2023 4th International Conference for Emerging Technology (INCET)
Belgaum, India. May 26-28, 2023

New Age Chess Engine

[1]Preethi
School of CSE & IS
Presidency University
Bengaluru, India
preethisrivathsa@gmail.com

[2] Mohammed Mujeer Ulla
School of CSE & IS
Presidency University
Bengaluru, India
mujerroshan@gmail.com

[3]Sapna R
School of CSE & IS
Presidency University
Bengaluru, India
sapna.aradya@gmail.com

[4]Mohan K.G
Department of CSE
GITAM University
Bengaluru, India
mkabadi@gitam.edu

2023 4th International Conference for Emerging Technology (INCET) | 979-8-3503-3575-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/INCET57972.2023.10170291

***Abstract*— A chess engine, used in computer chess, is a programme that analyses chess or chess-variant positions and produces a move or series of moves that it deems to be the strongest. The back end of a chess engine often has a command-line interface without any graphics or windowing. Engines are typically used in conjunction with a front end, a windowed Graphical User Interface (GUI) that allows for keyboard, mouse, or touchscreen interaction, such as a chessboard or winboard. It also enables the user to play against various engines without having to become familiar with each one's unique user interface. The availability of numerous chess engines for smartphones and tablets has increased their accessibility. This work aims to develop an artificial intelligence capable of amateur chess play.**

Keywords- Artificial Intelligence (AI), Chess Engine, GUI

I. INTRODUCTION

A chess engine, used in computer chess, is a programme that analyses chess or chess variant positions and produces a move or selection of moves that it deems to be the strongest. The back end of a chess engine often has a command-line interface without any graphics or windowing. Engines are typically used in conjunction with a front end, a windowed graphical user interface that allows for keyboard, mouse, or touchscreen interaction, like Pygame or WinBoard. A two-player board game is called chess. Regarded distinguish it from kindred games like xiangqi and shogi, it is occasionally referred to as Western chess or international chess.

The development of a chess engine's hardware and graphical user interface are also described in this paper [1], in addition to the software aspects of the engines under discussion. It is regarded as an efficient technique for running the engine because the majority of engines employ a variation of the MinMax search tree. Then, each variant is enhanced utilising fresh searching or trimming methods. To aid the machine in performing the calculations more quickly, the hardware is updated as necessary.

With the use of several tools, the GUI is enhanced.

The user feels more and more engaged when using an interactive GUI. Closer to playing the game itself, yet at the portability is convenient. 2D and 3D graphics, as well as Additional media, such as background music, is used to enhance the game has a slight edge over the original and some traditional game

In order to increase the number of chess players, more and more variations of the game itself are presently being created. Some of the most well-known variations of the game that are gaining popularity among players are the Surakarta chess and the hexagonal chess. The chessmen in these engines move differently from how they do in traditional chess, but they operate on the same principle. To give players the impression that they are playing a completely different game, the board is also built differently.

Each engine is built with the idea that it will function on a typical computer without the need for additional processing units, unless extremely complex game trees or extremely high definition images are added to them. In addition, many games are developed solely for network play between two connected client machines, allowing users to compete against each other or the computer.

Some of the basic definitions used in Chess game are: -

1. MinMax Algorithm [2]: It is a decision-based rule that is applied to a game tree to minimise loss in the worst situation. It was initially developed for two-player zerosum games like chess, but it has since taken on many different variations that allow it to function in more complicated chess variations as well as other games.
2. Pruning [3]: When the lowest node or the maximum node in a game tree has already reached the value that is necessary and further travelling is not necessary, that specific branch of the tree is pruned or disconnected from the traverse.
3. Surakarta Chess [4]: The square chessboard in this Chinese variation of war chess features six horizontals and six vertical lines. Chess pieces are placed on the 36 places where horizontal and vertical lines cross. Eight arc lines connect each of these lines. There are twelve chess pieces for each side to use throughout the game.
4. Hexagonal Chess [5]: The board for hexagonal chess has 91 cells in the shape of a hexagon. The most popular version is Gliski's variation, a symmetric hexagonal board. Depending on the creator, the chessmen move and follow different rules.

The AI community should take a moment to consider the significance of chess in the expanding goals of AI, review the contributions made so far, and predict what can be expected in the future now that computers have advanced to the grandmaster level and are competing for the World Championship. Despite the widespread interest in chess among computer scientists and the tremendous advancements made in the last 20 years, the AI community doesn't seem to have a great deal of respect for the subject. On the one hand, this is the fruit of success (brute force works, so why learn anything else), but it is also the outcome of the chess community's obsession with performance above everything else [6]. Additionally, chess has shown to be too difficult for several AI strategies that have been applied to it.

The growing interest in intelligent buildings, as the rest of the paper is organized as follows. Section II gives a summary of the related to chess engine work. Section III design of chess engine using AI, Section IV presents the implementation and result analysis of chess engine. Finally, Section V concludes on proposed work.

II. RELATED WORK

This chess engine, which was created twice because it originally failed to beat Garry Kasparov in 1996 but then succeeded in doing so in 1997, was made with help from [7]. It was the first chess computer to triumph over a grandmaster in competitive play. A single chip move generator is employed. Deep Thought 1 and Deep Thought 2 were stepping stones between the Deep Blue and the Deep Thought, however Deep Thought 2 was played at numerous public events between 1991 and 1995. Kasparov's initial defeat of Deep Blue I made it evident that it had some issues that needed to be fixed. As a result, a new chip was created that increased the number of features from 6400 to 8000. In essence, the chip generates the best move set it can by combining a software search in C language with a hardware search on the silicon chip. It uses the null window alpha beta method and is a very efficient search even without pruning.

This engine was created by Fogel et al [8], and it fixes the problems with machine learning. The evolutionary approach is more effective since the credit assignment method is flawed. Three artificial neural networks are used by the evolutionary algorithm to assess the value of plausible alternate places on the chessboard. The fittest player among the three is determined by the calibre of their moves. All three players begin with neural networks and play in random variation. The most physically fit players then produce the progeny for the following generation. This technique is highly helpful since it allows us to try addressing issues for which there are no known solutions and because the evolution of a high-level chess programme implies larger applicability to problem solving.

D. M. Raif and others [9] The major goal of this study was to integrate ceramic material art to the game in order to improve its aesthetic appeal. This paper transforms the standard chess pieces into cartoon ceramic chess pieces for more aesthetic appeal because cartoons are thought to be the greatest way to communicate a message to the game's audience. Each 3D character is made up of a collection of 2D images that were used to build it in accordance with the intended model and modify its movements. In order to give the game of chess an advantage over the other engines, each piece is given its own shape, size, texture, movement, and personality.

According to Omid E. David et al. [10], unlike traditional chess engines, genetic algorithms can be utilised to bypass raw force and truly advance AI to the point where it can outperform a human player. The method starts with a population, and as it iterates, the fitness increases, increasing the likelihood that a machine will outperform a person. Only 1-ply searches were feasible in the early phases of chess engines; however, thanks to the algorithm's evolutionary nature, tens of plies searches are now feasible in a tournament game. Chess and checkers are used in this research to test this hypothesis.

In this study, Nathaporn Karnjanapoomi et al [11] raise the prospect of abandoning the conventional notion of a two-player chess game in favour of allowing three players to participate simultaneously. In this game, the winner is the person who successfully captures the opponent's King. As a result, this game not only provides a disadvantage but also offers the chance for two players to work together to beat a third. This is accomplished by employing the Non-Dominated Adversarial Search algorithm, which forces the player to prioritise quantity over quality while making movements.

According to Rahul A. R. et al. [12], most game engines are readily defeated by Grandmasters because they do not account for the effects of a move performed in the present on the future. As a result, this engine assesses the position evaluation and delivers the actual difference in the positions of the players. This is significant because, in a game of chess, one side may willingly give up a coin in order to set up a more complex trap. To do this, the computer must assess the situation and the player's actions beforehand. Niching and multi-niching are then employed to win. Additionally, this technology enables players to adopt cutting-edge strategies like casting, which until recently was only feasible in real-world board games. As a result, this engine features the most accurate and effective AI for the game of chess.

III. CHESS ENGINE

A chess engine, used in computer chess, is a programme that analyses chess or chess-variant positions and produces a move or series of moves that it deems to be the strongest. Engines would be the minds of chess players on computers.

Chess programmes are intricate. However, in the simplest terms, they accomplish two crucial tasks:

1. Evaluate: Chess engines analyse each position individually to determine which one is superior. The majority of chess players score games using the same scoring system as is used by most chess computers, with pawns earning one point, minor pieces worth three points, etc. Either chess engine does this differently, but the majority of them consider the material on each side of the board, every threat there is, the king's safety, and the pawn structure. One number represents the total score of the best evaluation in the future. Because they were created by people, conventional engines assess data similarly to humans.
2. Search: Similar to skilled chess players, engines attempt to analyse the situation thoroughly. The better the move they can make today, the further ahead they can look, since they can assess positions that will follow the best moves in the future. Chess moves are classified as "plies" (layers), and the number of plies deep indicates how deep a move is. The majority of engines are already judging deeper and stronger than humans at 20 ply (10 white plays and 10 black moves). Engines can search more than 50 ply deep, depending on the time allotted and the complexity of the position.

Traditional chess engines use a search algorithm called MiniMax.They involve mainly 3 steps:

1. Move-generation: If a specific piece is picked, a function for move creation should provide all potential moves. This gives back a tuple with a list of legal moves from the specified state.
2. Board evaluation: A method that assigns a certain score to the current game condition is required. We can determine whose favour the game is now in based on the score. There are various ways to evaluate, but let's

keep things simple and state that this engine counts the number of pieces each player has and, depending on the number of pieces left, gives White or Black a higher or worse score depending on how many pieces are left.

3. Minimax: Minimax is a type of backtracking algorithm used in game theory and decision-making to determine a player's best course of action, provided that your opponent is likewise playing well. It is frequently utilised in two-player turn-based games like chess, backgammon, mancala, and tic tac toe. The two participants in Minimax are referred to as the maximizer and minimizer. The maximizer strives to achieve the maximum score, whereas the minimizer strives to achieve the lowest score. Each board state is accompanied with a value. If the maximizer has the advantage in a particular situation, the board score will typically be positive. In that board condition, if the minimizer has the advantage, it will typically be some negative number. For each form of game, specific heuristics are used to determine the values of the board.

A useful technique for addressing problems, the minimax algorithm aids in a thorough evaluation of the search space. The ability to execute decision making in artificial intelligence, however, is its most notable benefit. This has further paved the way for the creation of innovative and intelligent technologies, systems, and computers.

Limitations of Minimax: Its large branching factor makes it difficult to reach the desired state quickly. The performance and efficiency of the engine are reduced when unneeded nodes or branches of the game tree are searched for and evaluated. There are many options available for both minimum and maximum players. Due to time and space constraints, it is not possible to explore the full tree.

In this proposed work, a simplified version of Minimax is presented. The work is based on Negamax with Alpha-Beta Pruning.

A. Negamax

Minimax search is a subset of negamax search that makes use of the two-player game's zero-sum characteristic. In order to make the minimax method more straightforward to implement, this technique depends on the fact that max(a,b)= -min(-a,-b). More specifically, in such a game, the value of a position to player A is the opposite of the value to player B. Since the opponent must have valued the successor position, the player on the move searches for a move that maximises the negation of the value arising from the move. Whether A or B is moving, the previous sentence's logic still holds true. This implies that both positions can be valued using a single approach. Over minimax, which mandates that A choose the move with the highest successor value while B chooses the move with the lowest successor value, this is a coding simplification.

The game trees that NegaMax uses are the same ones that the minimax search algorithm uses. The tree's nodes and root node represent different game states for a two-player game, such as different game board configurations. The moves that are accessible to a player who is about to play from a specific node are represented by transitions to child nodes. Finding the node score value for the player playing at the root node is the goal of the negamax search. Figure 1 shows the Negamax on a two-person game tree of 4 piles.

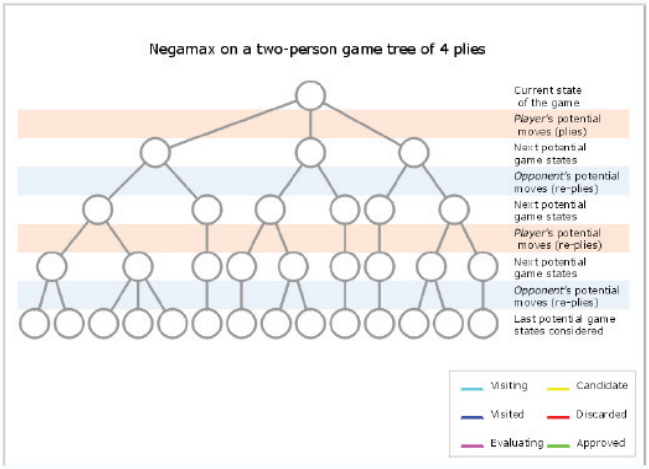

Fig. 1. Negamax on a two-person game tree of 4 piles

B. Negamax with AB-Pruning:

Negamax can benefit equally from algorithm advancements for minimax. Similar to how it is used with the minimax algorithm, alpha-beta pruning can reduce the number of nodes the negamax algorithm analyses in a search tree. Figure 2 presents Negamax with alpha-beta pruning on a two-person game tree of 4 piles.

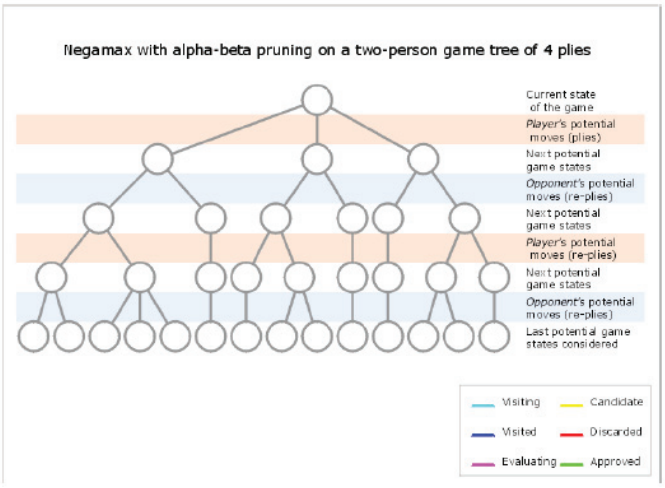

Fig. 2. Negamax with alpha-beta pruning on a two-person game tree of 4 piles

IV. IMPLEMENTATION AND RESULT ANALYSIS

First, make an empty hash and save it in the best_score instance variable. then invoke negamax while supplying the active game object. This will verify each potential game result. Look for the highest value in the best_score hash when the recursive calls are complete, and then return the key while making the best decision to prevent the opponent from winning.

When Negamax is called, the following steps take place:

- The current game object
- A depth of how far into recursive algorithm (starting at 0)
- An integer called alpha, starting at -infinity
- An integer called beta, starting at infinity

- An integer called color, starting at 1
- Alpha and beta represent lower and upper bounds for child node values at a given tree depth. The color indicates the current player, and will either be 1 for us, or -1 for our opponent.

Consider there to be a maximum of two moves from a position for purposes of explanation. The engine invokes the NegaMax function when it is the AI's turn. thinking about limiting the depth to 3.

The opening table file will be checked for moves using the key created from the current state as the function's first step (position). Assume it failed to discover an opening manoeuvre. All possible moves have now been generated (according to our simplification, only a maximum of two moves are allowed). Therefore, now Game Tree looks like Figure 3.

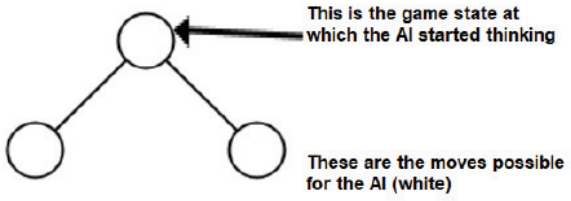

Fig. 3. Game Tree

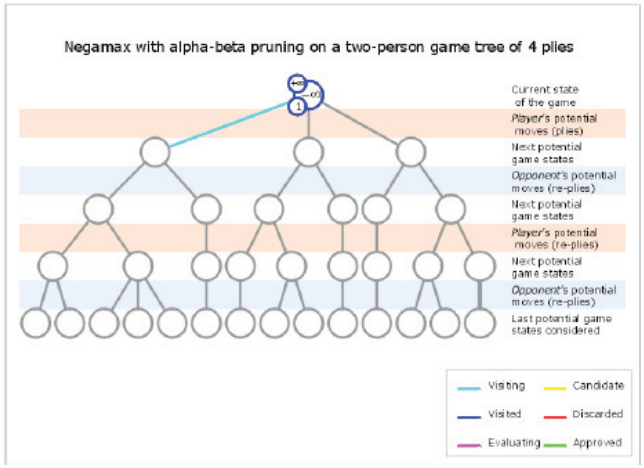

Fig. 4. Initially, Recursive call 1

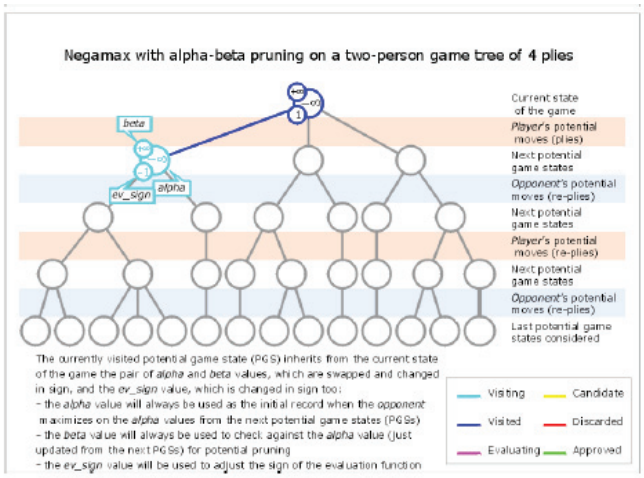

Fig. 5. Recursive call 2

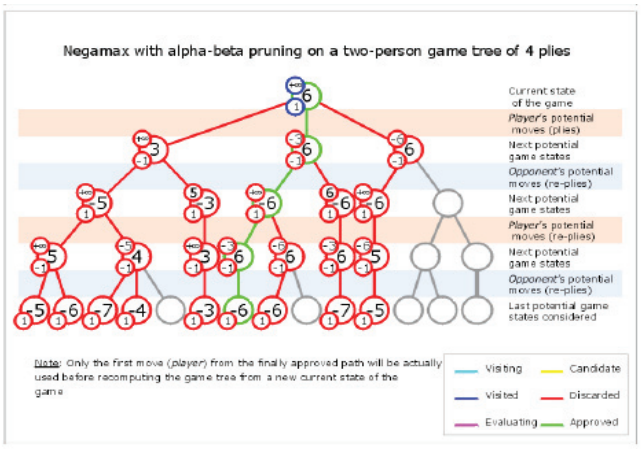

Fig. 6. 'n' Recursive call

Now again recursively it is called for the leaf nodes. Figure 4 and 5 shows the recursive game tree structure. Figure 6 presents the n^{th} Recursive call.

V. CONCLUSION

Chess is played with this application. A friend or the computer can be the opponent for the user. The majority of the processing is therefore performed on a list of strings because the game state is mostly maintained as a 2D list of strings. The GUI displays the current state on the screen using that state. Both click-and-drag and drag-and-drop movement of parts are supported by the GUI. When playing against a human, the AI analyses every move that could be performed by either player, up to a given depth. The AI assigns a score to each position after evaluating it.

The more favourable a position is for white, the higher the score value; conversely, the more favourable a position is for black, the lower the score value. The AI assumes best play from either side as it moves up the search tree and decides the best move to be played since it is aware that white will want to increase the score and black will try to decrease the score. The quantity of roles that need to be examined could become a concern. Many thousands of positions need to be assessed, even at three levels of depth. The AI must understand how to assess the board at each step in order to navigate the search tree as described above.

Currently, assessment function evaluates the board based on three key factors: a) The material for black and white. Every part has a value. and position is likely to be better the more pieces it have. b) Piece-square table values: For each piece, a table containing the optimal squares that specific piece should occupy is kept. c) Point deductions for blocked, isolated, and doubled pawns. d) A checkmate: If this occurs, the position receives a very high point, and the AI will try to move towards this if it can (or avoids it).

REFERENCES

[1] Anushka Nair, Kanksha Marathe, Prof. Suvarna Pansambal, "Literature Survey of Chess Engines", International Journal of Engineering Research & Technology (IJERT), Volume 5, Issue 01, 2017.

[2] Swaminathan B, Vaishali R and subashri T S R, "Analysis of Minimax Algorithm Using Tic-Tac-Toe", Intelligent Systems and Computer Technology, 2020.

[3] K. Greer, "Tree pruning for new search techniques in computer games", Advances in Artificial Intelligence, 2013:2, 2013.

[4] Liqun Zhang, Lili Ding, Zhenlai Li, "The Design of Surakarta Chess Battle Platform in Computer Game", 25th Chinese Control and Decision Conference (CCDC), IEEE 2013.

[5] Lingling Wang, Yiyang Wei, Feng Li, "Research on Hexchess Game System based Artificial Intelligence", WSEAS Transactions On Business and Economics, Volume 19, 2022.

[6] Robert Levinson, Feng-hsiung Hsu, "The Role of Chess in Artificial Intelligence Research", Proceedings of the 12th international joint conference on Artificial intelligence - Volume 1, 1991.

[7] Murray Campbell, A. Joseph Hoane Jr. b, Feng-hsiung Hsu, "Deep Blue", Elsevier, Artificial Intelligence 134 (2002) 57–83, 2002.

[8] Fogel, Hays, Hahn, and Quon, "A Self-Learning Evolutionary Chess Program", Proceedings of the IEEE, Vol. 92, no. 12, December 2004.

[9] D. M. Raif, R. Anwar, N. A. Ahmad, Z. Zakaria and M. F. A. Jalil, "Revision on Cartoon Character Integrate with Chess Concept for Industrial Ceramic Artware", IEEE Business Engineering and Industrial Applications Colloquium (BEIAC), 2013.

[10] Omid E. David, H. Jaap van den Herik, Moshe Koppel, and Nathan S. Netanyahu, "Genetic Algorithms for Evolving Computer Chess Programs", IEEE Transactions On Evolutionary Computation, Vol. 18, No. 5, October 2014.

[11] Nathaporn Karnjanapoomi, Pongpol Pramanpol, Voratep Lertratsamewong, Torsakuln Chacavarnkitkuln, Vazuthorn Rattanajongjittakorn and Chaicharn Thavaravej, "A Nondominated Adversarial Search Algorithm for a Three-player Chess Game", International Computer Science and Engineering Conference (ICSEC), 2013.

[12] Rahul A R and G Srinivasaraghavan, "Phoenix: A Self-Optimizing Chess Engine", IEEE International Conference on Computational Intelligence and Communication Networks, 2015.