Chess Software Based on Ai

[1]Rahul Gaikwad, [2]Shaila Pawar, [3]Vaishnavi Olekar, [4]Mahesh Namdas

A C Patil College of Engineering

Abstract

Chess, which is played between two players on a board, is an intellectual and mental game. Its unique rules of play help to better the player's mental and intellectual activities, and it has a huge following of players all over the world who are keen to play it. This document discusses the completely computerized chess game. The game is first computerized so that two players can play chess on a computer using all of the current rules. Second, the AI program was used to create the game, adding the computer's intelligence to make it more compelling for players to play directly against it.

Keywords: Chess, Chess AI, Game ,Chess Algorithm, Negamax Algorithm, PVP Game, Play with Computer.

I. INTRODUCTION

Two players compete in the strategy board game of chess, which is played on a checkered board with 64 squares organized in an 8 by 8 grid. Chess, which is played by millions of people globally, is thought to have originated from the Indian game chaturanga before the seventh century. In chess composition, there are several schools of thought and each emphasizes a distinct emphasis on the difficulty of the problems. The 19th century saw the initial rise in the popularity of chess studies. They are renownedly challenging puzzle involving of puzzles that involves intricate computations and tactical motifs. As a result, they serve as a useful reference point for research on artificial intelligence.

The area of computer science known as artificial intelligence (AI) is expanding quickly and is now more widely available. It is having an exponentially greater social and technological effect. The major players in the digital industry, including Google, Facebook, Amazon, and Microsoft, have been providing services and products based on AI. With the help of this technology, a variety of applications have been created, including ones for video games, financial services, and machine autonomy.

Investigating the development of chess engines throughout history can be one method to examine how AI has advanced. From 1997, when they defeated Garry Kasparov, the best human chess player at the time, to the present, when Google DeepMind created a neural network that played the game and defeated the greatest chess engine to date after only four hours of training. Chess is a challenging game that has not yet been mastered and is unlikely to be anytime soon. It is difficult to create an artificial intelligence that can comprehend and perform the game well.

Currently, there are numerous chess AI systems being used by professionals, but we built our AI system using a distinct methodology. We used the Negamax algorithm, an improved variant of the Min Max algorithm, which is typically used to create games for two players, such as chess. NegaMax's advantage over competing against computer systems is its effectiveness and reduced evaluation time for the same position.

Along with the AI algorithm, the project has developed a Player Vs. Player mode that enables players to compete against friends while adhering to all chess regulations. We had to create the game interface from scratch using the Pygame module in order to play in such an appropriate way. A platform module for Python called "pygame" enables coders to create video games. It has music and computer graphics libraries helpful for creating games in Python.

II. LITERATURE SURVEY

Deep Blue was created with assistance from Murray Campbell and others [1] Garry Kasparov was the reigning World Chess Champion when Deep Blue beat him in a six-game match in 1997. Deep Blue's chess engine uses a single chip move generator, which combines hardware and software searches on silicon chips to produce the optimal moveset. According to this drawback, it has some flaws that contributed to its defeat by chess expert Garry Kasparov in 1996. This defeat was brought on by the inability to find a sacrifice strategy.

By using the piece positions, material and positional values, and neural networks to analyse particular chessboard sections, Fogel et al. [2] created the software that learned to evaluate chessboard configurations. The computer software increased its play during evolution by nearly 400 rating points. Testing against Pocket Fritz 2.0 in a simulated tournament environment revealed that the evolved programme works better than master level. Three artificial neural networks are used by the evolutionary algorithm to assess the value of possible alternate locations on the chessboard. Evolutionary algorithms have the disadvantage of not understanding what an optimum solution is or how to determine whether a given solution is optimal.

In this effort, which serves as our chess engine,Tong Lai Yu [3] used the well-known open-source computer chess engine Beowulf. Although Beowulf is single threaded, it is easy to use and understand. It made use of the Beowulf single-threaded chess engine. The Negamax Search algorithm is used by Beowulf to scan a game tree. This has the drawback of being a text-based engine and lacking a true multimedia user interface with graphics.

Chess is examined through the perspective of genetic algorithms in the Rahul A. R. et al [4] With just a few learnable parameters, we completely teach a genetic player. Positional Value Tables (PVTs), which are frequently used in chess algorithms to assess the quality of a position, are optimised using multi-niche crowding. The player advances to the rank of an International Master with a very basic setup and just 1000 generations of evolution. The material difference between the player positions is returned after the position assessment has been assessed. It cannot see the opponent's motivation to sacrifice the piece for positional benefit because it depends on the evaluation of the pieces.

The duchess chess system was developed by Diego Real et al [5]. Each game can have up to six participants, and an extra coin known as the "Duchess" that can make Bishop and Knight moves is also added. This engine also includes a third player, the computer, to assist out if needed and to take the place of a player who loses the game. The Tree Strap Algorithm has made it possible to operate this engine, which was previously impossible due to a lack of effective technology. Searching cumulatively, or only searching the board where the changes have happened, speeds up the tree search. Due to this, the game still plays quickly even with six people online at once.

A novel method for automated chess commentary generation is examined by Hongyu Zang et al [6] with the goal of producing chess commentary writings in various categories (e.g., description, comparison, planning, etc.). In order to assist with encoding boards, forecasting moves, and evaluating scenarios, we incorporate a neural chess engine into text generation models. The models are improved by simultaneously training the neural chess engine and the generation models for various categories. We run experiments on five different categories in a benchmark Chess Commentary data set, and both automatic and human assessments yield encouraging findings.

Using the Native Development Kit (NDK) Dmitri Gusev et al [7] ported an open-source chess engine to Android, evaluated it against competing engines, and engaged in competitive play. A variety of difficulties and revelations emerged during the porting process, some of which might be shared by other mobile application ports while others are probably specific to the chess game. They discovered that the architecture of chess engines based on the universal chess interface (UCI) protocol allowed rapid adoption of an advanced user interface and that only a small number of changes were required to have a functioning engine.

Giraffe, a chess engine developed by Matthew Lai [8], employs self-play to learn all of its domain-specific information with only a small amount of hand-crafted programming knowledge. Giraffe's learning system conducts automatic feature extraction and pattern recognition, in contrast to earlier attempts using machine learning only to perform parameter-tuning on custom evaluation functions. The trained evaluation function works on par with the evaluation functions of modern chess engines, which all contain thousands of lines of meticulously hand-crafted pattern recognizers that have been fine-tuned over many years by both human and computer chess masters. The most effective end-to-end machine learning chess programme to date is called Giraffe.

The primary goal of D.M. Raif, et al [9] 's article was to increase the game's graphical appeal by incorporating ceramic material into the artwork. This paper transforms the standard chess pieces into ceramic cartoonist chess pieces for greater visual appeal because cartoons are thought to be the best method to communicate a message to the game's audience. Each 3D character is made up of a collection of 2D images that are used to build it in accordance with the intended model and modify its motions. As a result, each component is given a unique form, texture, size, and movement to set it apart from other engines.

According to Omid E. David et al. [10], genetic algorithms, as opposed to other traditional chess engines, can be used to bypass sheer force and advance AI to the point where it can outperform a human player. The algorithm begins with a population, and as it iterates, the fitness increases, increasing the likelihood that a computer will outperform a person. Initially, chess engines could only perform one move at a time, but as algorithms have advanced, they have made it possible to perform tens of lines of searches during a tournament game.

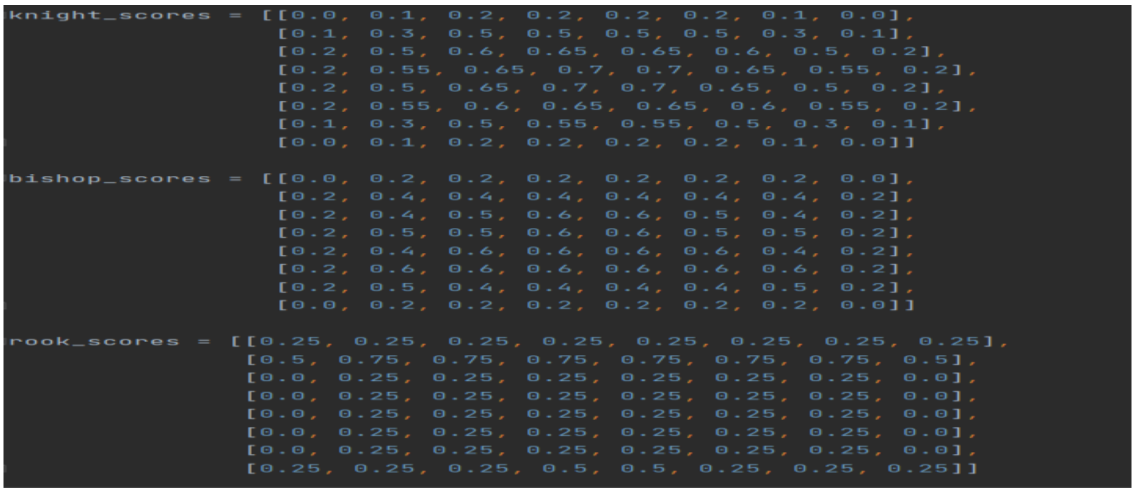
III. PROPOSED SYSTEM

NegaMax, an AI algorithm, and Alpha Beta Pruning are both used in the undertaking “Chess Software based on AI" which allows people to play chess. The game's rules and legal movements have all been established and are in accordance with how Chess is played in reality.

“Chess Software based on AI" allows the user to play chess effectively on a computer as well as against our in-built Chess AI, which was created to be a tough challenge for the player in order to progress in this traditional sport of our country that is now played internationally. Additionally, our suggested system has a very user-friendly GUI and a move record for each action taken during the game. Along with all of this, our system also has a logger of all moves, which essentially displays every move made during a game in the conventional style of notation used in professional chess tournaments and provided by the FIDE.

In addition to using the algorithms NegaMax and Alpha Beta Pruning, we have also used an array system, which enables AI to determine which piece is better suited for a particular location during the intermediate phase of the game of chess, where there is no tactical sequence of moves and a need for positional gameplay to gradually increase the utility and scope of the pieces. To achieve this, we have assigned each position a value with regard to each piece that ranges from 0.0 to 1.0.

For comparison, the same positional valuations for the Knight, Bishop, and Rook are provided below.



```
knight_scores = [[0.0, 0.1, 0.2, 0.2, 0.2, 0.2, 0.1, 0.0],
                 [0.1, 0.3, 0.5, 0.5, 0.5, 0.5, 0.3, 0.1],
                 [0.2, 0.5, 0.6, 0.65, 0.65, 0.6, 0.5, 0.2],
                 [0.2, 0.55, 0.65, 0.7, 0.7, 0.65, 0.55, 0.2],
                 [0.2, 0.5, 0.65, 0.7, 0.7, 0.65, 0.5, 0.2],
                 [0.2, 0.55, 0.6, 0.65, 0.65, 0.6, 0.55, 0.2],
                 [0.1, 0.3, 0.5, 0.55, 0.55, 0.5, 0.3, 0.1],
                 [0.0, 0.1, 0.2, 0.2, 0.2, 0.2, 0.1, 0.0]]

bishop_scores = [[0.0, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.0],
                 [0.2, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.2],
                 [0.2, 0.4, 0.5, 0.6, 0.6, 0.5, 0.4, 0.2],
                 [0.2, 0.5, 0.5, 0.6, 0.6, 0.5, 0.5, 0.2],
                 [0.2, 0.4, 0.6, 0.6, 0.6, 0.6, 0.4, 0.2],
                 [0.2, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.2],
                 [0.2, 0.5, 0.4, 0.4, 0.4, 0.4, 0.5, 0.2],
                 [0.0, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.0]]

rook_scores = [[0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25],
               [0.5, 0.75, 0.75, 0.75, 0.75, 0.75, 0.75, 0.5],
               [0.0, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.0],
               [0.0, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.0],
               [0.0, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.0],
               [0.0, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.0],
               [0.0, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.0],
               [0.25, 0.25, 0.25, 0.5, 0.5, 0.25, 0.25, 0.25]]
```

Fig. 1 Positional Valuation for Pieces

IV. SYSTEM WORKING

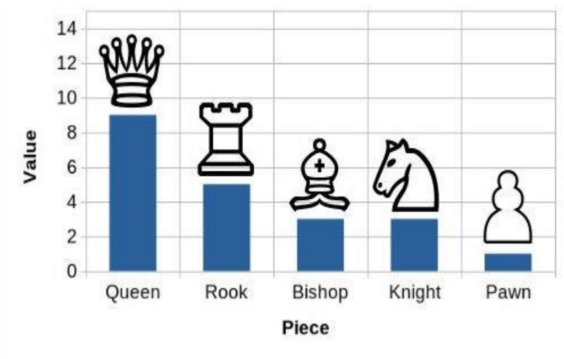


Fig. 2 Pieces values chart

Evaluation:

The most crucial element used to create chess AI is the evaluation function. This module takes a chess location as input and outputs a number. If this number provides us an evaluation of 0, it indicates that both players are in an equal position. More advantage goes to white people when the positive number is greater; more advantage goes to black people when the negative number is lower. The algorithm searches through various data before producing an assessment. This has a hardcoded great mastermind imitation. For a basic grasp of chess and chess engines, consider this. Heuristic function is the type of evaluation function used in this undertaking.

A heuristic function is a function that measures the chessboard, assigns a weight to each measurement, and then calculates the numerical worth of each player's advantage. The value of each chess component is being measured in this instance. The values are as follow Rook: 5, Knight:3 Bishop: 3, Pawn: 1, Queen: 9.

Additionally, our programme offers signup and login options. Additionally, we created a GUI that enables users to select from a variety of settings. The PVP Mode (Player Vs. Player) and the Chess AI Engine are the two operating modes of "Chess Software built on AI." The PvP option aids the user in playing chess with a friend in accordance with all Chess rules. While the Chess AI, which was created by us and employs the Negamax algorithm, allows the user to play the game against a computer.

NegaMax :- Negamax search is a variant form of minimax search that relies on the zero-sum property of a two-player game. This algorithm depends on the fact that max(a,b) = -min(-a,-b) to make the minimax algorithm more straightforward to implement. More specifically, in such a game, the worth 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 maximize the negation of the value coming from the move. Whether A or B is moving, the preceding sentence's logic still holds true. This implies that both positions can be valued using a single method. This is a coding simplification over minimax, which requires that A selects the move with the maximum-valued successor while B selects the move with the minimum- valued successor.

Alpha Beta Pruning:- A search algorithm called alpha-beta pruning aims to reduce the number of nodes in its search tree that are assessed by the minimax algorithm. It is an adversarial search method frequently used for automated two-player game play (Tic-tac-toe, Chess, Connect 4, etc.). When at least one scenario that shows a move to be worse than a previously studied move has been identified, the evaluation of that move comes to an end. Such actions do not require further analysis. When used on a typical minimax tree, it produces the same move as minimax but removes branches that are unable to affect the ultimate choice.

V. DESIGN

Algorithm:-

1 : Start

2 : Enter username and password

3 : If there is no login details then redirect to signup page

4 : Once Sign up is done ,go to login page to enter username and password.

5 : If username and password is matched then redirected tochoosing the mode page

6 : If PVP mode is chosen then redirect to PVP Game

7: If Play with AI is chosen then redirect to Chess AI Game

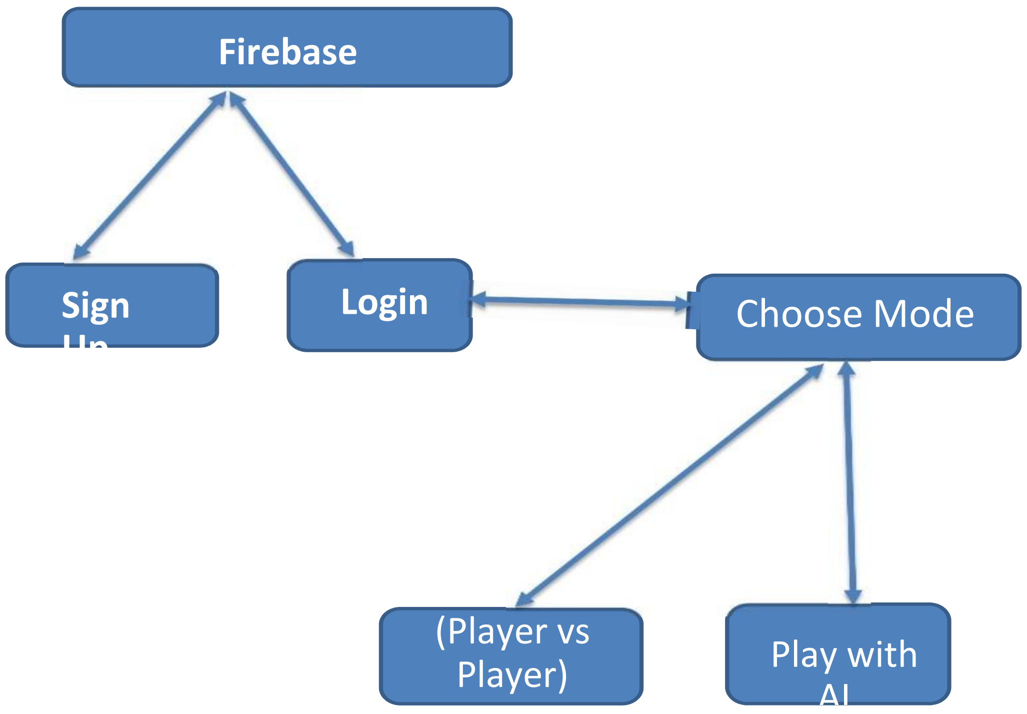


Fig. 3 Working Algorithm

VI. RESULT

1. Sign up Page

On the Sign Up Page, you have to do the registration by giving your email id andpassword which will then be stored in firebase.

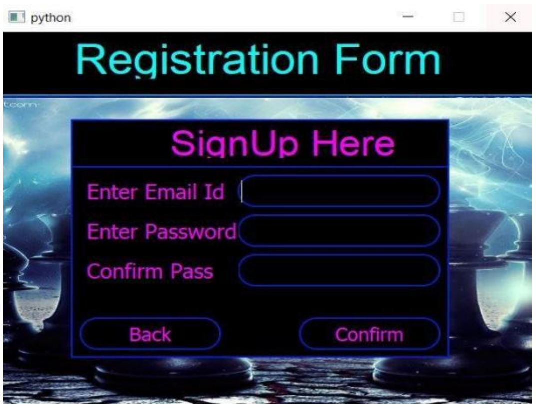


Fig. 4 Sign Up Page

2. Login Page

On the login page, you have to write the email id and password which you used for registration.This is then checked by the firebase if it is correct or not.

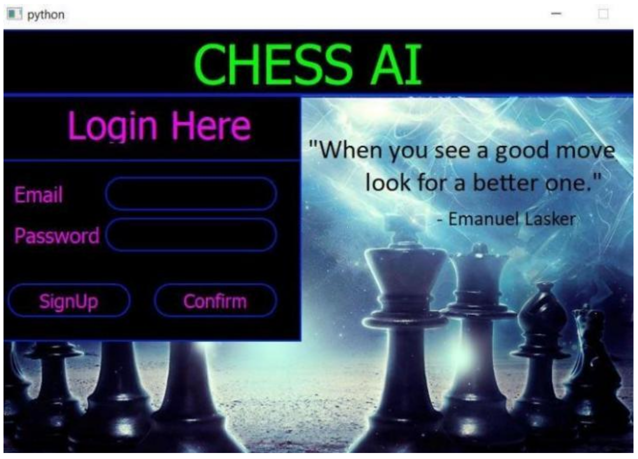


Fig. 5 Login Page

3. Homepage

This is the homepage which allows you to chose the mode which are the player vsplayer mode or play with AI mode.

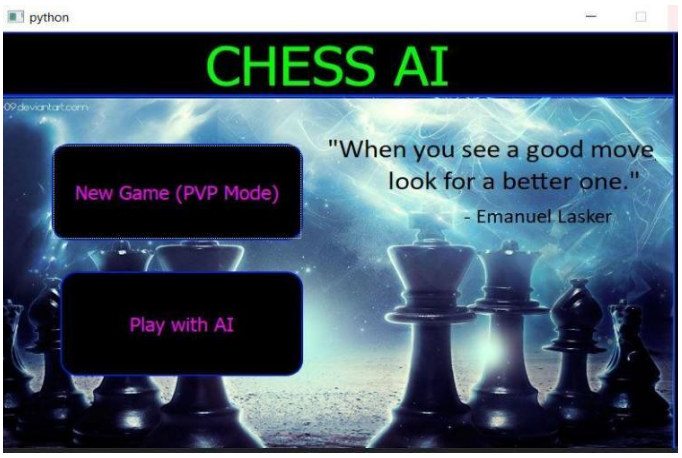
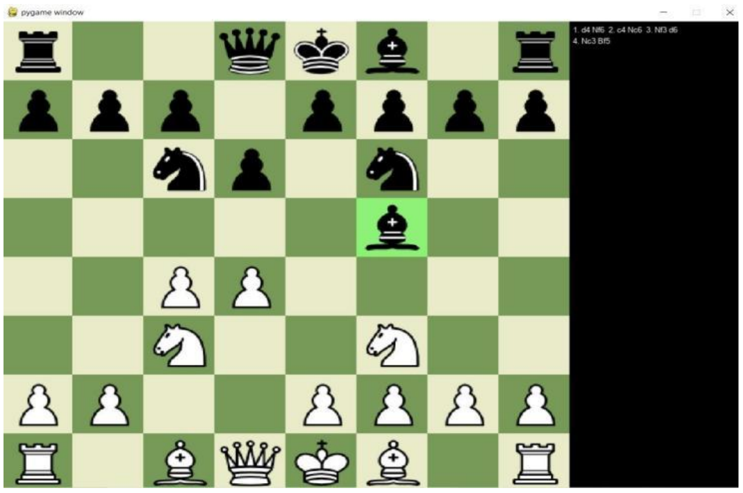


Fig. 6 HomePage

4. Chess AI

This is our main Chess AI software in which we can play against the computer.There is also move log in side which shows all the moves played in agame.



```
1. d4 Nf6 2. c4 Nc6 3. Nf3 d6
4. Nc3 Bf5
```

Fig. 7 Chess AI Mode

VI. CONCLUSION

There were previously Chess AI based on Min Max Algorithm but we used a more enchanced version of it i.e. The NegaMax Algorithm along with also enabling the AI to see positional advantages. Our system is better in terms of positional play also which is very essential when it comes to Intermediate level and Advance level of Chess gameplay.

ACKNOWLEDGEMENT

We are thankful to of IT department, **Shaila Pawar** for helping, giving guidance & giving information in project "Chess Software based on AI ".

REFERENCES

1. Murray Campbell, A. Joseph Hoane Jr. b, Feng-hsiung Hsu, Deep Blue, Elsevier2002, Artificial Intelligence 134 (2002) 578.
2. Fogel, Hays, Hahn, and Quon, A Self-Learning Evolutionary Chess Program,Proceedings of the IEEE, Vol. 92, no. 12, December 2004.
3. Tong Lai Yu, Chess Gaming and Graphics using Open-Source Tools, IEEEComputer Society, 2009..
4. Rahul A R and G Srinivasaraghavan, Phoenix: A Self-Optimizing Chess Engine,2015 IEEE International Conference on Computational Intelligence and Communication Networks.
5. Diogo Real and Alan Blair, Learning a Multi-Player Chess Game with TreeStrap,2016 IEEE Congress on Evolutionary Computation.
6. Hongyu Zang and Zhiwei Yu and Xiaojun Wan Institute of Computer Science and Technology, Peking University The MOE Key Laboratory of Computational Linguistics, Peking University Center for Data Science, Peking University "Automated Chess Commentator Powered by Neural Chess Engine".
7. Corey Abshire and Dmitri Gusev Indiana University, Purdue University Columbus, IN, U.S.A., Columbus, IN, U.S.A "Firenzina: Porting a Chess Engine to Android".
8. Matthew Lai Imperial College london Departnent of Computing "Giraffe: Using Deep Reinforcement Learning to Play Chess".
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", 2013 IEEE Business Engineering and Industrial Applications Colloquium.
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.