

4次元空間立体四目並べプレイソフトの作成

水 間 輝*・苫米地 宣 裕**

Development of the Software to Play 4-Dimensional Cubic Line-4 Tic-Tac-Toe

Akira MIZUMA* and Nobuhiro TOMABECHI**

Abstract

This paper presents an original game called 4-dimensional cubic line-4 tic-tac-toe in which the player who has made a line-4 in 4-dimensional space will win the game. The purposes of the game are as follows; 1. invention of a new game joyful and useful for enhancement of the thinking ability, 2. educational usage to experience 4-dimensional space, 3. realization of artificial intelligences to win thinking games. The software to play the 4-dimensional cubic line-4 tic-tac-toe is developed, which is composed of 3 functional blocks as follows; 1. 3-dimensional computer graphics to display the 4-dimensional cubic line-4 tic-tac-toe, 2. supporting functions to recognize the 4-dimensional board, 3. search function of the best move.

Keywords: development, software, game, 4-dimensional, line-4, tic-tac-toe

1. ま え が き

思考ゲームをプレイするプログラムの研究は、人工知能研究上の多くの課題を含んでおり、近年、情報科学の重要な一分野として認められるようになった [1]-[3]。

ゲームプログラミング研究において一つの転機となった出来事が、1997年に起こった。当時のチェス世界チャンピオン Kasparov 氏をコンピュータ DeepBlue が破った事である。ゲームプログラミング研究はテレビゲームのソフトウェアのような、単に利潤を目的とするものではなく、人間の日常的な行動を探る重要な研究課題となっている [4]。

一方、4次元空間は、人間が知覚できない空間、現実には存在しない空間と一般には思われているようであるが、論理代数の分野では、多

次元空間が日常的に取り扱われる。もし、4次元空間を使用した思考ゲームがあれば、人工知能研究の観点から興味深いものであると同時に、4次元空間を認識するために、大きな教育的意義を有すると考えられる [5], [6]。

本研究では、4次元空間上に4連を作るという新しい思考ゲームを提案し、本ゲームをプレイするプログラムを開発する。

本ゲームは、次のような意義を有すると考えられる。① プレイして面白くかつ思考能力を育成する新しいゲームの創造、② 4次元空間を体験するという教育的試み、③ ゲームに勝利するための人工知能の実現。

本研究では、次の3つの機能を備えたプログラムを作成した。

- I. 4次元空間立体四目並べを表示する3DCG
- II. 4次元空間立体四目並べの認識を助ける補助機能
- III. 4次元空間立体四目並べの最善手着手

平成19年1月5日受理

* (株) ケーヒンエレクトロニクステクノロジー

** システム情報工学科・教授

2. 4次元空間立体四目並べ

本研究では、通常の3次元空間立体四目並べを4次元に拡張した4次元空間立体四目並べゲームを提案する。

通常の3次元空間立体四目並べを図1に示す。

2.1 4次元空間立体四目並べのルール

以下のように定める。

[ルール1] 盤は、 x 軸、 y 軸、 z 軸、 w 軸を有する。各軸は、4つの点を含む。したがって、盤の広さは、 $4 \times 4 \times 4 \times 4$ となる。ゲームフィールドは256となる。以下、盤上の1点を、 $P(x, y, z, w)$ と表す。

[ルール2] 先手と後手は交互に着手をする。

[ルール3] 先手と後手は、駒の色によって識別する。

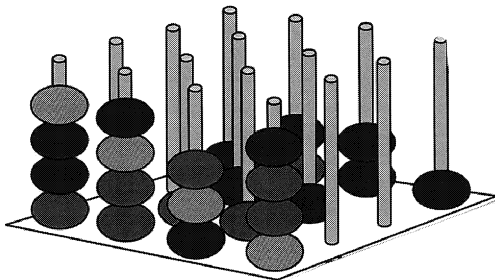


図1 通常の3次元空間立体四目並べ

[ルール4] x 軸、 y 軸方向の着手位置は、任意に選択できる。

[ルール5] z 軸方向の着手位置 $P(x, y, z, w)$ は、 $P(x, y, z-1, w)$ がすでに着手済みの場合のみ、着手可能とする。すなわち、 z 軸方向は、下から ($x=0$ から) 積み上げ式に着手する。

[ルール6] w 軸方向の着手位置 $P(x, y, z, w)$ は、 $P(x, y, z-1, w)$ 、および $P(x, y, z, w-1)$ がすでに着手済みの場合のみ、着手可能とする。すなわち、 w 軸方向も、下から ($w=0$ から) 積み上げ式に着手する。

[ルール7] 4次元空間上に4連ができれば勝利とする。

[ルール8] 4連を作成できずに盤面に着手できない場合、引き分けとする。

[注1] x, y, z 軸は、通常の3次元空間に対応する。 x, y, z だけを考えて、通常の3次元空間立体四目並べと同じになる。

[注2] ルール6に示すように、 w 軸は、(z 軸の場合と同様に) 下の座標から積み上げ式に着手することとする。こうすることによって、空間が w 軸方向に広がっていく様子が知覚されると期待される。

2.2 4次元空間の表現方法

本研究では、4次元空間を次のように表現する。

[表現1] 4次元空間は、 x 軸、 y 軸、 z 軸お

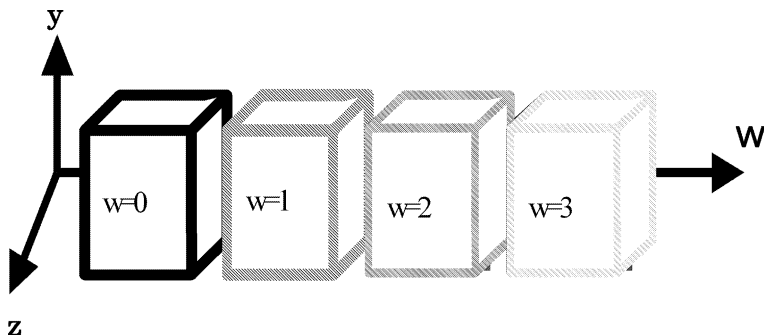


図2 w 軸方向に盤が順次表示されていく概念図

および w 軸の 4 軸を用いて表す。

〔表現 2〕 4 次元空間は、3 次元空間の立体盤を w 軸方向に 4 個並べることにより表示する。ただし、通常は、 $w=0$ の盤面のみを表示し、 $w=1\sim 3$ は、その空間への着手が可能になった時に表示を開始する。こう表示することによって、空間が w 軸方向に広がっていく様子が知覚されると考えられる。

〔表現 3〕 盤面を認識しやすくするため、立体盤全体を移動、回転する機能を設ける。

〔表現 4〕 局面が進むと、盤面の認識が困難になるので、3 連 4 連が作成できたとき、駒の色を変える。

図 2 に、表現 2 に基づいて、 w 軸方向に、盤が順次表示されていく様子を示す。

3. 4 次元空間立体四目並べ CG の作成

前述の 4 次元空間表現方法に従って、4 次元空間立体四目並べゲーム CG の作成を行った。

3.1 立体盤の表示

〔表示 1〕 4 次元空間は、3 次元の立体盤を w 軸方向に 4 個並べることにより表示する。初期状態では $w=0$ の盤面のみを表示する。 $w=1\sim 3$ の盤面は、 w 軸が着手可能になった場合に表示を開始する。

〔表示 2〕 先手は駒を赤色とし、後手は駒を青色とする。

3.2 盤面の認識補助機能の付加

4 次元空間立体四目並べの盤面認識を容易にするため、以下のような補助機能を付加する。

〔補助 1〕 先手の場合、着手候補のポールを

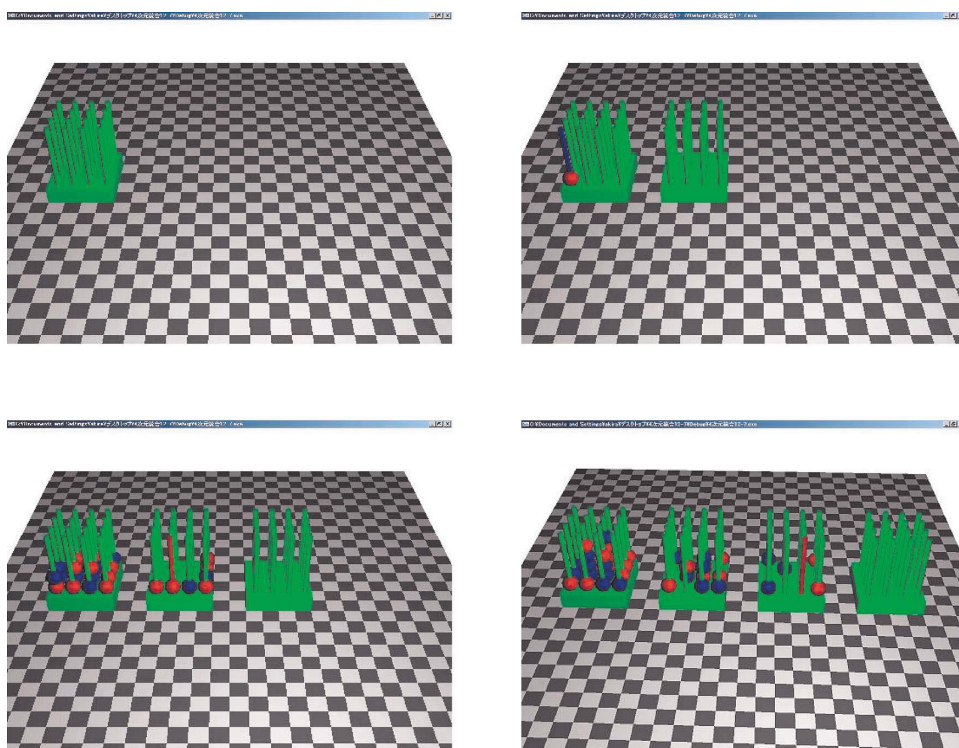


図 3 作成した 4 次元空間立体四目並べ CG

赤にし、後手の場合ボールの色を青にする。

〔補助2〕 着手候補を x 軸方向に、任意に移動する機能を付加する。

〔補助3〕 着手候補を w 軸方向に、任意に移動する機能を付加する。

〔補助4〕 x 軸, y 軸, あるいは, z 軸を中心に盤全体を回転する機能を付加する。

〔補助5〕 x 軸方向, y 軸方向, あるいは, z 軸方向に盤を移動する機能を付加する。

〔補助6〕 固定した視点から表示する機能を付加する。

〔補助7〕 3連, 4連が作成できた箇所の駒の色を変化させる。

〔補助8〕 ホップアップメニューを使用した, 対戦方法の表示をする。

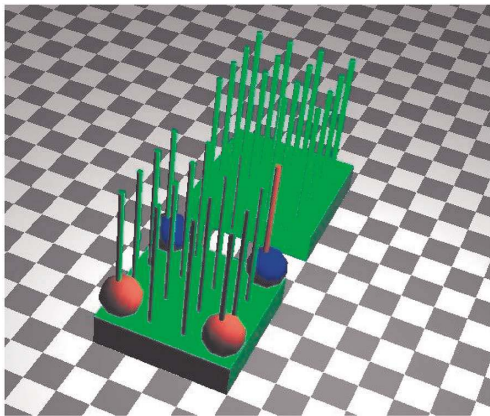


図4 z 軸を中心に回転した場合

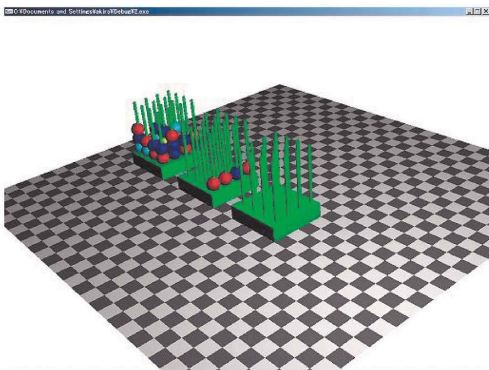


図5 盤を固定視点(斜め後方)から見た場合

〔補助9〕 DOS 画面に, 現在の局面, 2連の個数, 3連の個数, 着手済みの箇所, 及び, 空きの箇所を表示する。

〔補助10〕 DOS 画面に, 完成した4連の場所を表示する。

3.3 CGプログラムの作成・実行

4次元空間立体四目並べのグラフィック表示を行うプログラム, および, 補助機能プログラムを作成した。開発環境は, Microsoft Visual C++を使用した。また, グラフィックの表示には OpenGL の GLUT を利用した。

完成したプログラムは約 800 行となった。

図3に, 作成したCG画面を示している。図において, 左上は着手前, 右上は, 1手着手後を示

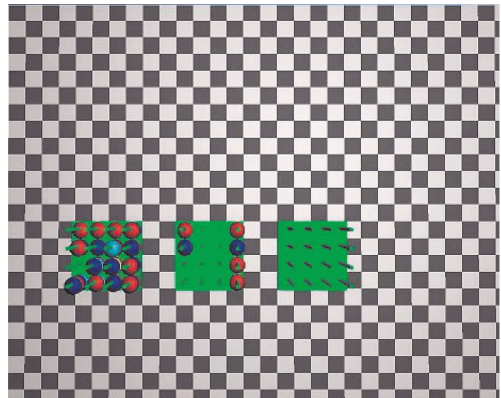


図6 盤を固定視点(真上)から見た場合

着手済み	0 16	空き	32 48
着手済み	0 21	空き	42 69
着手済み	0 12	空き	4 8
着手済み	25 12	空き	51 38
着手済み	5 21	空き	37 53
着手済み	21 25	空き	5 29
4連方	=	6	
60 61 62 63		124 125 126 127	188 189 190 191 252 253 254 255
56 57 58 59		120 121 122 123	184 185 186 187 248 249 250 251
52 53 54 55		116 117 118 119	180 181 182 183 244 245 246 247
48 49 50 51		112 113 114 115	176 177 178 179 240 241 242 243
44 45 46 47		108 109 110 111	172 173 174 175 236 237 238 239
40 41 42 43		104 105 106 107	168 169 170 171 232 233 234 235
36 37 38 39		100 101 102 103	164 165 166 167 228 229 230 231
32 33 34 35		96 97 98 99	160 161 162 163 224 225 226 227
○ 28 30 ×		82 83 84 85	156 157 158 159 220 221 222 223
24 × ○ 27		88 89 90 91	152 153 154 155 216 217 218 219
20 × ○ 23		84 85 86 87	148 149 150 151 212 213 214 215
× 17 18 ○		80 81 82 83	144 145 146 147 208 209 210 211
× 13 14 ○		76 77 78 79	140 141 142 143 204 205 206 207
08 ○ × 11		72 73 74 75	136 137 138 139 200 201 202 203
04 × ○ 07		68 69 70 71	132 133 134 135 196 197 198 199
× 01 02 ○		64 65 66 67	128 129 130 131 192 193 194 195

図7 DOS画面に局面の情報を表示

している。以下、左下、右下の順に、着手の進行に伴って、 $w=0$, $w=1$, $w=2$, $w=3$ と、 w 軸方向に盤が順次現れていく様子が分かる。

図4に[補助4]を用いて z 軸方向に回転した盤を、図5に[補助6]を用いて固定視点・斜め後方から見た盤、図6に[補助6]を用いて固定視点・真上から見た盤、図7に[補助9]を用いてDOS画面に連の個数を表示している様子を示している。

4. 最善手探索

思考ゲームプログラムの中核的機能である最善手探索機能の組み込みを行った。

4.1 探索アルゴリズム

本研究では、深さ優先探索を用いる。すなわち、先読みの深さが所定の深さに達したら、局面評価を行い、自分の手番では評価値が最大となるよう、相手の手番では評価値が最小となるよう着手するとして最善手を求める。

<探索アルゴリズム>

[手順1] プレイヤの決勝点がある場合、着手する。

[手順2] 相手プレイヤの決勝点がある場合、阻止する。

[手順3] プレイヤに追い勝ちがある場合、着手する。

[手順4] [手順1]～[手順3]まで該当箇所がないならば、着手可能な箇所をリストアップする。

[手順5] リストアップした中から、以下の条件の場所を探索する。

- ① プレイヤの4連になる個数。
- ② プレイヤの3連になる個数。相手プレイヤの3連になる個数。
- ③ プレイヤの2連になる個数。
- ④ 相手プレイヤの2連になる個数。

[手順6] 探索した結果により、該当箇所の多い順からリストアップの内容を並べ替え、リ

ストアップの内容を2/3に削減する。

[手順7] 設定した深さまで先読みを行い、評価値を計算する。評価が最大になった候補を最善手とし、着手する。

[手順8] プレイヤの4連があるならば、ゲームを終了する。

[手順9] 盤に着手する場所がなければ引き分けとし、ゲームを終了する。

[手順10] 次の手番がコンピュータなら[手順1]に戻り、違うなら思考アルゴリズムを終了する。

4.2 局面評価方法

局面評価を次のように行った。

[評価1] プレイヤの決勝点になる個数。

[評価2] 相手プレイヤの決勝点を阻止する個数。

[評価3] プレイヤの3連になる個数。

[評価4] 相手プレイヤの3連を阻止する個数。

[評価5] プレイヤの2連になる個数。

[評価6] 相手プレイヤの2連を阻止する個数。

[評価7] プレイヤの1連になる個数。

[評価8] 相手プレイヤの1連を阻止する個数。

[評価9] プレイヤの着手可能な個数。

[評価10] 相手プレイヤの着手不可能な個数。

以上の[評価1]～[評価10]の合計した値を

90	81	82	×	124	125	126	127	188	189	190	191	252	253	254	255	
58	57	58	59	120	121	122	123	184	185	186	187	248	249	250	251	
○	53	54	55	116	117	118	119	180	181	182	183	244	245	246	247	
48	49	50	51	112	113	114	115	176	177	178	179	240	241	242	243	
44	45	46	×	108	109	110	111	172	173	174	175	236	237	238	239	
40	×	42	×	104	105	106	107	168	169	170	171	232	233	234	235	
×	○	×	39	100	101	102	103	164	165	166	167	228	229	230	231	
32	33	34	35	96	97	98	99	160	161	162	163	224	225	226	227	
28	○	30	×	92	93	94	95	156	157	158	159	220	221	222	223	
24	○	26	○	88	89	90	91	152	153	154	155	216	217	218	219	
×	○	×	○	84	85	86	87	148	149	150	151	212	213	214	215	
○	17	×	19	80	81	82	83	144	145	146	147	208	209	210	211	
×	×	×	×	77	78	○	140	141	142	143	204	205	206	207		
○	○	10	×	72	×	74	75	136	137	138	139	200	201	202	203	
×	×	×	×	68	69	○	71	132	133	134	135	196	197	198	199	
×	○	1	×	64	65	66	○	128	129	130	131	192	193	194	195	

図8 実行画面

表 1 平均探索回数の比較

	深さ 4	深さ 5	深さ 6	深さ 7
深さ優先探索法	10,000	165,000	198,000	2,900,000
$\alpha\beta$ 探索法	13,000	200,000	1,900,000	21,000,000

表 2 平均探索時間 (秒) の比較

	深さ 4	深さ 5	深さ 6	深さ 7
深さ優先探索法	1	1	2	24
$\alpha\beta$ 探索法	1	1	7	67

局面の評価値とする。

4.3 プログラム作成・実行

前項で述べた探索アルゴリズムの手順に基づいて最善手探索を行うプログラムを作成した。開発環境は、Microsoft Visual C++を使用した。

完成したプログラムは全部で約 1,500 行となった。

図 8 に、実行時の画面を示す。

4.4 探索方法の比較

次の 2 つの方法についてプログラムを作成し、比較を行った。

方法 1. 深さ優先探索法 (ただし、評価値の高い着手候補についてのみ探索する)

方法 2. $\alpha\beta$ 探索法 (ただし、全着手候補について探索する)

比較した結果を表 1、表 2 に示す。

深さ優先探索法と、 $\alpha\beta$ 探索法を対戦させた結果、深さ優先探索法の先読みの深さが $\alpha\beta$ 探索法より深いならば、深さ優先探索法が勝利を収めた。

表 1、表 2 から、深さ優先探索法の方が $\alpha\beta$ 探索法よりも、探索効率が良いことが分かる。これは、深さ優先探索法が先読みの中で、着手候補の評価値を求め、値の高い順に並べ替えていることが原因と考えられる。

5. ま と め

本研究は、新しい思考ゲーム、4 次元空間立体四目並べの提案を行い、以下の機能を有するプログラムを作成した。

I. 4 次元空間立体四目並べを表示する 3DCG

II. 4 次元空間立体四目並べの認識を助ける補助機能

III. 4 次元空間立体四目並べの最善手着手

本研究では、4 次元方向を w 軸とし、盤面を w 軸方向に並べて 4 次元空間を表現した。4 次元空間の認識は、 w 軸方向に着手が可能になったとき盤を順次追加表示していくこと、および、認識補助機能を付加することで、ある程度、可能となることが分かった。

最善手探索については、4 次元空間を用いることで探索空間が膨大になるので、ゲームの数学的性格を解明し、最善手探索に組み込むことが必要となる。

次の段階では、4 次元空間を体験するという教育的試みを検証し、さらに 4 次元空間の広がりをもさらに認識できる機能を追加する予定である。

参 考 文 献

- [1] 飯田弘之, “ゲームプログラミングの発展と AI”, 情報処理, Vol. 37, No. 6, pp. 536-542, 1996-6.

- [2] 苫米地宣裕, “立体四目並べの数理”, 八戸工業大学情報システム工学研究所紀要, Vol. 11, pp. 1-4, 1999-3.
- [3] 小野寺優, 立体型四目並べプレイロボットシステムの構成に関する研究, 平成 16 年度八戸工業大学大学院修士論文。
- [4] 松原 仁, 竹内郁雄, ゲームプログラミング, 共立出版, 1998.
- [5] 水間 輝, 苫米地宣裕, “4 次元空間立体 4 目並べプレイプログラムの構想”, 計測自動制御学会東北支部第 216 回研究集会資料集, No. 216-2, 2004-6.
- [6] 水間 輝, 苫米地宣裕, “4 次元空間立体四目並べプレイプログラムの開発”, 平成 17 年度第 2 回情報処理学会東北支部研究集会資料, 2006-1.