KANTEC Chess

Version 1.0.0

Authors: Esther Anaya, Kirby Burke, Calla Chen, Noah Mathew, Trang Nguyen, and Artin Tammadon

Affiliation: UCI EECS

Table of Contents

(do at the end when everything else in the manual is completed)

Glossary

Artificial Intelligence – refers to the simulation of human intelligence in machines that are programmed to think and learn like humans

Castling – A special move involving the king and one of the rooks, allowing both pieces to move simultaneously.

Check – A condition that occurs when a player’s king is under threat of capture on the opponent's next turn. A king so threatened is said to be in check.

Checkmate – A condition that occurs when a king is placed in check and has no legal moves to escape.

Data Structures – a way of organizing and storing data in a computer so that it can be used efficiently

Directory – a virtual container used for organizing files and other directories on a computer’s file system

Draw by Agreement - Players may agree to a draw if neither player believes they can win.

Draw by Threefold Repetition - If the same position occurs three times with the same player to move, the game is drawn.

Draw by Fifty-Move Rule - If no pawn is moved and no capture is made in the last fifty moved by both players, the game is drawn.

En Passant- A special pawn capture that can occur when a pawn moves two squares forward from its starting position and lands next to an opponent’s pawn.

Graphical User Interface (GUI) – a type of user interface that allows users to interact with electronic devices using graphical icons and visual indicators, as opposed to text-based interfaces

Interface – a point where two systems, subjects, organizations, etc., meet and interact with each other

Linux Distribution – a complete operating system package that includes the Linux kernel along with various system utilities, libraries, software applications, and configuration files

Pseudo-GUI – interfaces that mimic the appearance or behavior of a true graphical user interface (GUI) but lack the full functionality or capabilities associated with traditional GUIs.

Sacrifice- Intentionally giving up material (pieces or pawns) in order to gain positional advantage or launch an attack

Software Architecture – the high-level structure of a software system, encompassing the design decisions that shape its behavior, functionality, and performance.

Stalemate – When the player who has to move has no more legal moves available. The game immediately ends in a tie, and each player gets half a point.

Utility – a program specifically designed to help manage and tune system or application software

Tutorial

Introduction:
Welcome to the tutorial for KANTEC Chess! This guide aims to provide you with a comprehensive understanding of how to use the program effectively, from starting a new game to interacting with the gameplay and navigating various end-game scenarios. Information regarding the different figures mentioned in this tutorial can be found in the Appendix.

“Chess is the struggle against error (Zukertort).”

Goals:
The primary goal of this manual is to assist users in understanding the functionalities of the KANTEC Chess. By the end of this tutorial, you should be able to navigate through the game seamlessly, make informed moves, and, most importantly, have fun with chess!

At this point, as our very interested, intelligent, and competitive player, you’re probably wondering, “Since this is a game, how am I supposed to win?” Well, great question! Your main goal as a player is to put the program in a “checkmate” (**Figure A4**). In this scenario, your opponent’s king is put in “check” (**Figure A3**). The difference is when a king is put in “checkmate”, there are no available moves to rescue the king from the “check”. Usually, when a king is put in “check”, there are three possible options to rescue the king:

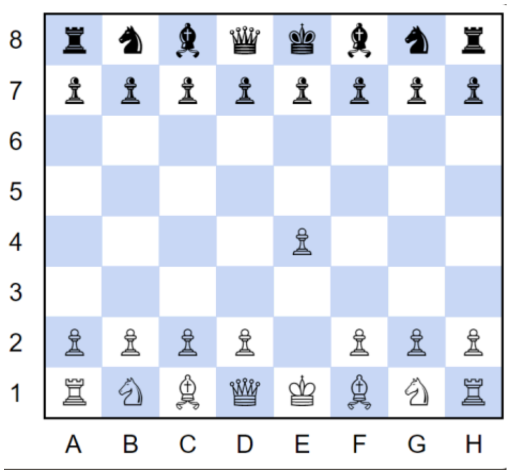
1. Using one of your pieces, capture the piece currently putting the king in check.
2. Place one of your pieces in front of the king to protect it from the opponent’s piece that is doing the check.
3. Move the king out of check.

Now that we’re familiar with the objectives of the game, let’s dive into how to play it in KANTEC Chess!

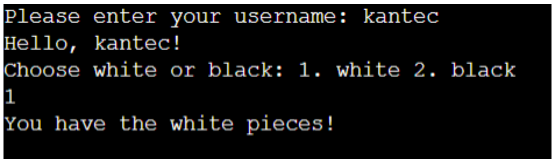
Usage Scenario:

Beginning the game

- Once you load KANTEC Chess, a standard chess board will be printed with all the pieces on top.



- From then on, the game will ask you to type in your name and your choice to be black or white.



```
Please enter your username: kantec
Hello, kantec!
Choose white or black: 1. white 2. black
1
You have the white pieces!
```

- Once you've made your selections, the turn-based sequence can begin. At any time during the game, you may enter "quit" to end the game.

During the Game

- To make a move, simply type in the square of the piece you want to move. For example, if you want to move a pawn on "A2", type in "A2" to start moving that pawn. This will not be case-sensitive, so you can type in "a2" as well.
- Keep in mind that you must enter a valid square letter and number to move a piece. If what you enter is out of what the chessboard supports (for example, "a9"), or if it's a square without a piece, the game will let you know what went wrong and prompt for a valid answer.
- Once a piece is selected, the game will ask if you want hints on how to move that piece. These hints will display the available move(s) that you have. If you choose to do so, you will then be able to see what square you can move to.
- Here is a list of all the legal moves in a standard game of chess:**(Figure A2)**
    - Pawn: **(Figure A2.2)**
        - Move forward one square.
        - On its first move, it can move forward two squares.
        - Capture diagonally one square forward.

- En-passant capture (special capture under specific conditions).
- Promotion (when reaching the opponent's back rank, it can be promoted to any other piece except a king).

- Knight: **(Figure A2.4)**
    - Move in an "L" shape, two squares in one direction (either horizontally or vertically) and then one square perpendicular to that direction.
    - Can jump over other pieces (and the only piece that can do so!)
- Bishop: **(Figure A2.5)**
    - Move diagonally any number of squares.
    - Each team has a black bishop and a white bishop. Black bishops will always stay on black squares, and white bishops will only stay on white squares due to their diagonal movements.
- Rook: **(Figure A2.1)**
    - Move horizontally or vertically any number of squares.
- Queen:**(Figure A2.6)**
    - Combines the moves of the bishop and rook, so it can move diagonally, horizontally, or vertically any number of squares.
- King: **(Figure A2.7)**
    - Move one square in any direction (horizontally, vertically, or diagonally).
    - Can castle with a rook under specific conditions (king hasn't moved, the rook involved hasn't moved, there are no pieces between the king and rook, and the king is not in check).
    - Castling:
        - The king moves two squares towards a rook on its initial square, then the rook moves to the square next to the king.
        - Can be kingside (short castling) or queenside (long castling).

- When you are moving your pieces, there are a couple things you cannot do. If your king is in “check”, you have to get out of check first. You cannot make other moves until you do so. Similarly, you cannot make a move that puts your king in “check”. Check figures **A3.3**, **A3.4**, and **A3.5** to see the different types of “checks”. In addition, you cannot move to a square that your pieces occupy.
- Once a destination square is selected, the game will print the modified board along with the move you just made. If you capture your opponent’s piece during a move or put them in check/checkmate, the game will also let you know.
- Once you make your move, your opponent will do the same, and the process repeats until a player gets a “checkmate”(**Figure A4**), “stalemate” (**Figure A5**), or types “quit”.

Ending the Game

- The game will print out the modified board.

- If there’s a winning player (which, hopefully, is you), the game will print out the name of the player.
- If player types “quit” or if it’s a “stalemate”, the game will print out the condition that the game ends with.
- At the end, you will choose to start a new game or to exit the game. Depending on your choice, the whole process will repeat.

Features:

Undo

What happens if you accidentally make a move you realized was a bad move? No worries! On the next move, type in “undo” instead of a new piece and you will be able to undo your previous move.

Black or white? The choice is yours!

As previously stated in the tutorial, every time you start a new game, you have a choice of going first (white) or going second (black). This is especially helpful for you to practice playing against both sides and prepare you for chess games in the real world.

Hints

Chess is hard! Sometimes, especially as a beginner player, you might need a little extra help when it comes to moving pieces. That is where the hints will come in handy. During the game, once you make an invalid move or request for hints, the game will show you available moves for the piece you selected.

Display of the Chess Board

At the beginning of the game, players will be able to see a standard 8x8 chessboard with ranks and files labeled. Each time a player makes a move, the game will display the board and the move that was made. This is an essential feature of the game that allows both players to clearly see what is happening and make moves accordingly. By the end, the modified chess board and the ending condition will be displayed as well.

A Standard Game

KANTEC chess supports the standard rules of chess, which means that special moves such as en passant, castling, and promotion are all allowed.

Move Log

Every time that a player makes a move, the game keeps track of said move in a text file. If desired, the game will print out the moves that were made as part of the hints provided for the player.

Timing

While the computer player does need some time to process your move and move accordingly, it shouldn’t take more than one minute to make a move.

Now that you've acquainted yourself with KANTEC chess, dive in and enjoy the game! If any queries pop up, consult this user manual for assistance and clarity. Best wishes and have a blast!

Installation

System Requirements

KANTEC Chess is compatible with Linux distributions that support Linux Kernel 4.14 or newer. Minimum recommended distributions include:

- Debian 9
- Ubuntu 18
- Fedora 27
- Red Hat Enterprise 7

Minimum recommended system hardware requirements:

- 10 MB disk space
- 1+ GB RAM

Setup and Configuration

Unpack the **Chess_V1.0.tar.gz** file using **gunzip** or a comparable utility.

Run **./KANTEC_Chess** to launch the game.

Each run of the game generates a **move_log.txt**.

Uninstalling

To uninstall KANTEC_Chess, run the **./uninstall** program from the game directory.

Documentation of functionality

KANTEC Chess follows the official rules of chess.

User Interface:

The visual interface of the game contains a textual representation of the chess board. The textual representation consists of ASCII characters.

Player versus Computer:

The KANTEC chess game entails a match between the user and the computer. Before commencing the game, the user will designate their preferred color, either white or black, thereby determining their side, while the opposing color will be assigned to the computer. A comprehensive log documenting each move executed by both parties will be maintained throughout the game. Furthermore, participants, both human and computer, shall take turns making decisions and executing moves, alternating between their respective chess pieces. Each side will be allocated a time frame of one minute to deliberate and execute their moves.

Additional Functions:

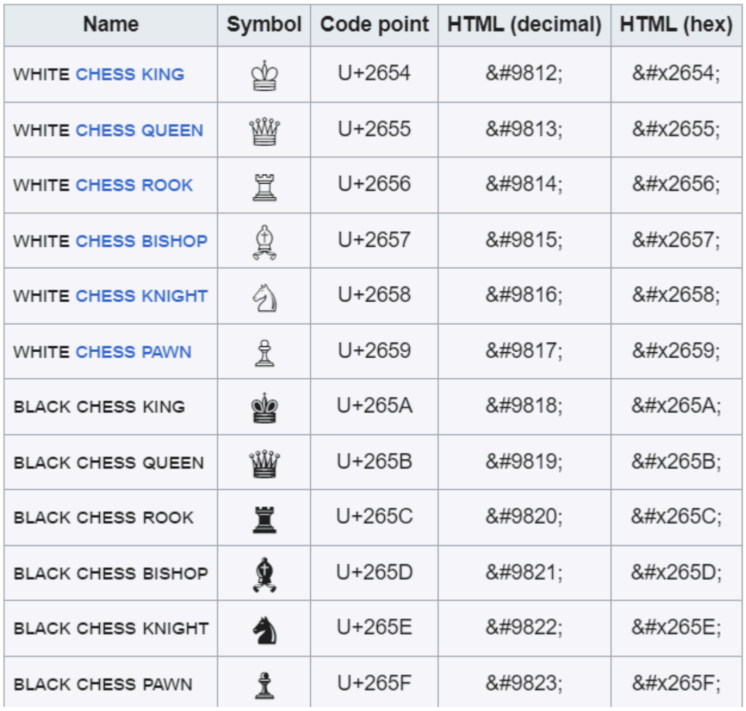
KANTEC Chess supports the official algebraic notation of chess moves.

The official algebraic notation of chess moves is a standardized system used to record and communicate chess games. In this notation:

1. Each square on the chessboard is identified by a unique coordinate consisting of a letter and a number.

2. The letters a-h represent the files (columns) from left to right, while the numbers 1-8 represent the ranks (rows) from bottom to top.

3. Pieces are denoted UNICode symbols as listed within the table:



Name	Symbol	Code point	HTML (decimal)	HTML (hex)
WHITE CHESS KING	♔	U+2654	&#9812;	&#x2654;
WHITE CHESS QUEEN	♕	U+2655	&#9813;	&#x2655;
WHITE CHESS ROOK	♖	U+2656	&#9814;	&#x2656;
WHITE CHESS BISHOP	♗	U+2657	&#9815;	&#x2657;
WHITE CHESS KNIGHT	♘	U+2658	&#9816;	&#x2658;
WHITE CHESS PAWN	♙	U+2659	&#9817;	&#x2659;
BLACK CHESS KING	♚	U+265A	&#9818;	&#x265A;
BLACK CHESS QUEEN	♛	U+265B	&#9819;	&#x265B;
BLACK CHESS ROOK	♜	U+265C	&#9820;	&#x265C;
BLACK CHESS BISHOP	♝	U+265D	&#9821;	&#x265D;
BLACK CHESS KNIGHT	♞	U+265E	&#9822;	&#x265E;
BLACK CHESS PAWN	♟	U+265F	&#9823;	&#x265F;

4. When a piece makes a move, the initial square it moves from is specified first, followed by the destination square. For example, if a knight moves from e2 to g3, it would be written as Ng3.

5. If two identical pieces can move to the same square, the file or rank of the starting square is included to differentiate them. For example, if there are two knights that can move to g3, and one is on d2 and the other on f2, their moves would be Nd2g3 and Nf2g3, respectively.

6. Captures are indicated by replacing the destination square with an 'x' followed by the piece captured. For example, if a pawn on e4 captures a piece on f5, it would be written as exf5.

7. Special moves are denoted by specific symbols: O-O for kingside castling, O-O-O for queenside castling, '+' for check, '#' for checkmate, '=' for pawn promotion, and 'e.p.' for en passant captures.

It is KANTEC UCI's goal to integrate our chess program to follow the standardized chess notations in order to keep our input and output systems universally recognizable. Additionally our ver 1.0.0 chess program will use a pseudo-GUI with the use of UNICode font symbolizations to output a mock-chessboard along with corresponding pieces (including color attributes) which we plan to later transition to a full GUI layout.

Troubleshooting/Error

Error Check:

Error checking will be implemented into the KANTEC chess via a set of assertion and abstract error checking functionalities. Error checking will occur in every aspect of implementation including, turn based errors, piece selection, valid movement check (legal move troubleshooting), capture piece error check, and endgame validity. The standard rules of chess will be implemented within our program with multiple error checks and troubleshooting steps in place to verify the chess program works as intended according to the standardized rules of chess.

Data structure preface: The Chessboard object will be a 2D array[8][8] each containing a pointer variable of *Piece datatype. The Piece datatype will be a structure that contains two enumeration-type members, a piece type and a piece color. Due to the elements of the array being composed of pointers, empty squares will consist of null pointers.

ListPieceMoves() Preface: There exists a ListPieceMoves() function for each PieceType as each piece has different rules for their movement. The ListXPieceMoves() function should be called in correspondence with the correct piece (ie. ListPawnMoves should be called when attempting to list a Pawn Pieces moves), and should be able to ascertain the pieces position (whether the Piece is passed in with a position member able to be accessed, or the position being passed in).

Turn based error Check:
Troubleshooting will be divided into multiple sets of validity checking functions. The chess program must use turn based alternation in order to instigate the progression of the game. Therefore when sequencing turn progression the functional loop that must be implemented must have an error check or strict implementation design such that each player may only make one move on their designated turn.

Piece Selection:
Once the player is executing their turn on their designated move two sets of error checks will occur. An initial check function will be performed CheckValidSquare() in order to check that the selected square has a piece present int it (Is the Piece* pointer NULL). If not NULL, perform a second check CheckValidTeam() in order to check that the selected piece belongs to the team of the player who is executing their turn.

Movement Sub-Check Functions:

- ListPieceMoves() - Lists the possible available movements for the selected Piece.
- IsLegal()- Checks the Legality of each movement. Illegal chess moves are subchecked within the following functions
    - KingInCheck()- Movements cannot be made that purposely place your king in check.
    - A king protection function will also be directly declared within IsLegal()

Legal Moves:
The LegalMoves() function will be implemented to calculate all the legal moves in a chess program that a player will be able to execute within their turn. This function will hash the board and identify the current player's pieces. Next it will generate all the possible moves for each piece (ListPawnMoves(), ListRookMoves, ListBishopMoves()... ListMoves() function will execute depending upon the piece type). Next the movements of each piece will be verified for legality and the illegal moves will be removed (IsLegalPawnMove(), IsLegalRookMove(), IsLegalBishopMove()... IsLegalMove() function will execute depending upon the piece type). Finally the list of legal moves available during this turn will be compiled in a digestible and formatted list and returned to the calling function.

- The main usage of this function will be used in conjunction with the CalculateBestMove() function which will have the list of legal moves passed into the function and will return the best move to the program.

Troubleshooting:

Should the program exit during the middle of the game in which the EndGameStatus() has not been reached yet and the player has not requested to exit() the program, a "Crash Error, Game has crashed on Turn X" Error will be output to the user. Additionally a troubleshooting message will be output to the user “Program Failed to Start, please revise installation” in the case that the program fails to start.

EndGameCondition:

The end game condition check will be a troubleshooting check that executes after each player's turn that checks if an end game condition has been met on the board. The end game conditions include a checkmate, a stalemate, timeout, resignation, or in advanced cases a repetition draw.

Program Functionality

It is KANTEC UCI’s goal to create a program which does not contain any launch or runtime errors to provide users with a smooth chess playing experience. In the name of transparency our error checking methods have been made open source as listed above in the spirit of program improvement for not just our chess program but every chess program such that we can create a future in which chess software can thrive. With the functionality of troubleshooting error messages and stages of validity checks placed between frictional functionality points, we hope to create a program that executes and runs correctly as intended by the rules and capabilities of standardized classic chess.

License Agreement and Disclaimer of Warranty

By using the Kantec Chess software ("the Software") provided by KANTEC UCI, you agree to be bound by the terms of this License Agreement and Disclaimer of Warranty. If you do not agree to these terms, you must not use the Software.

This document serves as a declaration of copyright ownership for the work titled KANTEC Chess, created by © KANTEC UCI on 2 April 2024.

License Agreement:

License Grant: KANTEC UCI grants you a non-exclusive, non-transferable license to use the Software for personal or commercial purposes, subject to the terms and conditions herein.

Restrictions: You may not sublicense, rent, lease, loan, sell, distribute, or create derivative works based upon the Software. You may not reverse engineer, decompile, or disassemble the Software, except to the extent such activities are expressly permitted by applicable law.

Ownership: KANTEC UCI retains all right, title, and interest in and to the Software, including all intellectual property rights therein.

Termination: This license is effective until terminated. Your rights under this license will terminate automatically without notice from KANTEC UCI if you fail to comply with any term(s) of this license.

Updates: KANTEC UCI may, from time to time, provide updates, upgrades, or new versions of the Software, which may be subject to additional terms and conditions.

Governing Law: This Agreement shall be governed by and construed in accordance with the laws of the Henry Samueli School of Engineering as well as standard California Law. Any dispute arising under this Agreement shall be resolved exclusively in the courts located within local California court jurisdiction.

Disclaimer of Warranty:

THE SOFTWARE IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. KANTEC UCI DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KANTEC UCI DOES NOT WARRANT THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE.

Limitation of Liability:

IN NO EVENT SHALL KANTEC UCI BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF KANTEC UCI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Acknowledgment:

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU FURTHER AGREE THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN YOU AND KANTEC UCI, WHICH SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER

COMMUNICATION BETWEEN YOU AND KANTEC UCI RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

References

Figure A1:
"Dicebreaker." *How to Play Chess | Dicebreaker*, 2022,
www.dicebreaker.com/games/chess/how-to/how-to-play-chess

Figure A2:
“Newafter.” *Chess Rules For Kids Pdf,*
https://r.search.yahoo.com/_ylt=AwrOuya6bAxmoMkI5kyjzbkF;_ylu=c2VjA2ZwLWF0dHJpYgRzbGsDcnVybA--/RV=2/RE=1712119098/RO=11/RU=https%3a%2f%2fnewafter954.weebly.com%2fchess-rules-for-kids-pdf.html/RK=2/RS=y.DUo_UKXWtv_sP3K3qZwIU.dr4-

Figure A3 and A4:
"How to Checkmate Your Opponent." RagChess, September 12, 2018,
https://www.ragchess.com/how-to-checkmate-your-opponent/

Appendix

Figure A1: Chess Chess Board

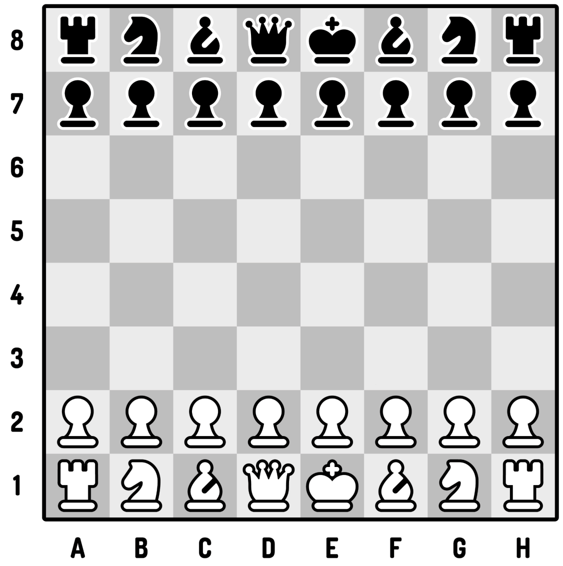


Figure A1.1: Rook

Figure A1.2: Pawn

Figure A1.3: Knight

Figure A1.4 Bishop

Figure A1.5: Queen

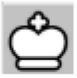
**Figure A1.6**: King

Figure A2: Chess Piece Movements

<u>**Rook**</u>:

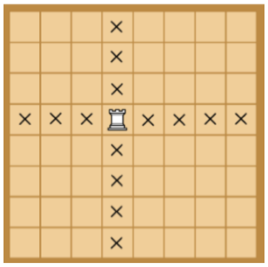


Figure A2.1: can move in all empty spaces in four straight lines

Pawn:

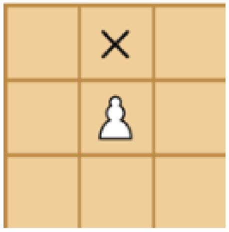


Figure A2.2: Move: one space forward only

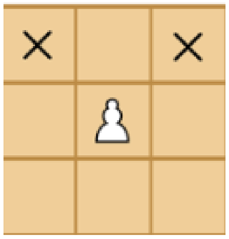


Figure A2.3: Capture: one space diagonally forward

Knight:

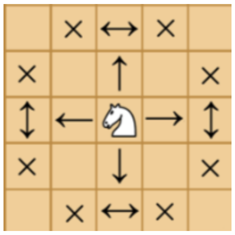


Figure A2.4: Leap two spaces obliquely (in an L-shaped path)

Bishop:

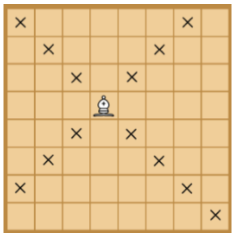


Figure A2.5: All empty spaces in four diagonals

Queen:

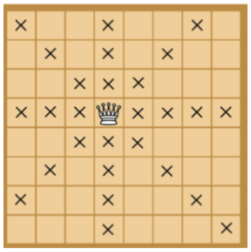


Figure A2.6: all empty spaces in any direction

King:

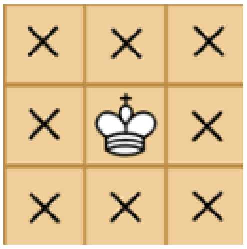


Figure A2.7: only one space in any direction

Figure A3: Different chess pieces delivering a check

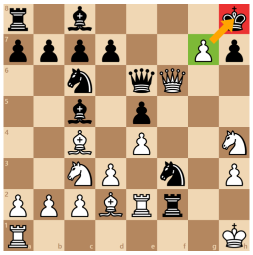


Figure A3.1: A pawn delivering a check

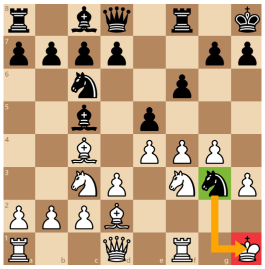


Figure A3.2: A knight delivering a check

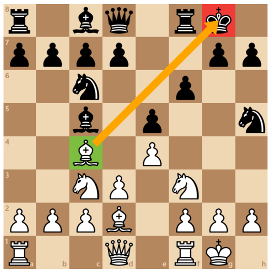


Figure A3.3: A bishop delivering a check

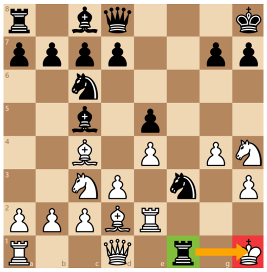


Figure A3.4: A rook delivering a check

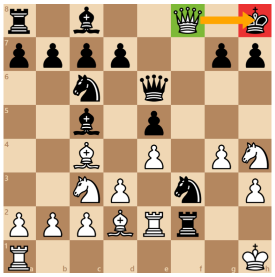


Figure A3.5: A queen delivering a check

Figure A4: Scenarios for a checkmate

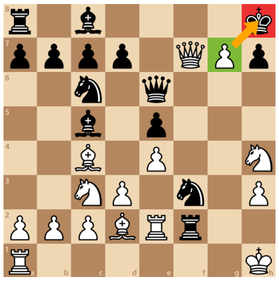


Figure A4.1: A pawn delivering a checkmate, despite the board being almost full of chess pieces. The king is under attack as it can't move and can't recapture the pawn because the queen protects it

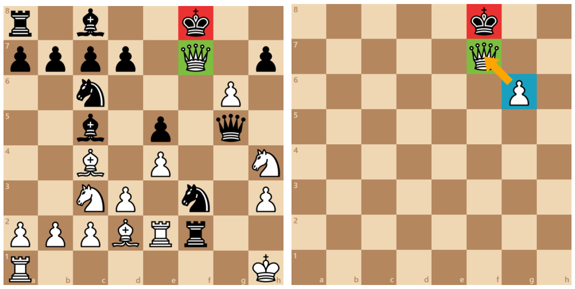
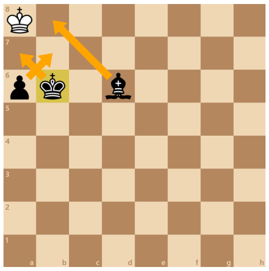
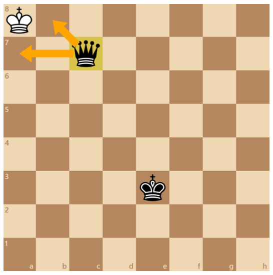


Figure A4.2 & A4.3: This is a scenario where a checkmate includes a sacrifice of strong pieces just to end the game. Scenario: Black is able to checkmate white in one move, however, white moved first and played Qf7, checkmate. The pawn on g6 and the bishop on c4 protect the queen from the king's recapture. Figure A4.3 shows the same checkmate scenario with the rest of the pieces removed from the board.

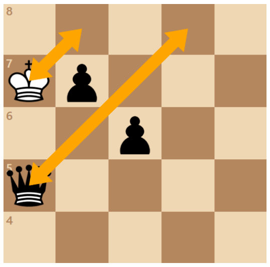
Figure A5: Scenarios of a Stalemate



<u>Figure A5.1</u>: White’s king cannot move



<u>Figure A5.2</u>: The queen removes the king’s safe squares, and the king is not in check



<u>Figure A5.3</u>: Draw by representation as the king and queen will move back and forth. No other moves are possible for white.

Title page (10 points)✅

- Software title, version✅
- Author/producer, affiliation✅

Front matter (20 points)

- Table of contents
- Glossary (of chess terms)

1 Computer Chess (20 points)

1.1 Usage scenario

- Sketch of a typical screenshot

1.2 Goals

1.3 Features

2 Installation (15 points)✅

2.1 System requirements✅

2.2 Setup and configuration✅

2.3 Uninstalling✅

3 Chess Program Functions and Features (20 points)

3.1 Detailed description of function 1

- User input, program output, screen shot, etc.

3.2 Detailed description of function 2

- User input, program output, screen shot, etc.

3.3 …

Back matter (15 points)

- Copyright✅
- Error messages✅
- Index