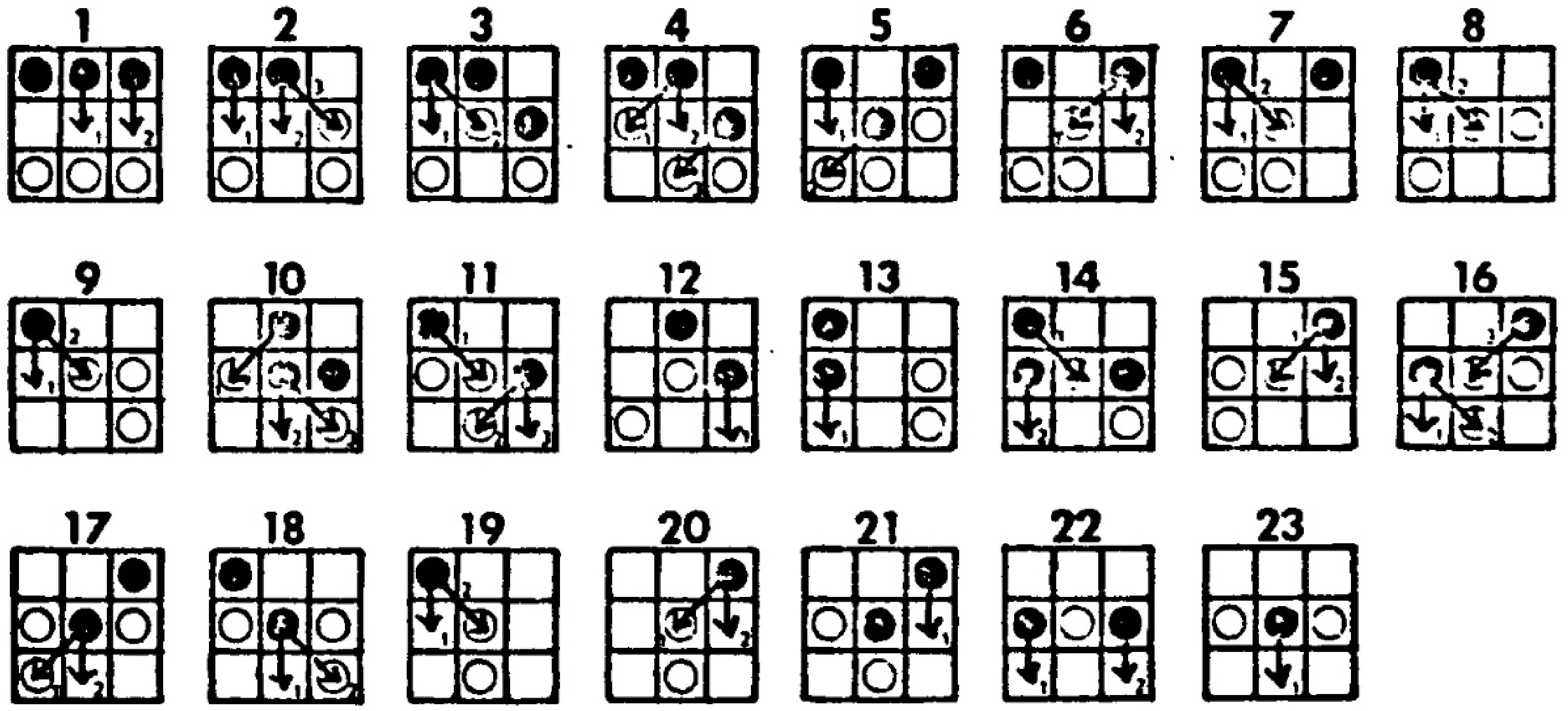
HE/APAWN (CYBERNETIC GAME) BY JOHN RAUSCH Page 1 of 3
(BASED ON PROGRAM BY JACOB JACOBS)

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	ENTER PROGRAM CARD #1.				
2	INITIALIZE. LET PROGRAM RUN FOR AN ARBITRARY LENGTH OF TIME, THEN PRESS R/S		A	R/S	?
3	ENTER PROGRAM CARD #2.				
	RULES ARE -- THERE ARE THREE PAWNS ON EACH SIDE OF A THREE-BY-THREE BOARD TO START.				
	ONLY TWO TYPES OF MOVES ARE PERMITTED:				
	1) ADVANCE ONE SQUARE FORWARD TO AN EMPTY SPACE				
	2) CAPTURE AN ENEMY PAWN BY MOVING ONE SQUARE DIAGONALLY AS IN CHESS.				
	THE GAME CAN BE WON IN THREE POSSIBLE WAYS:				
	1) BY ADVANCING A PAWN TO THE THIRD ROW.				
	2) BY CAPTURING ALL OF THE ENEMY PAWNS.				
	3) BY ACHIEVING A POSITION IN WHICH THE ENEMY CANNOT MOVE.				
	YOU ALWAYS MOVE FIRST. THERE ARE THREE POSSIBLE OPENING MOVES, BUT TWO ARE MIRROR IMAGES OF EACH OTHER, SO ONLY ONE IS PERMITTED. THIS IS BECAUSE BOTH MOVES WOULD EVENTUALLY LEAD TO IDENTICAL LINES OF PLAY AND WOULD NOT HELP TO DEMONSTRATE THE CYBERNETIC PROPERTIES OF THIS PROGRAM. THERE ARE TWENTY-FOUR POSSIBLE BOARD CONFIGURATIONS AFTER YOUR MOVES. THESE ARE SHOWN IN THE ACCOMPANYING DIAGRAM. YOU MUST TELL THE HP-65 THE BOARD CONFIGURATION AFTER EACH MOVE. THE HP-65 WILL THEN RESPOND WITH ITS MOVE. IF ITS MOVE IS ZERO, IT FORFEITS THE GAME BECAUSE NO ACCEPTABLE MOVE HAS BEEN FOUND. FOR EXAMPLE, ONCE IT HAS BEEN BEATEN FROM BOARD CONFIGURATION FIFTEEN, THE NEXT TIME IT FINDS THE BOARD LIKE THIS MOVE ONE IS UNACCEPTABLE SO IT FORFEITS THE GAME. OTHERWISE, THE HP-65 INDICATES WHICH OF THE MOVES IN THE BOARD CONFIGURATION YOU SPECIFIED THAT IT WANTS. PLAY CONTINUES UNTIL SOMEBODY WINS. IF THE HP-65 WINS, JUST START OVER BY ENTERING YOUR INITIAL BOARD CONFIGURATION. IF YOU WIN, EITHER BY YOUR MOVE OR THE HP-65 FORFEITING, THE HP-65 MUST BE "PUNISHED". THIS IS DONE SIMPLY BY PRESSING B . THE SAME MISTAKE WILL NOT BE MADE AGAIN!				
	THE FOLLOWING IS A SUMMARY OF THE INPUT YOU PROVIDE:				
	*BOARD CONFIGURATION AFTER YOUR MOVE..	1...24	A		MOVE(1-3)or FORFEIT (0)
	*PUNISHMENT WHEN HP-65 LOSES.		B		0
	AS PLAY CONTINUES, THE HP-65 WILL PLAY BETTER AND BETTER UNTIL IT EVENTUALLY BECOMES UNBEATABLE. IN FACT, THE BETTER YOU PLAY, THE QUICKER IT BECOMES A "MASTER". TRY PLOTTING THE WON-LOST RECORD ON GRAPH PAPER. ALSO TRY DUMB MOVES; IT ONLY LEARNS BY MISTAKES!				

You move first.



You move second.



Note: Configurations 6 and 7 are the same except for the moves. Use 6 until it results in a forfeit. When this occurs, immediately enter 7 and press A so the HP-65 can select a move. From this point on always use 7 in the usual manner.

HP-65 Program

HEXAPAWN (CYBERNETIC GAME) CARD #1 BY JOHN RAUSCH

Page 2 of 3

KEY ENTRY	CODE SHOWN	COMMENTS
f	31	HP-65 MOVES FIRST
REG	43	CLEAR REGISTERS,
8	08	THEN STORE MOVES.
3	03	
8	08	
8	08	
6	06	
0	00	
7	07	
STO 1	33 01	
3	03	
1	01	
3	03	
9	09	
5	05	
8	08	
3	03	
STO 2	33 02	
3	03	
4	04	
3	03	
1	01	
4	04	
STO 3	33 03	
GTO	22	
0	00	
LBL	23	HP-65 MOVES SECOND.
B	12	
f	31	CLEAR REGISTERS.
REG	43	THEN STORE MOVES.
1	01	
6	06	
7	07	
7	07	
7	07	
2	02	
1	01	
5	05	
STO 1	33 01	
1	01	
6	06	
7	07	
5	05	
6	06	
7	07	
3	03	
5	05	
STO 2	33 02	
5	05	
2	02	

KEY ENTRY	CODE SHOWN	COMMENTS
4	04	
4	04	
1	01	HP-65 USERS CLUB
3	03	65 NOTES V2 N3
STO 3	33 03	
LBL	23	GENERATE SEED --
0	00	
DSP	21	SET DISPLAY FOR
.	83	NO DECIMALS.
0	00	
0	00	START SEED AT ZERO.
LBL	23	
1	01	
g	35	ADD π TO PRIOR X AND
π	02	CONTINUE UNTIL R/S
+	61	
STO 6	33 06	
GTO	22	
1	01	

REGISTERS

R_1 1 MOVES

R_2 2 MOVES

R_3 3 MOVES

R_4

R_5

R_6 RANDOM # SEED

R_7

R_8

R_9

LABELS

A	HP-65 1ST
B	HP-65 2ND
C	
D	
E	
0	GEN.SEED
1	LOOP
2	
3	
4	
5	
6	
7	
8	
9	

FLAGS

1

2

TO RECORD PROGRAM INSERT MAGNETIC CARD WITH SWITCH SET AT W/PRGM

HP-65 Program

HEXAPAWN (CYBERNETIC GAME) CARD #2 BY JOHN RAUSCH

Page 3 of 3

SWITCH TO W/PRGM PRESS [f] [PRGM] TO CLEAR MEMORY

KEY ENTRY	CODE SHOWN	COMMENTS
STO 8	33 08	BOARD CONFIGURATION.
RCL 4	34 04	STORE CONFIGURATION.
.	83	THEN STORE PRIOR MOVE
1	01	NUMBER & 2 (CONFIG) IN
X	71	7 IN CASE OF FORFEIT.
RCL 5	34 05	
+	61	
STO 7	33 07	
2	02	RAIZE 2 TO THE POWER OF
RCL 8	34 08	THE BOARD CONFIGURATION
g	35	RESULT MUST BE ROUNDED.
Y^x	05	
EEX	43	
9	09	
+	61	
gLSTx	35 00	
-	51	
STO 5	33 05	
3	03	SET FOR THREE ITERATIONS
STO 8	33 08	
RCL 6	34 06	GENERATE RANDOM NUM-
9	35	BER FROM ONE TO THREE
π	02	FOR FIRST MOVE ATTEMPT.
+	61	STORE IT IN 4.
f-1	32	
√x	09	
f-1	32	HP-65 USERS CLUB
INT	83	65 NOTES V2 N3
STO 6	33 06	
3	03	
X	71	
1	01	
+	61	
f	31	
INT	83	
STO 4	33 04	
LBL	23	MOVE ATTEMPT LOOP
0	00	
3	03	ADD ONE TO MOVE NUM-
RCL 4	34 04	BER. IF RESULT IS MORE
1	01	THAN 3, RESET MOVE
+	61	TO ONE.
gX>Y	35 24	
CLX	44	
1	01	
STO 4	33 04	
2	02	RECALL 1,2 OR 3 DEPEND-
RCL	34	ING ON MOVE NUMBER.
gX≠Y	35 21	
gX>Y	35 24	

KEY ENTRY	CODE SHOWN	COMMENTS
1	01	
2	02	
3	03	
0	00	
gR↓	35 08	1, 2 OR 3 NOW IN X
RCL 5	34 05	DIVIDE MOVE BY 2 (CONFIG).
÷	81	
f-1	32	TRUNCATE TO DECIMAL.
INT	83	FRACTION.
.	83	
5	05	IF .5 IS LESS THAN OR
gX≤Y	35 22	EQUAL TO FRACTION, THE
RCL 4	34 04	MOVE IS ACCEPTABLE.
RTN	24	
g	35	MOVE IS NOT ACCEPTABLE
DSZ	83	TRY ANOTHER.
GTO	22	
0	00	
RCL 7	34 07	NO ACCEPTABLE MOVES.
↑	41	PUT PRIOR MOVE NUMBER
f	31	AND 2 (CONFIG) IN 4 AND 5
INT	83	FOR PUNISHMENT SO IT
STO 5	33 05	WON'T GET INTO THIS
-	51	SITUATION AGAIN.
1	01	
0	00	
X	71	
STO 4	33 04	
CLX	44	SET DISPLAY AT ZERO TO
RTN	24	INDICATE FORFEIT.
LBL	23	PUNISH --
B	12	
RCL 5	34 05	2(CONFIG-1)IS SUBTRACTED
2	02	FROM 1,2, OR 3 DEPEND-
÷	81	ING ON MOVE NUMBER.
gR↑	35 09	THIS PREVENTS THIS
2	02	MOVE FROM BEING
RCL 4	34 04	ACCEPTABLE AGAIN.
STO	33	
-	51	
gX>Y	35 24	
gR↑	35 09	
3	03	
gX=Y	35 23	
gR↑	35 09	
2	02	
gR↑	35 09	
1	01	
CLX	44	
RTN	24	

REGISTERS

R_1 1 MOVES

R_2 2 MOVES

R_3 3 MOVES

R_4 CURRENT MOVE #

R_5 CURRENT 2 (CONFIG)

R_6 RANDOM #SEED

R_7 PRIOR MOVE AND 2 (CONF)

R_8 LOOP OTL AND WORK

R_9 SCRATCH

LABELS

A	CONFIGUR
B	PUNISH
C	
D	
E	
0	LOOP
1	
2	
3	
4	
5	
6	
7	
8	
9	

FLAGS

1

2

TO RECORD PROGRAM INSERT MAGNETIC CARD WITH SWITCH SET AT W/PRGM