IoT and Intelligent System First Year

Course: Experiential Learning

Course Code: DA1001

## Chess Engine with ai

by

**Radhey Sharma**

Reg. No: 2193111078

Under the guidance

of

**Mr. Vinod Kumar**

**Assistant Professor**

Iot, School, Manipal University Jaipur, Jaipur (Raj.).

**Department of IoT**

**School of Computer Science and Information Technology**

**Faculty of Engineering**

**Manipal University Jaipur, India**

March, 2022

# Certificate

This is to certify that the project titled **Chess Engine with AI** is a record of the bonafide work done by **Radhey Sharma** (Reg No:219311078) submitted for the partial fulfillment of the requirements for the completion of the Experiential Learning (DA1001) course in the Department of IoT of Manipal University Jaipur, during the academic session July-November 2021.

|  |
| --- |
|  |
| Mr. Vinod Kumar  Assistant Professor  Department of Computer and  Communication Engineering |
|  |
| Prof. Vijaypal Singh Dhaka  Head of the Department  Department of Computer and  Communication Engineering |

# Abstract

Chess is a game that develops mental abilities such as concentration, abstract reasoning, critical thinking, problem-solving, strategic planning, and creativity. The program runs on a C code that the player can use to play single-player mode or multiplayer mode and can reset the board at any time during the game. They can also play multiple games without executing the file again. The program tracks movements on the board and other important details like the king’s position, pawn’s piece status, or rook’s piece status. This program, however, does not keep a record of the previous moves, and the data is lost once another move is played. The program uses the console to show the board. A minimax-based algorithm is used to generate moves for the computer and it can also print a list of all possible moves for the player if they like.

# Introduction

# This project aims to make a program that allows a user to play chess against an AI that uses an algorithm to make pseudo-random moves.

# Scope:

# The scope of this project is to make a functioning chess engine that enables its user to play chess with the computer or another user.

# Objectives:

# There are three main objectives of this project:

# 1. Create a chessboard with chess pieces.

# 2. Add movement constraints and rules of the chess game.

# 3. Create an AI that uses a Mini-Max algorithm to make moves.

# Requirement Analysis and Specifications:

# The different requirements of the project are:

# 1. A Chess board and its layout & design.

# 2. The Chess pieces design & moves generation technique and logic.

# Analysis:

# It is the detailed study of the various operations or moves of the system. The key question is what must be done to solve the problem.

# Design:

# The most creative and challenging phase of the system life cycle is system design. The design is a solution to develop the know-how of the new system under consideration. It provides the understanding and procedural details necessary for implementing the program.

# Testing:

# Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct the results will be successfully achieved. Effective testing early in the process translates directly into long-term cost-saving, a reduced number of errors. System testing is done when all the modules of the system are in working order and have been tested independently for proper working. All the pieces are put into one system and tested to determine, whether it meets the user's requirements. System testing is designed to uncover weaknesses that were not found in earlier tests like program testing in which only syntactical and logical are removed. The purpose of System Testing is to consider all the likely variations to which it will be subjected and then push the system to its limits.

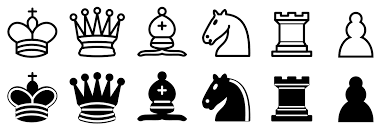
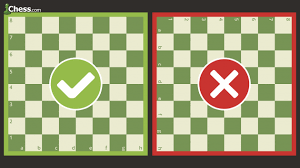
# Program Testing: It checks types of errors that are Syntax and Logic.

# User Acceptance Testing: An acceptance testing has the objective of telling the user about the validity and reliability of the system that is the developed system is functioning according to the needs prescribed by the user. This testing is done with the actual data in presence of the user at the user’s place.

# Basics of Chess:

# Chess is a game for two players using a chequered board of sixty-four squares with eight pieces, viz. a king, queen, two rooks, knights, and bishops, and eight pawns each, with the object of placing the opponent's king in checkmate. The game begins with the position shown above. The White player (the player of the light-colored pieces) moves first. Then each player takes a single turn. In fact, a player must move in turn. In other words, a move cannot be skipped. When setting up the pieces, keep in mind two things. The light-colored square goes on the player's right, and Queens goes on their color next to the Kings on the center files. You may not move a piece to a square already occupied by one of your own pieces. You may capture an opposing piece by replacing that piece with one of your own pieces if it can legally move there.

# 



# The primary objective in chess is to checkmate your opponent's King. When a King cannot avoid capture then it is checkmated and the game is immediately over. If a King is threatened with capture but has the means to escape, then it is said to be in check. A King cannot move into check, and if in check must move out of check immediately. There are three ways you may move out of check:

# 1) Capture the checking piece.

# 2) Block the line of attack by placing one of your own pieces between the checking piece and the King. (Knights cannot be blocked.)

# 3) Move the King away from check. If a King is not in check, and no other legal move is possible, then the position is said to be in a stalemate. A stalemated game is a draw or a tie.

# The Pieces and The Rules:

# KING: The King is the most important piece.

# 

# When it is trapped so it cannot move without being captured, then the game is lost. This trap is called checkmate. The King can move one square in any direction. A King can never move into check, or onto a square where it can be captured by an opponent's piece. If a King is not in check, and no other legal move is possible, then the position is said to be in a stalemate. A stalemated game is a draw or a tie.

# QUEEN: The Queen is the most powerful piece.

# 

# The Queen can move to any square in any direction as long as her path is not blocked. Her range and the ability to attack many pieces at once is the source of its power.

# ROOK: The Rook is the next very powerful piece after Queen.

# 

# The Rook can move to any square along with its file or row as long as its path is not blocked. Its range is the source of its power.

# BISHOP: The Bishop comes next to Rook in terms of power.

# 

# Bishop can move to any square along its diagonals as long as its path is not blocked. Its range is the source of its power.

# KNIGHT: The Knight is nearly as powerful as the Bishop.

# 

# The Knight is the only piece that can hop over other pieces in an L-shaped path. This ability makes it particularly powerful in the early stage of a game when the board is crowded with pieces.

# PAWN: The Pawn is the least powerful piece because of its poor mobility.

# 

# The Pawn may move only one square forward if its path is not blocked. However, it may move as an option one or two squares forward on its first move only. It may capture only diagonally one square. It may not capture forward. It may not move backward. The lowly Pawn usually does not last long, but if it is able to reach the 8th row or rank, then it can promote itself to any other piece except the King. A Pawn thus promoted is replaced by that piece. Therefore, it is possible to have more than one Queen, or two Rooks, Bishops, or Knights on the board at one time.

# Some Special Moves and Cases:

# 1) Castling:

# castling chess for Sale OFF 62%

# Castling is an important move in chess. It allows a player to quickly move both the King to safety and the Rook to the center for battle. For this reason, wise players carefully guard their ability to castle and usually castle early in the game. Likewise, clever players will attempt to prevent their opponent from castling. When castling the player moves his King two squares toward one of the player's Rooks and moves that Rook to the opposite side of the King. A player may not castle if either the King or the Rook involved have already moved. Also, the King may not castle out of, through, or into check. There must be no pieces between the King and Rook when castling.

# 2) En-passant:

# En passant - Wikipedia

# A player may capture another player's pawn in passing (En Passant) under very specific circumstances. This move is designed to prevent a player from taking advantage of the two-square first move rule for pawns which might allow them to pass their opponent's pawn(s) without a chance to capture. The capture is made exactly as if the pawn moved only one square on the first move. In the picture, Black's pawn moved up two squares as is its right. White captured the pawn by removing it from the board and placing the passed white pawn on the square marked ep before playing another move. This move, like any other, is optional and can occur as often as a similar situation arises between pawns.

# 3) 50-move draw:

# The game is said to be a draw if none of the sides have moved a pawn or captured a piece for 50 moves.

# 

# Literature review

Computational intelligence techniques were combined with games for the first time in 1959 when Samuel applied a simple reinforcement learning algorithm to the board game Checkers. After the early success of Samuel, the research in the field remained silent for a long time. But as part of artificial intelligence research, a few researchers have worked on applying classical AI techniques, basically especially custom-made search algorithms, to board games such as Chess and Checkers. This direction of research ultimately led to the much-publicized victory of the IBM Deep Blue Chess computer overworld Chess champion Gary Kasparov in 1997.

The legendary game designer Sid Meier defines a game as “a series of meaningful choices”. In discussing these and other definitions, game designer Raph Koster also gave his wonderful effects in the “Theory of fun game design” book. According to Koster, a game is fun to play because we learn the game as we play; we understand and learn the patterns underlying the game, and finally, master how to play it.

Thus Botvinnik for this reason placed games like chess into the class of inexact problems. As the player cannot, in general, see the exact influence of a move on the final goals of the game, it follows that her reasoning must be heuristic. Also S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach Prentice Hall, 1995 also had the explanation of using Artificial Intelligence Technique providing a better approach towards the designing of a chess game.

# Methodology

To create the chessboard, a 2D Array is used. Its rows are numbered 1 to 8 from bottom to top and its columns are numbered 1 to 8 from left to right. Chess pieces have different values assigned to them and the computations will be carried out by various functions.

The variables used for the program are as follows:

board[8][8]: The chessboard is an 8x8 array.

undoboard[8][8]: Previous state of the board variable.

castle[2][2]: Used to check if castling is possible or not.

undocastle[2][2]: Previous state of the castle variable.

king[2]: Used to store the location of the king.

undoking[8][8]: Previous state of the king variable.

enpassant[2][8]: Used to store the status of all pawn pieces.

undoenpassant[8][8]: Previous state of the enpassant variable.

posmov[512][2]: Used to store all possible moves for current positions.

nummov: Used to count number of moves played.

turn: Used to verify a side’s turn.

prmte: Used to check if a pawn is promoting.

undoprmte: Previous state of the prmte variable.

move50: Used for 50-move draw rule.

undomove50: Previous state of the move50 variable.

side: Used to store player’s side.

reset( ): Used for initialization of various variables.

undo( ): Used to undo a move to go back to previous position.

set( ): Used to set variable conditions. undo( ) cannot be used to undo the set( ) changes.

showboard( ): Used to print the current positions on the board.

check( ): Used to check if a position is under attack.

move( ): Used to make changes on the board.

islegal( ): Used to check if a move is legal or not.

promote( ): Used for pawn promotions.

copy( ): Used to copy all elements of a matrix to another.

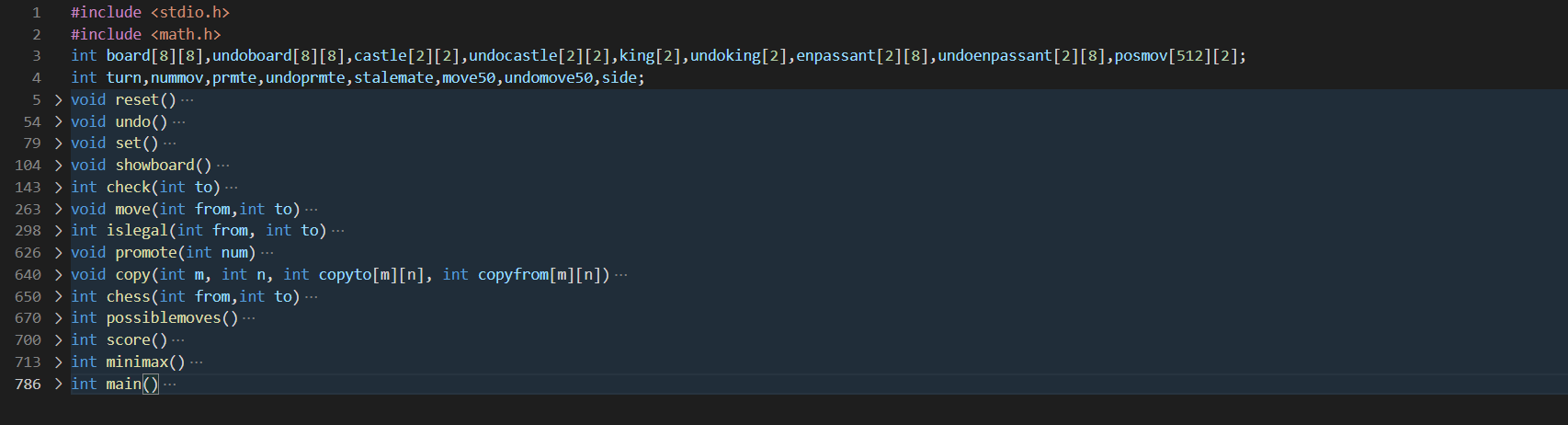
chess( ): A function that contains all logic functions for calling and computing the game.

possiblemoves( ): Used to calculate all possible moves.

score( ): Used to calculate an arbitrary score to help the algorithm find moves with preferable outcomes.

minimax( ): Used to calculate preferable moves for the computer.

main( ): The function for taking inputs and calling the chess function.



# Results

# Printing the board

# 

# Allowing pieces to move

# 



# Making some rules for moving and capturing



# 



# 



# Making the players play only on their turn

# 



# Adding special moves:

# 1. Castling

# 



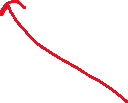
# 2. En passant

# 



# Adding check, checkmate and stalemate

# 



# Printing list of all values

# (Typing 1 to print all legal moves for the current positions)

# 

# Making the starting menu.

# 



# Making an algorithm for the computer.

# 



# (Moves highlighted in yellow are played by the player while the moves highlighted in red are the ones calculated by the computer)

# Conclusions

This project presented the classic and standard version of Chess implemented in a console. The execution has successfully created a game using only the C language. The project had two aims, all of them which were achieved. The aims of the project were important as they identified what the project was trying to achieve. Their aims were:

• The first aim is to allow two users or players to play the game of chess.

• The second aim of the project is to make a working algorithm for single-player mode.

The final report for the project was good, as it showed how the academic question was created from the initial problem. The next section of the report was the literature review, which contained quality sources from various different formats such as books and websites. The sources gave excellent background knowledge of the topic and aided in answering the academic question.

The planning for the project was very well carried out as it initially had two weeks of slack available. This allowed for leniency if any task took longer than anticipated. All tasks were completed in the order they were anticipated. The artifact was superbly developed and implemented. Findings from the literature review were used to ensure that the game covered several different learning styles.

The testing methods used in the project were appropriate. The system test ensured that all features of the chess game were functional. There were no other suitable tests found that would test the game as thoroughly as the system test.

# Future prospects

This project was challenging and helped in understanding the basics of algorithms. However, this project is far from a complete chess engine and is missing some important features. Limitations of the chess game were found from the usability test results, which identified some features that could have enhanced the user experience. Three major limitations arose during the working on the code:

1. Storing all the moves and positions could allow to undo moves and help in making a better algorithm.

2. Making an algorithm with variable difficulty: Asking user for input from a range to set the depth of the algorithm which is currently constant.

3. The program is not user-friendly and using it is not convenient. Using a GUI would enhance the experience for both the users and the programmers for the testing.

# References

# Websites:

# Chess.com, (2022).

# URL: [*https://www.chess.com/learn-how-to-play-chess*](https://www.chess.com/learn-how-to-play-chess)

# (For the images and clarification in Chess rules.)

The last date accessed: 06-01-2022

* Wikipedia, (2022).

URL: [*https://en.wikipedia.org/wiki/Minimax*](https://en.wikipedia.org/wiki/Minimax)

(For Mini-Max Algorithm of the decision theory.)

The last date accessed: 29-12-2021

Books:

* Yashavant Kanetkar, Let us C, (2022), 5th Edition, BPB Publications.

(For help in C programming)

# Acknowledgments

I would like to express my special thanks of gratitude to my teacher Mr. Vishnu sir as well as our HOD Professor Vijaypal sir who gave me the golden opportunity to do this wonderful project on the topic “Chess engine with AI”, which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them. Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.