Verify Performance

1. Simulation Environment
    - Hardware
        - 8 GB 2133 MHz LPDDR3
        - 3.1 GHz Intel Core i5 × 2
    - Software
        - Xcode version 10.1
2. Main parts of the Algorithm
    - *double WithAI();*
    - *double WithoutAI();*

Simulate with real player

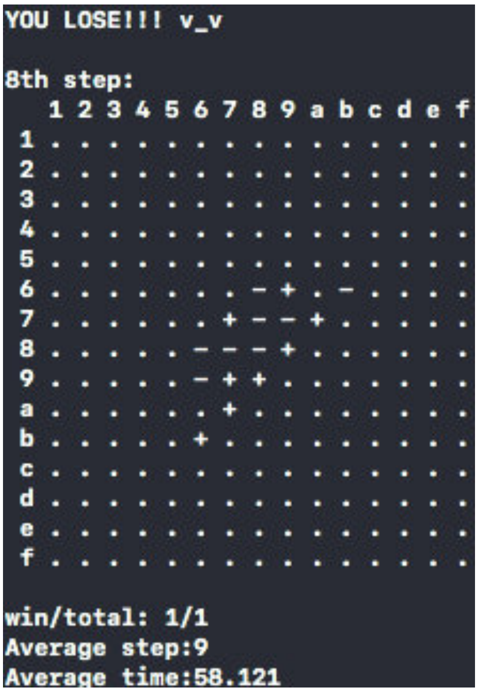


Figure1. Against real player with AI

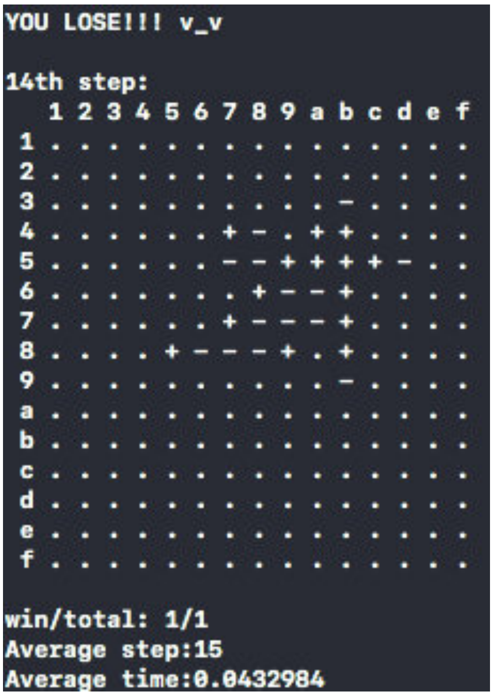


Figure2. Against real player without AI

Simulate with pseudo player

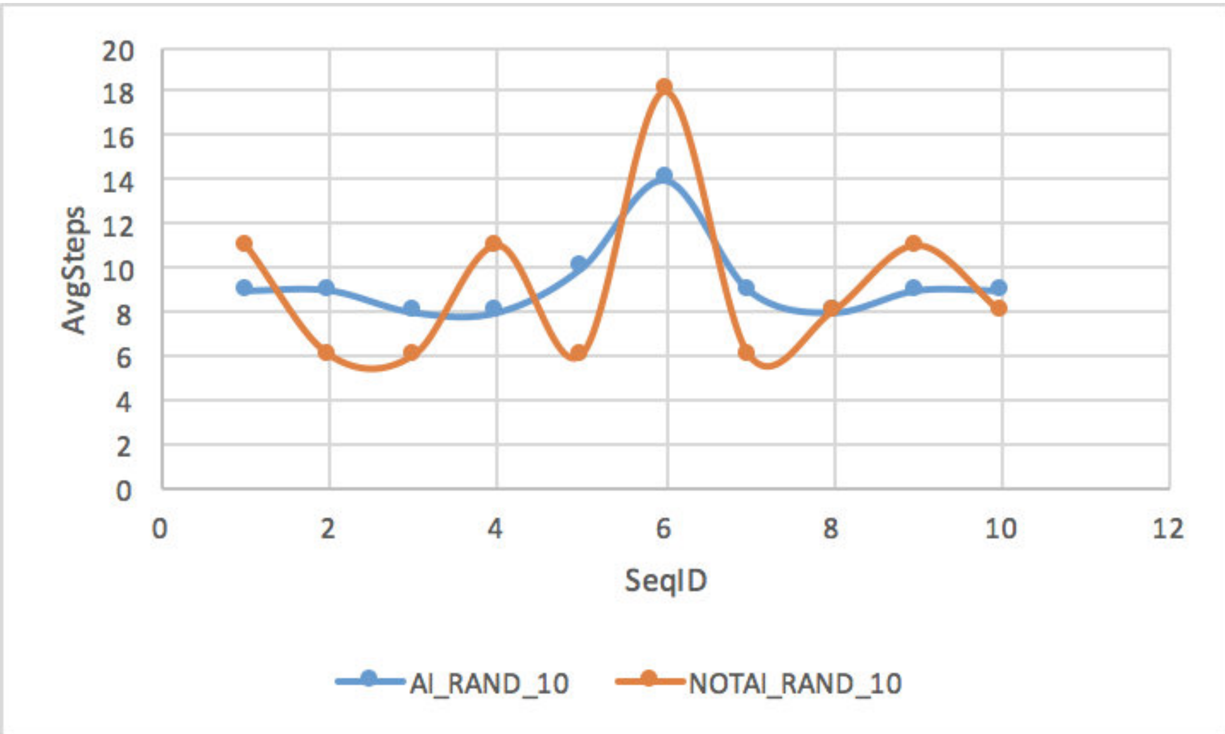


Figure3. Steps needed against different Algorithm

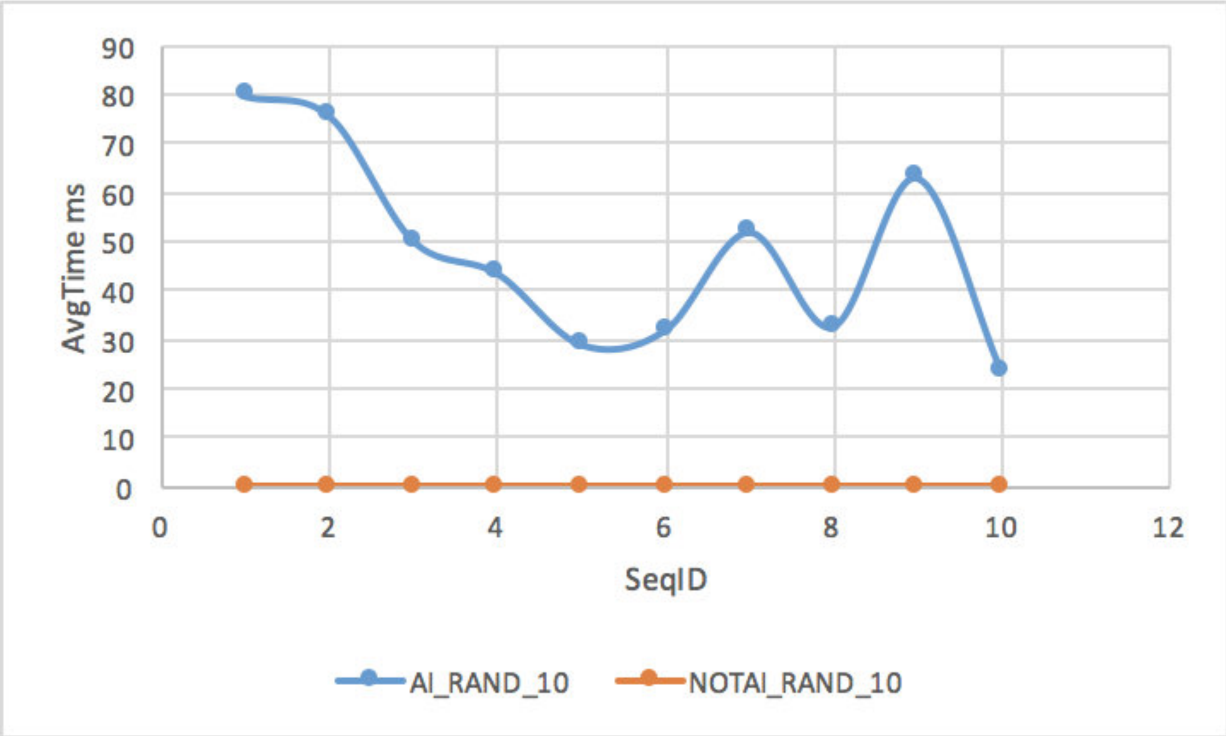


Figure4. Time needed against different Algorithm

Simulate with different SeachDeep

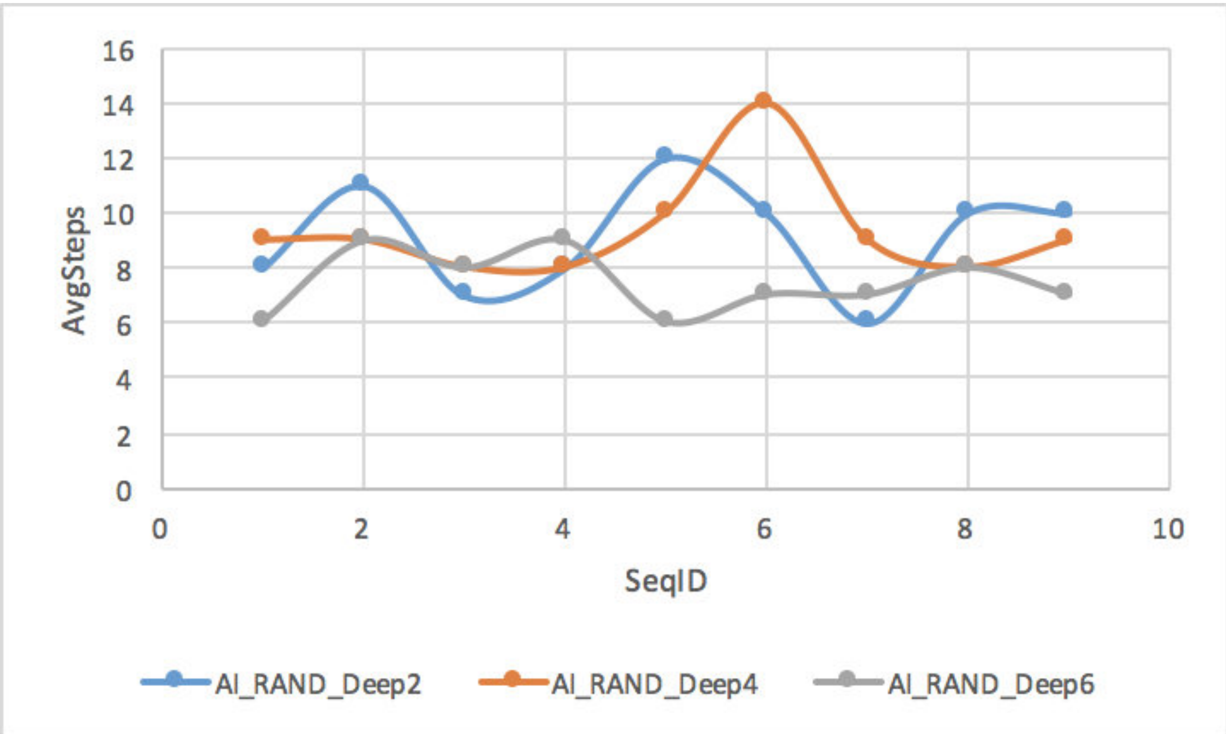


Figure5. Setps needed against different search Deep

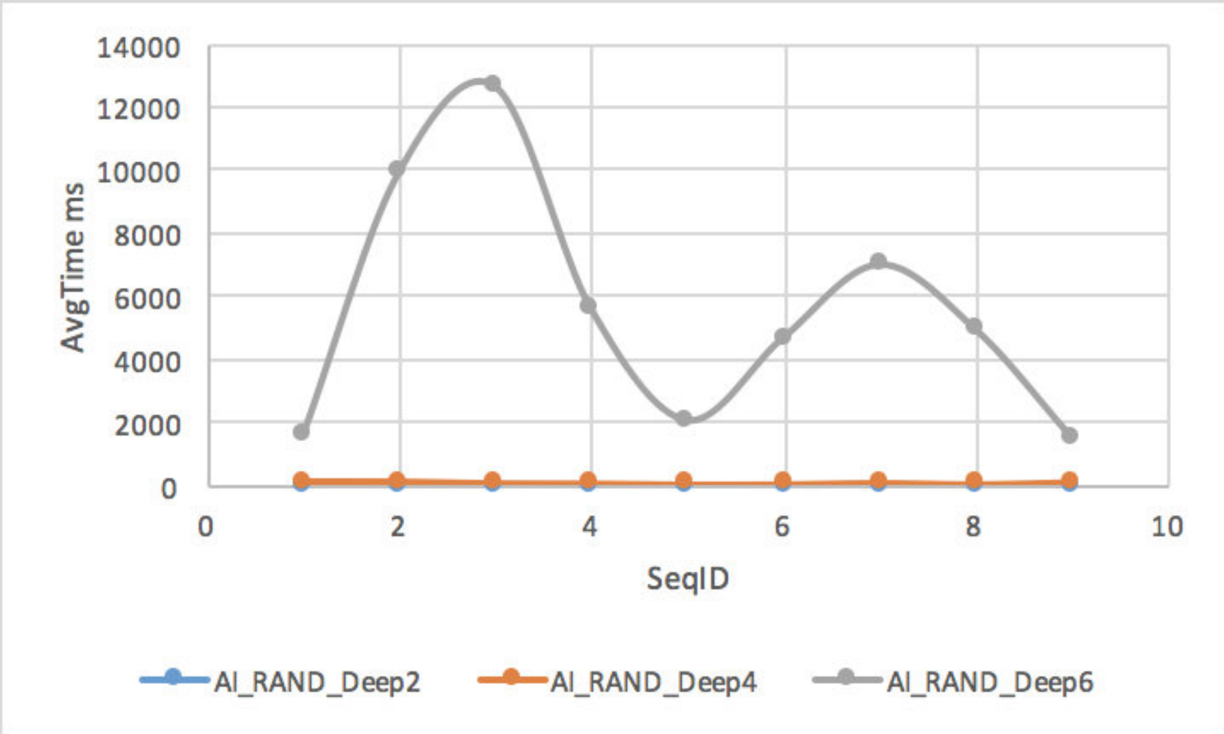


Figure6.a. Time needed against different search Deep

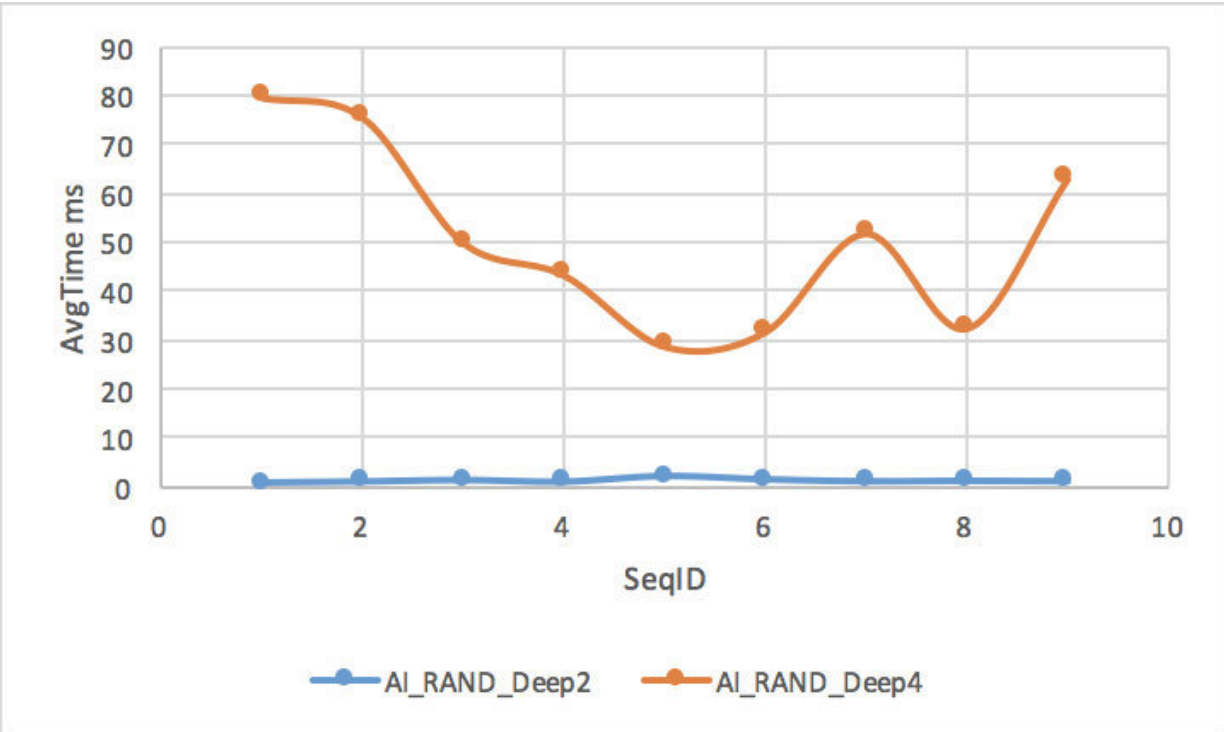


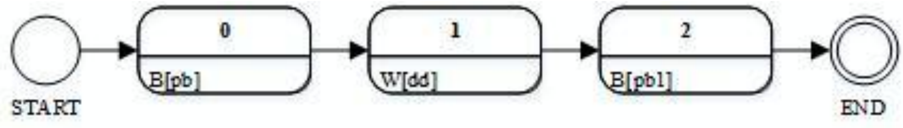
Figure6.b. Time needed against different search Deep

Checkerboard Pattern Matching

Smart Game Format

1. SGF is the abbreviation of 'Smart Game Format'. The file format is designed to store game records of board games for two players. It's a text only, tree based format. Therefore games stored in this format can easily be emailed, posted or processed with text-based tools. The main purposes of SGF are to store records of played games and to provide features for storing annotated and analyzed games (e.g. board markup, variations).
2. Example
   - (;GM[1] SZ[19] HA[0] KM[3又3/4子] AP[MultiGo:4.4.4] DT[2018-02-06] PB[柯洁]BR[9P] PW[朴廷桓]WR[9P] GN[柯朴十番棋第1局] PC[北京] RE[B+R] ;B[pd] ;W[dd])
   - (;B[pd] ;W[dd])

Deterministic finite Automaton

1. Is a finite-state machine that accepts or rejects strings of symbols and only produces a unique computation (or run) of the automaton for each input string.
2. Use DFA to denote the checkerboard.
3. 

Boyer-Moore

1. The Boyer–Moore string-search algorithm is an efficient string-searching algorithm that is the standard benchmark for practical string-search literature. The algorithm preprocesses the string being searched for (the pattern), but not the string being searched in (the text). It is thus well-suited for applications in which the pattern is much shorter than the text or where it persists across multiple searches. The Boyer-Moore algorithm uses information gathered during the preprocess step to skip sections of the text, resulting in a lower constant factor than many other string search algorithms. In general, the algorithm runs faster as the pattern length increases. The key features of the algorithm are to match on the tail of the pattern rather than the head, and to skip along the text in jumps of multiple characters rather than searching every single character in the text.

Pseudo Algorithm

1. Before start the algorithm
   - Transform all the pre-known checkerboard into DFA separately denote as D_p
   - Transform the current checkerboard into DFA denote as D_c
   - Transform all the pre-known checkerboard into String which can be used by Boyer-Moore algorithm separately denote as S_p ,like *K[1]K[2]K[3]...K[N]*
   - Transform the current checkerboard into String denote as S_c ,like *C[1]C[2]..C[M]*
2. The pseudo Algorithm

Algorithm 1 Pseudo Algoithm for Checkerboard Pattern Match

Input: The DFA list for pre-known checkerboard, D_p; The strings for pre-know checkerboard, S_p; The current checkerboard;

Output: The best choice for the next step;

1: Get the DFA from the current checkerboard denote D_c;
2: Get the string from the current checkerboard denote S_c;
3: Get the match coefficients for each DFA from S_p against S_c;
4: Get the best position for next step by compare the selected DFA from D_p and D_c;
5: **return** The best position for next step;

Bibliographical Review

Turn-Based

1. The Gobang game is turn-based, which means that the defender takes a step then the attacker takes a step.

Perfect Information Game

1. In game theory, a sequential game has **perfect information** if each player, when making any decision, is perfectly informed of all the events that have previously occurred, including the "initialization event" of the game (e.g. the starting hands of each player in a card game).

System Constituent Parts

1. The knowledge representation part includes the representation and implementation of the game, the knowledge base such as the opening game and ending game libraries, as well as the data structure of the entire game system, which is the basis of the game system. Efficient storage and presentation technology is its main research direction, such as the "bitboards" representation technology of chess. Since chess is 8X8, the basic unit of storage and processing is "bit". When processed by bit, the processing efficiency can be multiplied.
2. The move sequences generation function finds legal moves according to the rules of the game, and the travel generation function can also combine the corresponding chess knowledge to further filter out more reasonable moves, and optimize the order of the move sequences to improve Game tree search efficiency. The game design of different kinds of games, the design realization of the move sequences generation function is very different.
3. The evaluating function is used to evaluate the situation and give appropriate value to reflect the pros and cons of the situation. In most cases, the evaluation value uses an integer value. In general, the valuation function is not directly used to evaluate the current situation. However, according to the development of the game tree, the situation is formed after the assumption of the several steps of the two sides, and the evaluating function is called to evaluate the situation. Since the game tree of many board games is extremely large, it is impossible to fully expand, and it can only be expanded to a certain depth, and then evaluated using the evaluation function. The speed and accuracy of the evaluation have a direct influence on the search efficiency of the game tree. Since the speed and accuracy are contradictory to a certain extent, the impact on the system is also very large, so the design realization of the valuation function is a key point of the high-level game system.
4. Game tree search is the backbone of the game system. The solution of the whole tree is

based on the minimax theorem. In the following, By searching for the entire game tree (traversal) to find the best way to play against the game in the current situation, called a game tree search. The game tree searches for the leaf node (the last node of the expanded game tree) and uses the valuation function to obtain the situation estimate. The process of the game tree search is the computer 'thinking' process, and the search results reflect The intelligence of the game system, that is, the level of chess in the game system. For humans, the deeper the thinking, the more accurate the judgment. The computer game tree search is the same. The more the game tree search layer, the more accurate and reliable the search results. The level of computer chess is mainly determined by the depth of the game tree search. In theory, the search without time limit can find the best solution of the game, and the actual game should determine the move within a reasonable time. Therefore, the depth of the game tree search must be limited by the specific hardware and software environment and time. For the relationship between search depth and chess power, taking chess as an example, Western scholars have a more consistent view that the 14-layer depth (7 steps for each side) game tree search is roughly equivalent to 2800 points, which is a very small number of top professional The score that the player can reach. The professional chess player's calculation depth is generally only a dozen layers (that is, the outcome after ten steps is calculated to determine the current move). But unlike the computer, the computer can accurately determine the situation after more than a dozen steps. Human calculations are extremely selective and there are many missing calculations. Since the depth of search is a bottleneck restricting the chess power of the computer, the research of search algorithms has become the main research field of computer chess in the past half-century.

Bibliography

[1] Li Xiali, Wu Licheng, Li Yongji. Research on the Computer Game of Tibetan "Jiu" Chess Based on Chess Type[J]. Journal of Intelligent Systems, 2018, 13(04): 577-583.

[2]Zhang Jiajia. Design and Implementation of the gobang Platform[J].Computer Knowledge and Technology,2012,8(22):5409-5411.

[3]You Guirong. Construction Strategy of Valuation Function in Game Search Algorithm[J]. Journal of Fujian Commercial College, 2005(06): 82-85+88.

[4] Li Shuting, Gao Yan. Design and Implementation of gobang Based on Android[J].Software Guide, 2016, 15(04): 98-99.

[5] Zhang Mingliang. A new game tree search algorithm and its application[D]. Suzhou University, 2007.

[6]Cheng Yu, Lei Xiaofeng. Research and Improvement of Alpha-Beta Search Algorithm in Gobang[J]. Computer Engineering, 2012, 38(17): 186-188.

[7] Dong Hongan, Jiang Xiuying. The Core Algorithm of Intelligent Gobang Game Program[J]. Journal of Zaozhuang University, 2005(02): 61-65.

[8]WANG Changfei, CAI Qiang, LI Haisheng.Design and Implementation of Intelligent Gobang Algorithm[J].Journal of System Simulation,2009,21(04):1051-1054.

[9]Zhang Mingliang, Wu Jun, Li Fanchang. Design of evaluation function of the game system of gobang machine[J]. Journal of Computer Applications, 2012, 32(07): 1969-1972+1990.

/*

[1]李霞丽,吴立成,李永集.基于棋型的藏族“久”棋计算机博弈研究[J].智能系统学报,2018,13(04):577-583.

[2]张佳佳.五子棋对战平台的设计与实现[J].电脑知识与技术,2012,8(22):5409-5411.

[3]游贵荣.游戏搜索算法中估价函数的构造策略[J].福建商业高等专科学校学报,2005(06):82-85+88.

[4]李舒婷,高燕.基于Android的五子棋游戏设计与实现[J].软件导刊,2016,15(04):98-99.

[5]张明亮. 一种新的博弈树搜索算法及其应用研究[D].苏州大学,2007.

[6]程宇,雷小锋.五子棋中Alpha-Beta搜索算法的研究与改进[J].计算机工程,2012,38(17):186-188.

[7]董红安,蒋秀英.智能五子棋博弈程序的核心算法[J].枣庄学院学报,2005(02):61-65.

[8]王长飞,蔡强,李海生.智能五子棋算法的设计实现[J].系统仿真学报,2009,21(04):1051-1054.

[9]张明亮,吴俊,李凡长.五子棋机器博弈系统评估函数的设计[J].计算机应用,2012,32(07):1969-1972+1990.

*/