Design and Implementation of connect6 Intelligent Game System

Zengyu Cai¹, Chunfeng Du¹, Xiaoshuang Guo¹ and Jianwei Zhang²*

¹ School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan, 450002, China
² Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, Henan, 450002, China
* zhangjw@zzu.edu.cn

Abstract. As an important part of artificial intelligence, computer game is also considered as one of the most challenging research directions in the field of artificial intelligence. After understanding the research background and development status of the connect6 chess game, the paper analyzes the requirements and feasibility of the system according to relevant theories. This paper describes the intelligent game and the chess shape, and uses the "maximum-minimum" principle, evaluation function and other related technologies to design and implement the development of the connect6 chess game system. After testing the system's functions and intelligence, the system can achieve the desired results.

1. Introduction

With the development of science and technology, artificial intelligence not only is closely related to computer science, but also has a certain relationship with many disciplines such as medicine, psychology and education. At present, artificial intelligence has been booming in many fields such as robotics, machine learning, expert systems, and distributed artificial intelligence [1]. The intelligent chess game not only is a branch of the computer game, but also attracts a lot of attention as an iconic example in the field of artificial intelligence [2]. As ubiquitous online games emerge in an endless stream, chess games are constantly being expanded, such as flying chess, gobang, and chess. At the same time, as a simple and fair chess game, the connect6 chess game is also constantly moving towards intelligent development [3-4]. This paper designs and develops the connect6 intelligent game system, which mainly includes four modules: game search, evaluation function, travel generation and opening library [5]. Each module uses unique techniques and corresponding methods to solve problems that may be encountered in smart games.

2. System function

2.1. Functional Analysis

The overall requirement for system development in this paper is to realize the intelligent game six-game program with certain chess power on the basis of realizing the function of ordinary chess games. The system must also satisfy the human-machine intelligent game on the basis of satisfying the human-human game. The board representation module is used to display the connect6 board, and the board size of the connect6 board is 19×19. The walk generator is used to generate the legal behavior of the party...
or the other party in the current situation [6-7]. The evaluation function is used to estimate the situation of the move. Two valuation methods are used in this system, based on the valuation of "road" and the valuation based on "chess type" [8]. The search module mainly performs a search for the game tree and strategy. The basic structure model of the system is shown in Figure 1.

![Figure 1 System basic structure model](image)

### 2.2. Process Analysis

After the game starts, whether it is a mouse click or the computer drops the piece, it is necessary to determine whether the position of the piece is legal. The position of the falling piece will be re-selected, if the position of the falling piece is illegal. Otherwise, update the situation and calculate the next position. Calculate the score to find the chess point with the highest score. Then, the other party plays the chess piece. And the system judges whether it is a victory. The process of the connect6 intelligent game system is shown in Figure 2.

![Figure 2. system flow chart](image)

### 3. Related theories and techniques

#### 3.1. Evaluation Function

In the game, the players are set to \( MAX \) and \( MIN \) respectively, and the game assumes that \( MAX \) goes firstly. Due to the storage space, the number of chess pieces and the search time limit, the complete search of the graph cannot be performed [9]. A static estimation function \( f \) is defined to estimate the current state of the game. It is advantageous for \( MAX \) when \( f \) is a positive number, and \( MIN \) is advantageous when \( f \) is a negative number. The following is a step of giving a very small search by taking "Tic Tac Toe" as an example. The principle of “maximal or minimum” value search is shown in Figure 3.
(1) The extended tree node generates a game tree.
(2) Each leaf node is evaluated by the evaluation function \( f \), and the evaluation value of the upper node of the upper layer is obtained.
(3) Use the value of the intermediate node to get the estimate of the middle node of the previous layer. The estimate obtained by the leaf node passing \( A_{15} \), is the smallest \([10-11]\) in \( A_1 \) walking, and it is advantageous for MIN when the value of \( f \) is minus. Therefore, the \( A_1 \) walking MIN node chooses \( A_{15} \) to walk. The MIN of the same layer \( A_2 \) and \( A_3 \) also selects the minimum value to walk.
(4) Repeat step (3) to take the maximum value of its branch at the MAX layer. The MIN layer takes the minimum value of its branch \([11]\) and terminates the operation to the root node.

3.2. Evaluation Function

The evaluation function is a comprehensive evaluation of the current game. According to the current chess situation, Black and White are evaluated separately to judge the current situation of both sides of the game and provide the judgment standard for the search engine \([12]\). Among the connect6 chess, there are complex chess types. The evaluation of the game is generally done by local scanning. This paper considers that the local scan is not comprehensive enough, so the evaluation function based on the "chess type" global scanning method is adopted. The evaluation function formula based on the "chess type" global scan is as follows:

\[
\text{TotalValue} = \text{ExternalValue} + \text{LocalValue} \\
\text{LocalValue} = \text{MyValue} + \text{ThreatValue} + \text{BalanceValue}
\]

In equations (1) and (2), \( \text{TotalValue} \) represents the total valuation of the situation. \( \text{ExternalValue} \) represents an estimate outside of the local. \( \text{LocalValue} \) represents a local estimate. \( \text{MyValue} \) represents its own local valuation. \( \text{ThreatValue} \) represents the other party's local estimate. \( \text{BalanceValue} \) represents the threshold at which the local estimate is unbalanced.

The evaluation function includes a \( \text{LocalValue} \) and an \( \text{ExternalValue} \). The external estimate can capture the type and number of moves that exist outside of the local. The external range not is scanned from the checkerboard \((0,0)\) position, but can contain all the pieces of the board and the minimum range outside the partial drop. The local estimate includes the own value, the opponent threat value, and the balance value.

\[
\text{MyValue} = \text{MyFirstValue} + \text{MySecondValue}
\]

\[
\text{ThreatValue} = \text{ThreatFirstValue} + \text{ThreatSecondValue}
\]

In the formulas (3) and (4), \( \text{MyFirstValue} \) represents the valuation of the first piece of the party. \( \text{MySecondValue} \) represents the valuation of the second piece of the party. \( \text{ThreatFirstValue} \) represents the threat value that simulates the opponent's first piece. \( \text{ThreatSecondValue} \) represents the threat value that simulates the opponent's second piece.

Equation 3 contains a chess analysis of the first piece of the game and a chess analysis of the second piece. And then the sum of the two is used as the own estimate. Equation 4 contains the threat value of...
the opponent's first piece and the second piece at the same board position. When the difference between
the MyValue and the other ThreatValue value is large, an imbalance adjustment is required. The
adjustment formula is as shown in Equation 5.

\[ \text{BalanceValue} = W1 + W2 \]  
(5)

\[ W1 = \begin{cases} 
\frac{\text{min1}}{2} & k1 \geq 2.5 \\
0 & k1 < 2.5 
\end{cases} \]  
(5)

\[ W2 = \begin{cases} 
\frac{\text{min2}}{2} & k2 \geq 2.5 \\
0 & k2 < 2.5 
\end{cases} \]  
(6)

In formulas (5), (6) and (7), \( W1 \) represents the difference between the valuation of the first piece of
the opponent and the valuation of the opponent's first piece. \( W2 \) represents the difference between the
valuation of our second pawn and the valuation of the opponent's second pawn. \( K1 \) represents the
imbalance factor of the first piece estimate, \( K1 = \text{MyFirstValue} / \text{ThreatFirstValue} \). \( K2 \) represents the
imbalance factor of the second piece valuation, \( K2 = \text{MySecondValue} / \text{ThreatSecondValue} \). \( \text{min1} \) represents the smaller value of the first piece of the party and the opponent, and \( \text{min2} \) represents the
smaller value of the first piece of the party and the other party. Adjust the imbalance of the two falling
pieces by formulas (6) and (7). When the ratio of the opponent's pawn to the opponent's threat value is
greater than 2.5, it is considered to be unbalanced. And then the imbalance is expanded to reduce the
value of BalanceValue. When it is less than 2.5, it is considered that no imbalance occurs and the
imbalance value is not changed. Through the imbalance adjustment, the way to balance your own and
the other party's threats would be chosen to help your own offense and defensive moves.

4. Function test

4.1. Basic Function Test

After the program is started, a human-machine battle or a human-person battle would be selected. At
the same time, you can choose to open the background music. If you choose to play Black, the first piece
is a black piece. When you choose to play white, the second piece and the third piece will be white
pieces. The system includes connect6 rules and system environment related introduction. After testing,
basic functions can be achieved.

(a) Repent before chess

(b) After repenting

Figure 4. White repentance chess
Retreat and repentance pieces can be selected during the game. Retreated as shown in Figure 4. Repentance is shown in Figure 5. In Figure 4, since the white color pieces are played two steps at a time, the chess is also retreated at the same time. Figure 5 is a black chess piece. Picture (c) stands for a picture of a chess piece. Picture (d) represents a schematic diagram of repenting two pieces.

4.2. Basic Function Test
The system intelligence test is mainly aimed at the human-machine battle game mode. According to the chess player’s falling chess pieces on the chessboard, the computer obtains the best move according to the evaluation function and the game search calculation and analysis, so as to realize the connect6 intelligent game.

Select the player to hold the black color piece firstly, and the computer as the back hand to hold the white color piece. After the chess piece is dropped, the best way to walk is based on the evaluation function evaluation and the game search. There are two cases tested. The chess player's opening game is shown in Figure 6.

When the computer first drops the black color piece, the player then chooses to drop the white color piece. After the chess pieces are dropped, the chess pieces that fall by the players can form "living two", "single", "sleeping two" three kinds of chess shapes. In any case, the computer can form a "live three" chess shape in the next game. After testing and analysis, the computer movement is the highest score. The computer opening game is shown in Figure 7.

5. Conclusions
The system designed in this paper has been able to initially complete the man-machine battle, the human-person battle, and the judgment of winning or losing. After testing, the system is stable and can be
implemented. In addition to the basic functions, the system also adds the repentance function to make
the game more user-friendly. It adds the function of saving the game and opening the game to facilitate
the player to record the battle situation. At the same time, it also adds game help to enable the player to
quickly understand the game rules and environment. In order to enhance the entertainment of the game,
the system added background music. All in all, the connect6 intelligent game system designed and
developed in this paper can complete the functions required by the intelligent game system. The research
results can contribute to the development of the connect6 chess game in theory and practical application.

Acknowledgements
This work is supported by National Natural Science Foundation of China No.61672471, Key
Technologies R & D Program of He’nan Province (No.172102210059 and No.172102210060), He’nan Province University science and technology innovation team(No.18IRTSTHN012) and Plan For
Scientific Innovation Talent of Henan Province (No.184200510010), Special Program for Fundamental Research Business Fees of Universities Affiliated to Henan Province(16601000020).

References
[1] Shang W, An P, Wan M, et al. (2017) Research and development overview of intrusion detection
technology in industrial control system[J]. Application Research of Computers, 34(2): 328-333.
[2] Hannaway D B. (2010) Research and Application of Cultivation-Simulation-Optimization
Decision Making System for Rapeseed (Brassica napus L.) [C]// Computer and Computing
Technologies in Agriculture Iv-- Ifip Tc 12 Conference,ccta 2010 Selected. pp. 441-456.
[3] Guan-Jun X U. (2012) Research on application and optimization of malfunction alert for
computer monitoring system in cascaded control center of Lancang River[J]. Mechanical &
Electrical Technique of Hydropower Station. pp. 34(2): 328-333.
[4] Chen X, Chen Z, Li H, et al. (2016) Research and implementation of the spectrum analysis system
based on FPGA[J]. Electronic Measurement Technology.
[5] Wang X L. (2014) Search optimization of chess game in Lianzhu mode [D]. Anhui University.
[6] Lin, Z D. (2014) Intercomparison of the impacts of four summer teleconnections over Eurasia on
East Asian rainfall[J]. Advances in Atmospheric Sciences, pp. 31(6):1366-1376.
[7] Yuan C. (2016) Improvement of Chinese Chess Search Algorithm [D]. Donghua University.
[8] Huang J P, Zhang D, Miao H. (2009) Research and Implementation of Six-Player Intelligent
Game System[J]. Computer Knowledge and Technology, pp. 5(25): 7198-7200.
[9] Song X L. (2012) Research and Implementation of Chinese Chess Game Tree Search
Algorithm[D]. Shenyang University of Technology.
[10] Jiang W J. (2013) Research and application of artificial intelligence in online games [D]. Shanghai
Jiaotong University.
[11] Barriga N A, Stanescu M, Buro M. (2017) Game Tree Search Based on Non-Deterministic Action
Scripts in Real-Time Strategy Games[J]. IEEE Transactions on Computational Intelligence &
AI in Games, PP (99):1-1.
[12] Takeuchi S, Kaneko T, Yamaguchi K. (2011) Evaluation of Game Tree Search Methods by Game
Records[J]. IEEE Transactions on Computational Intelligence & AI in Games, pp. 2(4):288-302.