INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS

ISSN(print): 2643-9840, ISSN(online): 2643-9875

Volume 05 Issue 10 October 2022

DOI: 10.47191/ijmra/v5-i10-15, Impact Factor: 6.261

Page No. 2670-2675

# Research and Design of Hexagonal Gaming System

**LI Yuqi[1], WEI Yiyang[2], LI Yongyuan[3]**

[1,2,3] School of Management Science & Technology, Anhui University of Finance & Economics, Bengbu Anhui，233030，China

**ABSTRACT:** In recent years, Hexagonal chess has gradually appeared in various machine gaming competitions, and is popular among the public because of its simple and fair game rules. In this system, a computer game program is studied and designed with Hexagonal Chess as the object of study. In order to further improve the search speed of the algorithm, the UCT algorithm is used in this paper. This algorithm not only improves the search time of Monte Carlo tree algorithm, but also improves the performance of the algorithm in space.

**KEY WORDS:** computer games; hexagonal chess; search algorithms; monte Carlo trees; UCT algorithms

## I. INTRODUCTION

With the rapid development of computers, people have become interested in playing chess on computers, and so chess games have gradually become known. However, as the world gradually transitioned from the information age to the age of artificial intelligence, people were no longer satisfied with the simple games for everyone that they used to play. At this point, people wanted to fight against more difficult computers, also known as AI games. AI games are a challenging topic in the field of artificial intelligence. It uses machine chess playing as the main research vehicle, imitating the thinking process of human chess playing. Key technology research and system design of checkers complete information game[1]. Since the 1950s, many famous scholars in the world have started to dabble in the field of AI games. John von Neumann, the father of computing, proposed the most famous minimum limit theorem for game tree search. John McCarthy, the founder of artificial intelligence, first introduced the concept of "artificial intelligence". Later, with "Deep Blue", "Super Deep Blue", "Hand Talk", "Chess Master" and the "AlphaGo" program that defeated the world Go champion, represent the success of today's AI game technology. The fundamental reason why AI games have evolved so rapidly is that scholars have focused on finding more efficient search algorithms, from the original minimax algorithms to artificial neural networks. The quest for more efficient search algorithms continues to be a quest for academics.

## II. BRIEF DESCRIPTION OF THE RULES OF HEXAGONAL CHESS

(1) Board representation

The Hex board is diamond shaped and usually consists of 10 x 10, 11 x 11, 14 x 14 hexagonal squares, while in the domestic Hex game the board basically consists of 11 x 11. The whole board has four borders, with symmetrical borders of the same colour. The board is shown in Figure 1.

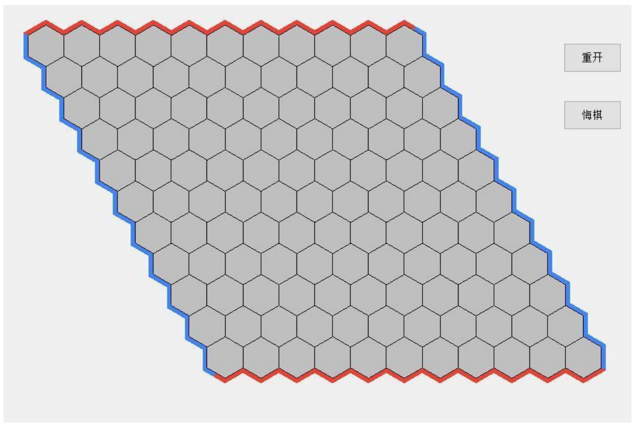

**Fig. 1 Board interface**

(2) Move rules

The rules of Hex Chess are relatively simple. (1) Before the game begins, both players select pieces of the corresponding colour, corresponding to the border. After each player has a piece of each colour, they choose their successive moves in turn. (2) Once the successive moves have been chosen, the game begins, with the first and second player taking turns to throw the pieces onto the board. Only one piece can be played at a time, i.e. occupying a hexagonal square to place a piece of your own colour. (3) At the end, the side that first forms a line with the border of the corresponding colour is the winner. As shown in diagram 2, on a board where the red piece is first to connect with the border, the red side wins.

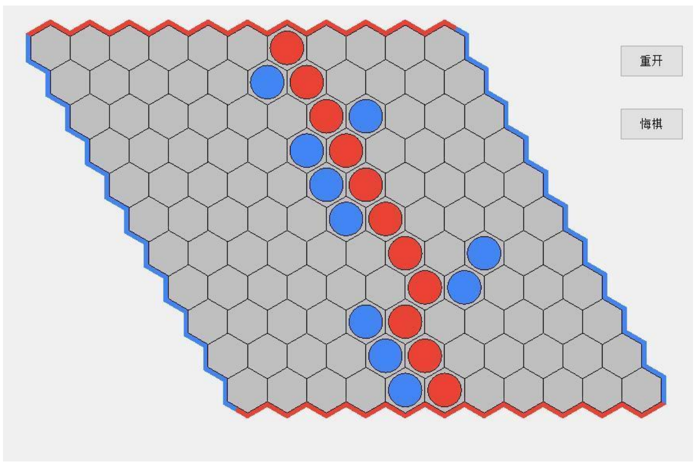

**Fig. 2 Red wins**

**III. STRUCTURAL DESIGN OF THE HEXAGONAL GAMING SYSTEM**

(1)Gaming system design

The computer game system consists mainly of the algorithm design and interface design of the combat platform. That is, the combination of the game process at the back end and the display interface at the front end. Firstly, the game design is the core of the entire game system. The core of game design is to get the best result of the game in a limited time. Its techniques consist mainly of search and valuation. Search is the process of constructing a tree based on initial conditions and extension rules and finding the nodes that match the target state by some algorithm[2]. And valuation is the assessment of how good or bad things are. This includes four specific parts, game tree search, node expansion, game simulation and valuation and node generation. Secondly, the interface design is relatively simple and specifically contains two parts, board initialisation and game rules. The specific design architecture is shown in Figure 3.

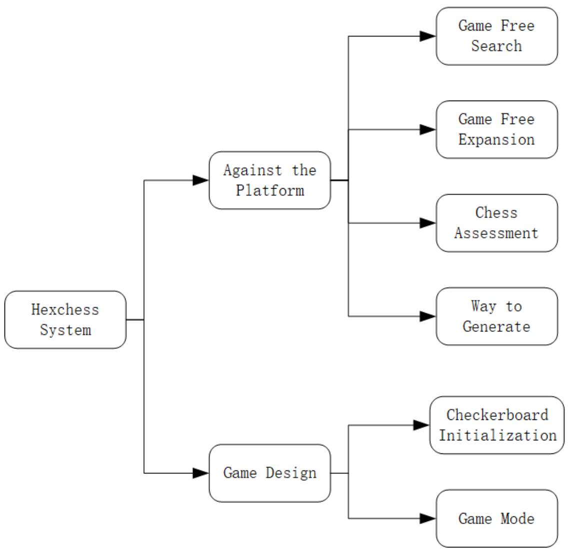

**Figure 3 Hexagonal gaming system architecture**

Once the structure has been designed, the question that needs to be considered is how the two players will play, i.e. the design of the game. According to the rules of Hexagonal Chess, the two players play alternating moves. The computer plays by using a specific search algorithm to find a specific move path, while the player places the pieces directly in the blank squares and the system automatically updates the board when either player has finished. The game flow is shown in Figure 5.

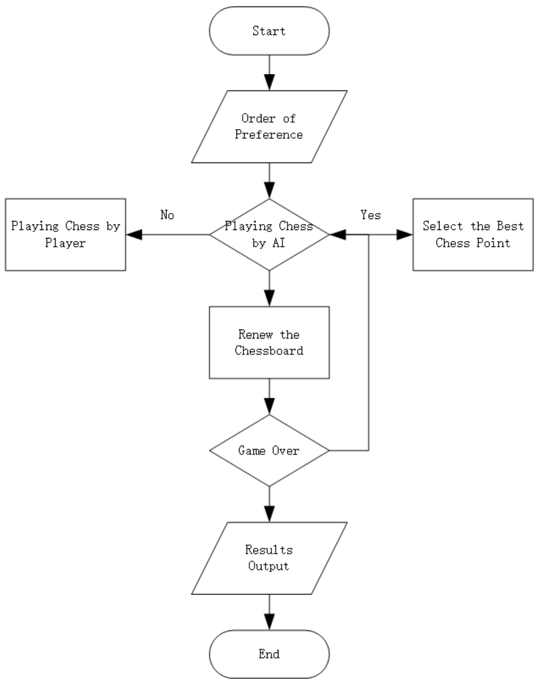

**Fig. 4 Flow chart of the Hexagonal Chess game**.

（2）Data structure design

The structural design includes the design of the board storage type, the board, the pieces and the implementation of the rules for the board. The definition of variables plays an important role in the writing of the program. Properly designed variables not only improve the readability of the program, but also provide greater clarity in the subsequent maintenance of the program[3]. The board uses an 11 × 11 two-dimensional array to hold the positions, where blank is denoted as -1, red as 0, and blue as 1. This design is convenient for counting the number of blank squares, red pieces, and blue pieces on the board, and enables better evaluation of the board.

```
Empty = -1
Red = 0
Blue = 1
```

## IV. DETAILED SYSTEM BACK-END DESIGN

（1）Search algorithms

Search is the core of AI chess, and there are many search algorithms that can be used in Hexagonal chess, such as the Extremely Small Value algorithm and the Negative Great Value algorithm in the Exhaustive Search algorithm, the Alpha-Beta algorithm and the PVS algorithm in the Crop Search algorithm, the Replacement Table heuristic and the History heuristic in the Heuristic Search algorithm and the Monte Carlo algorithm and the UCT algorithm in the Random Search algorithm. The UCT algorithm is simply the UCB formula applied to a Monte Carlo tree[4]. First of all, Monte Carlo trees are not considered an algorithm, but a generic term for a class of stochastic methods. The characteristic of this class of methods is that the approximate results are computed on random samples. As the number of samples increases, the probability that the result obtained is correct increases, but until the true result is obtained, there is no way to know if the result obtained so far is correct. The more you sample, the more you approximate the optimal solution. As opposed to the UCB formula, the UCB value can be understood as the expected value of the path, and the value is made up of two components. The UCB value of the node is equal to the sum of the average gain from the simulation of the node and the desire of the node to be simulated. Obviously, the fewer relative times a node is visited, the stronger the desire to not let go of a potentially valuable node. xj is the average gain, nj is the number of times the current node is visited, and n is the number of times the current node's parent node is visited. All Hexagonal Chess needs to do is to constantly adjust the variable parameters inside the UCB formula to get the best results.

$$I_j = \overline{x_j} + k\sqrt{\frac{2\ln n}{n_j}} \qquad (1)$$

（2）Algorithm implementation

The UCT algorithm has huge advantages in time and space compared to traditional search algorithms[5]. A study of point-grid chess gaming system based on UCT search algorithm. First of all, the move of the first piece, in fact, to a large extent, determines the winning or losing of the game. The optimal landing point for the first piece of the Hexagonal tempo is the position at coordinates (5,5) (initial coordinates 0,0), which is the centre of the board. It is also possible to use the UCT algorithm to simulate other moves, but this coordinate is relatively optimal.

Using the current position as the root node, the UCB formula is used to calculate the UCB value of each sub-node[6], and the sub-node with the greatest value is selected for exploration. There are three possibilities for exploration: unvisited nodes, leaf nodes and search depth reaching a fixed value. If the current node is not a leaf node or the search depth does not reach a fixed depth, the current node is used as the root node[7]. Continue to find and calculate the UCB values of all child nodes and repeat the above steps. Otherwise, simulate the matchmaking. The core code is shown below.

```
auto winrate = (uf->color_to_move() == colorAI) ? // whether to play for AI
(double)current->children[i]->cntWin / current->children[i]->cntTotal // winrate
1 - (double)current->children[i]->cntWin / current->children[i]->cntTotal;
new_score=ucb(winrate, current->cntTotal, current->children[i]->cntTotal);
//calculate the current node UCB value
if (max_score < new_score);//if max UCB value is less than current node UCB value
{max_score = new_score; select_index = i;}}//Current node UCB value recorded as maximum
current = current->children[select_index];//next selection from current node
```

If a leaf node is encountered or if the search depth reaches k, a simulated game is played against that leaf node and a Monte Carlo algorithm is used to obtain a random position to get the result of a win or a loss, and the value obtained during this calculation is updated to each level of ancestor node of that node. The evaluation function for this system is very simple, with a win returning 1 and a loss returning 0. The evaluation value is then used as the value for this search. The core code is shown below.

```
random_num = random(upper_limit);//Use the random function to obtain a random value
pos = alternative[random_num];//Use the random value to obtain the next position
```

However, when the UCT search algorithm terminates the simulation due to some factor, the system selects a node at the first level as the final strategy based on several criteria[8].

(1) The node with the largest UCB value.

(2) The node that has been searched the most times by the simulation.

(3) The node with the largest UCB value and searched the most times by simulation.

(4) The branch that has been simulated multiple times. The UCT algorithm simulation looks for the optimal branch, and when a branch is simulated multiple times and the UCB value is still high, it means that the branch is the optimal branch for simulation.

When a node is finally selected as the final decision, the system updates the values and search counts of all the nodes in the current situation and passes back the current win/loss result. The core code is shown below.

```
while (current)//until the root node jumps out of the loop
{if (colorAI == winner)//if the current position must win, add 1 to the search count and return to the parent node
current->cntWin++;
current->cntTotal++;//add 1 to the total number of searches
current = current->parent;}
```

A complete UCT algorithm is one that performs multiple UCT searches in a limited number and time. The paths obtained from each search are not necessarily the same, and the structure of the game tree changes. Because it is essentially a Monte Carlo tree algorithm, the final branch obtained will be the closest to the optimal branch if the number of searches or the time is increasing.

## V. CONCLUSIONS

This paper focuses on the research and design of the Hexagonal Chess gaming system. The Hexagonal gaming system consists of two main parts, the design of the board structure and the search algorithm. The structure of the board is based on a two-dimensional array to preserve the games. At the same time, the novel UCT algorithm is used in the search algorithm, which can control the search depth in a controlled search time or number of searches, and fundamentally improve the search speed of the game tree. Although the use of the UCT algorithm in this system can significantly improve Hex's gaming level, there are still some shortcomings, such as the winning rate of the first hand is greater than the winning rate of the second hand. Gradually improving the system in the future can improve and enhance the artificial intelligence of the system more effectively. In the future, neural networks can be combined with the Hex chess gaming system, which can far improve the computer's ability to play chess by way of self-learning through neural networks.

## ACKNOWLEDGMENT

This work was supported in part by the Natural Science Foundation of the Higher Education Institutions of 2021 National Project of Student Innovation and Entrepreneurship Training Programme under Grant No. 202110378171.

## REFERENCES

1) Yang, Zhoufeng. Key technology research and system design of checkers complete information gaming [D]. Shenyang University of Aeronautics and Astronautics,2018.
2) Wang Xin,Li Yuan,Wang Jingwen,Li Zhenyu. Research on Hex chess gaming system based on UCT algorithm[J]. Intelligent Computers and Applications,2020,10(06):116-119.
3) Gui Yiyong. Research on a game system for checkers[J]. Intelligent Computers and Applications,2020,10(04):32-34+39.

4) Jiao T. B. Research on robust communication of wireless optical and millimeter wave aggregation systems[D]. China University of Mining and Technology, 2021.
5) Zhu L.S.,Wang J.W.,Li Y. Research on point-grid chess gaming system based on UCT search algorithm[J]. Intelligent Computers and Applications,2021,11(02):129-131.
6) Zhang Yifang, Meng Kun. Research and analysis of UCT algorithm based on point grid chess[J]. Intelligent Computers and Applications,2020,10(04):27-31.
7) Ji Hui, Ding Zejun. Improvement of Monte Carlo tree search algorithm in two-player game problems[J]. Computer Science,2018,45(01):140-143.
8) Wang Wanwan. Research on UCT algorithm and computer game behavior of military chess robot[D]. Chongqing University of Technology,2019.