FOUR DIMENSIONAL TIC-TAC-TOE

Four dimensional tic-tac-toe is a logical extension of the familiar two dimensional tic-tac-toe. Once the basic concept of converting four dimensions into two is grasped, the game is easy to play.

Imagine first a four square by four square (16 squares) playing board. Stack three identical boards on top of the first such that each square on a board is the bottom of a cube. This is a three dimensional tic-tac-toe board, a cube composed of 64 small cubes. Any sequence of four cubes that spans the cube from one surface to another in a straight line is a winning combination. To visualize these combinations more easily, imagine that instead of four horizontal boards, there were four vertical boards, or two boards (slightly stretched) placed between the edges of the cube, forming an "X". All sequences of squares that are winning combinations on the boards in two dimensions are also winning combinations in three dimensions. There are no other winning combinations in three dimensional tic-tac-toe. The same can be tried in two dimensions using a one dimensional board to find the winning combination.

Before proceeding to four dimensions, the three dimensional playing board must be represented in two dimensions. This is simple to do by unstacking the four two dimensional boards which compose the three dimentional board, and lying them top to bottom in a column. Instead of trying to visualize a four dimensional tic-tac-toe board, it is much easier to convert it into three dimensions. A four dimensional board becomes four three dimensional boards side by side in a row. The three dimensional

representation can be compressed into two dimensions by unstacking the four two dimensional boards each three dimensional board is composed of, and placing them in columns. That leaves sixteen boards arranged four by four, each of which contains sixteen squares arranged four by four.

To find all the combinations use the two dimensional representation of a three dimensional board (four two dimensional boards in a column). Superimpose it on the four rows, four columns, and (with each board turned 45^{o}) the two diagonals of two dimensional boards which make up the four dimensional board. Each sequence of four squares that corresponds to a winning combination on the superimposed three dimensional board is also a winning combination in four dimensional tic-tac-toe. There are no other winning combinations.

The four dimensional board represented in two dimensions looks like a big two dimensional board whose squares are smaller two dimensional boards. The similarity is very useful. Each small board has ten winning combinations. Each combination can be represented by a sequence of four squares. The sequence can be given in two different directions. The same combinations can be used to specify a sequence of small boards within the big board. Two sequences, specifying a board and a square within that board are combined to specify a sequence of four squares in four dimensions.

Using the winning combinations of two dimensional tic-tac-toe, specified by sequences of squares in both directions, any two sequences may be combined to give a winning combination in four dimensional tic-tac-toe. In addition to these combinations, each board is a tic-tac-toe game in itself, and the square's position can remain constant while the boards follow some winning sequence.

How to Play the Game

The four dimensional tic-tac-toe program is a game in which one person plays against the computer. The player makes his move by selectin the number which corresponds to the square he wishes to occupy. The computer will show this move on a color television display controlled by a Cromemco Dazzler. The computer then makes its move, which is shown on the television display.

The first 1.75K of memory contains the program. The next .25K of memory is reserved for the program stack. The display is located from 2K to 2.5K (this must be static RAM). A .5K workspace fills the rest of the 3K. All of the non-program memory must be RAM.

The program starts from location 0000. When started, it disables the interrupt system, and turns on the Dazzler display. It then asks if it can play first. If a "Y" is typed, the computer will make the first move. If any other character is typed, then the player may make the first move. A playing board is then constructed on the display, and the computer is ready to accept the player's move. If the computer made the first move, that move will be displayed. The computer will then accept the player's move.

The player enters his move by entering from a ASCII keyboard the number of the square he wishes to take (see Figure 1). The computer will make sure the square has not been taken. The computer is then ready to accept another input. If the square is unoccupied, the square's number is output to the lights and is marked in yellow on the display. If the player is unsatisfied with his move, he may type a space. The computer will extinguish the yellow square, and wait for another move to be input. If the player is satisfied with his move, he may type a return and the

computer will change the yellow square to the player's color. The computer will then make its move. When that move appears on the screen and on the lights, the computer is ready to accept the player's next move. If the move appears in white, the game is a tie, and the program jumps to the beginning.

If the computer discovers that the player has four squares in a row, it will turn the winning squares green and jump to the beginning of the program. If the computer finds that it will have four in a row after it makes its move, it turns the possible winning squares yellow. If the player does not also have four squares in a row, the computer will turn the yellow squares white, output the winning square's number to the lights, and jump to the beginning of the program. If the player does have four in a row, the computer will turn those squares green and jump to the beginning of the program.

How the Program Operates

The computer makes its moves by examining each winning combination of four squares. The computer determines which of nine categories each combination fits into. The nine categories are: all the squares empty; one, two, three, or four squares occupied by the player and the rest empty; one, two, or three squares occupied by the player and the rest empty; or some squares occupied by the player and some by the computer. In the later case, the computer does nothing and continues on to the next combination. In the cases where zero, one, or two squares are occupied, the computer uses two words of memory corresponding to each square. This forms a sixteen bit word for each square. If all the squares are empty, then, for each square, the computer adds one to the

least significant word in memory corresponding to the empty square. If the computer or the player occupies one square, and the rest are empty, then for each square the computer adds one to the most significant word in memory which corresponds to the empty square. If the player has two squares, the computer adds 10H to the most significant words in memory corresponding to the two empty squares. The computer will not add more than 30H to any one square. If the computer has two squares, it adds 40H to the most significant words corresponding to the two empty squares. The computer will not add more than OCOH to any one square.

In the cases where three squares are occupied, the computer remembers which square was empty. If the player has three squares, then the computer must block him by taking the empty square. If the computer has three squares, then it will remember the empty square it needs to win, and forget about blocking. It will turn all four squares yellow, and continue on to see if the player has won.

If the player has four squares in a row, then the computer stops looking at winning combinations. It will turn the winning squares green and jump to the beginning of the program.

After all winning combinations have been looked at, the computer must decide which square it wants to take. If the player has won, the computer will not reach this point. It first checks to see if it has already chosen a winning square. If it has, it will output that square on the lights, change the yellow squares to red, and jump to the beginning of the program. If it has chosen a blocking square, it will output that square to the lights and display and wait for the player's next move.

If no square has already been chosen, the computer must look at the words of memory which correspond to the squares. The computer will pick the square whose two words of memory contain the greatest sixteen bit value. The selected square is output on the lights and display, and the computer waits for the player's move. If the game turns out to be a tie, the computer will output its move in red and jump to the beginning of the program.

Note- In addition to a Cromemco Dazzler to generate the color TV display, a CRT terminal or Teletype is required to play 4D TIC TAC TOE. Messages from the computer appear on the teletype or CRT display while the keyboard is used for input.

THE BOARD

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Fig. 1