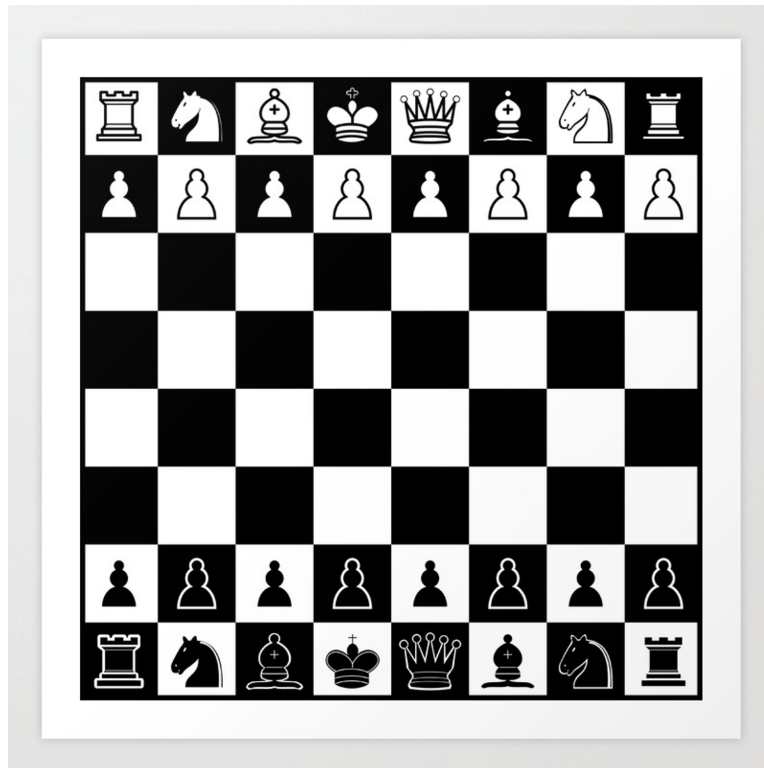# r/chess

## User Manual



**Authors: Joseph Balardeta, Kevin Lee, Ryan Kim, Li-Yu Ho, Aidan Woods**

**Software Version: 1.0 (OFFICIAL RELEASE)**

**Affiliation: r/chess Team**

# Table of Contents

# Glossary of Chess Terms

## The Pieces

- The queen can move horizontally, vertically, and diagonally across the board.
- A rook can move horizontally and vertically across the board.
- A bishop can move diagonally across the board.
- A knight can jump to eight different squares which are two steps forward plus one step sideways from its current position.
- The king can move in any direction, but only one step at a time. Also, the king must never move into check.
- A pawn can move only forward towards the end of the board, but captures sideways. From its initial position, a pawn may make two steps, otherwise only a single step at a time. If the pawn reaches the end of the board, it is automatically promoted to another piece (usually a queen).

## Other Terms

- Check - Check is a condition that occurs when a player's king can be captured by the opponent on the next turn. A player must get out of check in the next move.
- Checkmate - Checkmate occurs when a player's king is checked and there is no possible move for the king to escape the check. If a player's king is checkmated, the player loses the game.
- File - Each column on the chess board is called a file.
- Rank - Each row on the chess board is called a rank.
- Castle - Castling allows the two pieces, the king and a rook to move at the same time in one move. The king can either move two spaces to the left or to the right while the rook desired to perform castling jumps across to the other side of the king, ending up being right next to the king. In order to castle though,
    - The king and the rook you want to castle with cannot have moved.
    - There must not be any pieces in between the king and the rook.
    - The king cannot be in check.
    - The king cannot be in check after castling has taken place.
- En passant - May occur right after one player moves a pawn two squares forward and the opposing pawn could have captured the same pawn had it moved one square forward.
- Promotion - A pawn can be promoted to any other piece when it gets to its last rank.

# Computer Chess

## Usage Scenario

```
   +----+----+----+----+----+----+----+----+
8  | bR | bN | bB | bQ | bK | bB | bN | bR |
   +----+----+----+----+----+----+----+----+
7  | bP | bP | bP | bP | bP | bP | bP | bP |
   +----+----+----+----+----+----+----+----+
6  |    |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
5  |    |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
4  |    |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
3  |    |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
2  | wP | wP | wP | wP | wP | wP | wP | wP |
   +----+----+----+----+----+----+----+----+
1  | wR | wN | wB | wQ | wK | wB | wN | wR |
   +----+----+----+----+----+----+----+----+
     A    B    C    D    E    F    G    H
[Enter the position of the piece you want to move: a2

[Enter the new position you want your piece to move to: a3
   +----+----+----+----+----+----+----+----+
8  | bR | bN | bB | bQ | bK | bB | bN | bR |
   +----+----+----+----+----+----+----+----+
7  | bP | bP | bP | bP | bP | bP | bP | bP |
   +----+----+----+----+----+----+----+----+
6  |    |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
5  |    |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
4  |    |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
3  | wP |    |    |    |    |    |    |    |
   +----+----+----+----+----+----+----+----+
2  |    | wP | wP | wP | wP | wP | wP | wP |
   +----+----+----+----+----+----+----+----+
1  | wR | wN | wB | wQ | wK | wB | wN | wR |
   +----+----+----+----+----+----+----+----+
     A    B    C    D    E    F    G    H
[Enter the position of the piece you want to move: ▌
```

The program's ideal scenario is for people to play chess against a computer, and for their games to be improved after multiple playthroughs.

**Example of usage**: One is preparing for a chess tournament and wishes to practice with a computer or observe potential moves while watching two computers play against one another.

## Goals

- Fully playable game of chess with official rules
- AI that makes logical (and legal) moves

## Features

- Full chess board with moveable pieces
- Interactive main menu that presents the user with options
- Selectable black and white sides
- Player VS CPU
- CPU VS CPU
- Quick CPU (makes a move in under 60 seconds)
- Accessible log file with all made moves (.txt file)

# Installation

## System Requirements

- Linux OS running gcc version 1.7.1 or newer
- 50mb of disk space
- 50kb of free RAM

## Setup and Installation

To set up r/chess, Type 'tar -xvzf Chess V1.0 src.tar.gz', then 'cd Chess V1.0 src'. Launch the "r_chess" executable file in order to launch the game.

## Uninstalling

To uninstall the program, launch the "r_chess" executable file. Then in the menu, select the option that says "Uninstall" in order to uninstall and exit the program.

# Program Functions and Features

## int main()

- Shows the menu that prompts the user for actions they want to do.
  - 5 different options.
- Shows an interface on which the user sees the chess board and makes moves.
- Runs the chess game by repeating functions using while loops until a side wins or the user quits the game.

## void move(Game *game, MOVE *move)

- Allows a player's piece to move and change its position on the board.
- Updates a player's piece position on the board.
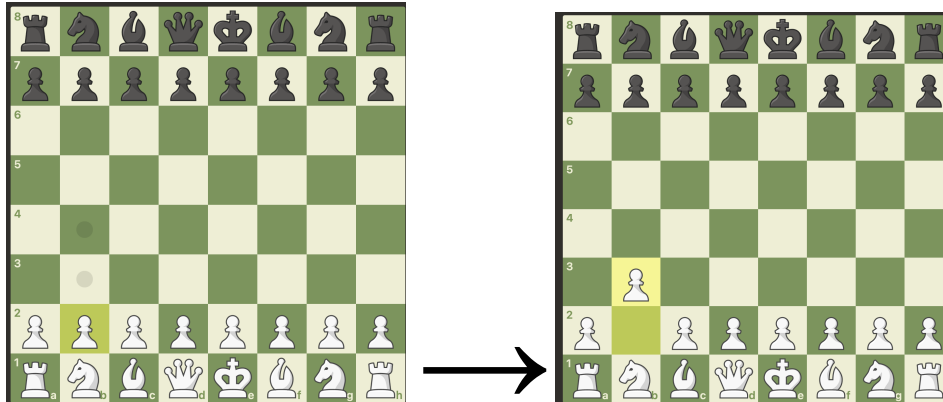
# int isLegalMove(Game *game, MOVE *move)

- A boolean function that checks if a piece's desired move is legal or not.
- Returns 0 if the move is illegal, returns 1 if the move is legal.



# int isInCheck(Game *game)

- A boolean function that checks if a player's king is in check.
- If a player's king is in check, then the player must move its king or eliminate the opponent's piece that is causing the king to be in check.
- Returns 0 if the king is not in check, returns 1 if the king is in check.

 (black bishop is checking white king)

# int isCheckmate(Game *game)

- A boolean function that checks if a player's king is in checkmate.
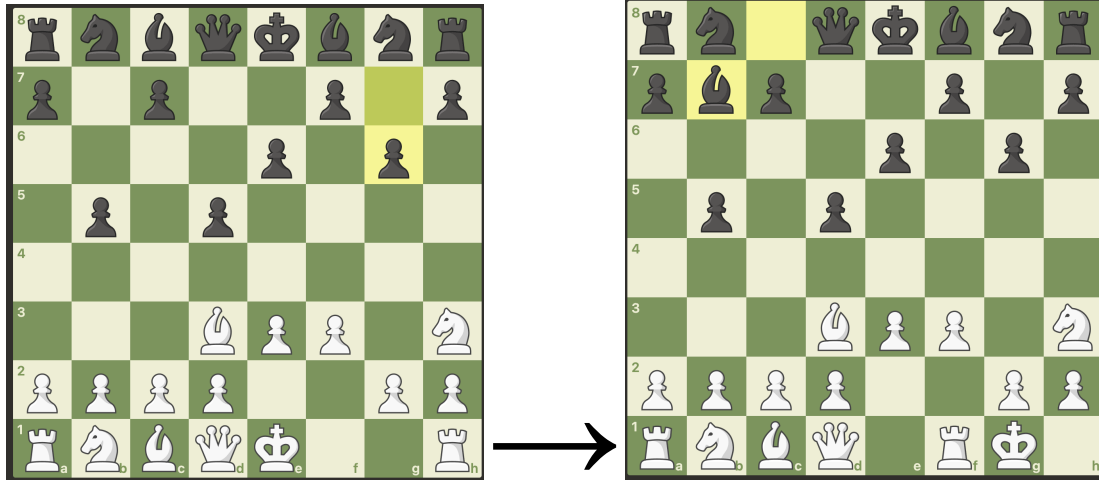- Returns 0 if the king is not in checkmate, returns 1 if the king is in checkmate.



# int isValidPieceMove(Game *game, MOVE *move)

- A boolean function that checks if a player's wanted piece is able to move in the wanted way.
- Deals with capturing, special moves (starting pawns, en passant, castling).
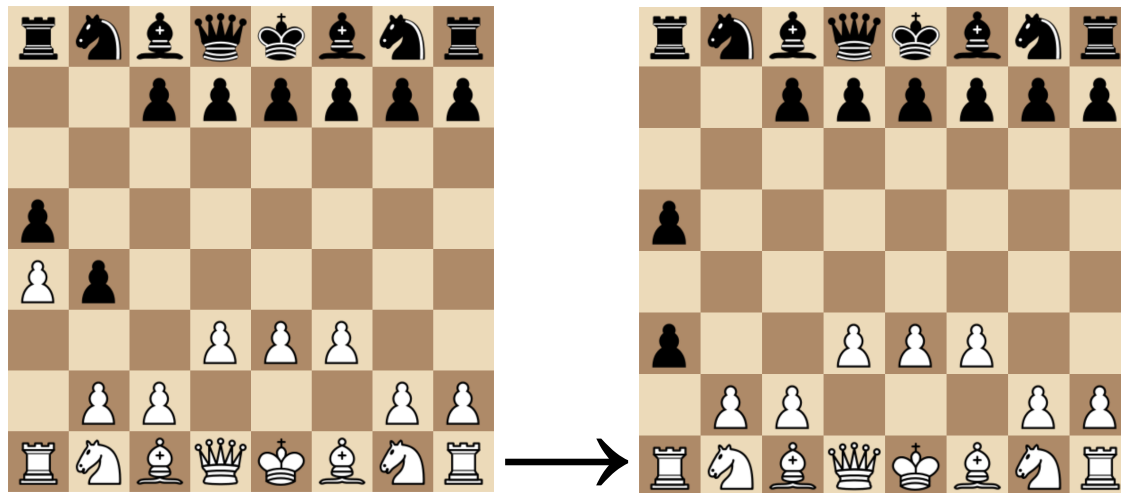
(castling)
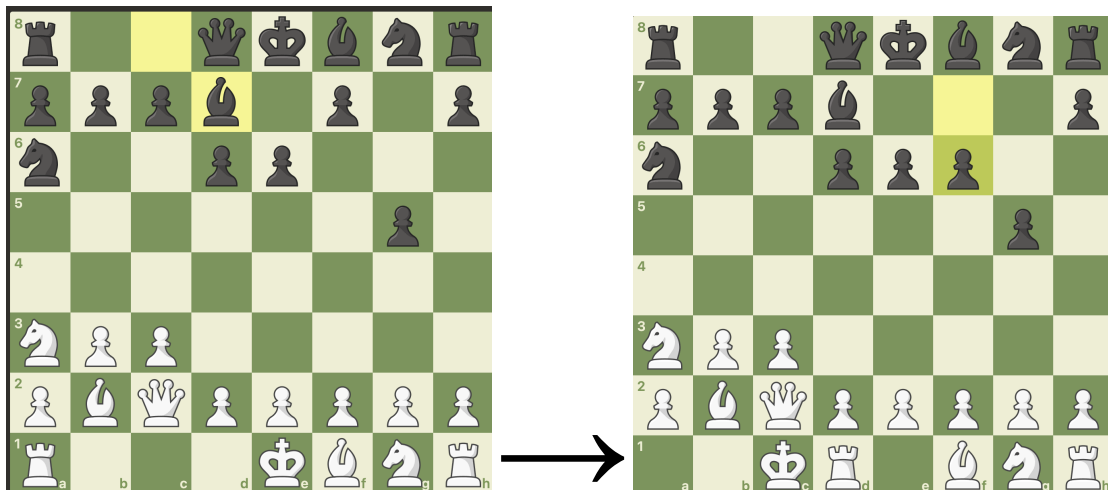


(King moves two spaces to the left and the rook is moved to the king's right)

(King moves two spaces to the right and the rook is moved to the king's left)

(en passant)

## Other Functions

## int isObstructed(Game *game, MOVE *move)
- A boolean function that checks if a player's wanted piece is able to move with the current position of other pieces.

## void checkPromotions(Game *game)
- A function that checks for all possible promotions in every instance.

## int exposesKing(Game *game, MOVE *move)
- A boolean function that checks for pieces that may threaten the king.

# Main Menu Options

- Option for Player VS CPU
  - Option for the player to choose sides (black or white, once the player has chosen either the first or second option)
- Option for CPU VS CPU
- Option for the player to quit the game
- Option for the player to uninstall the game and to exit the program
- Option for the player to access a readable log of all moves made during the game (via a text file)

# License and Disclaimers

**\*Presented images may not represent the actual program\***

# Error Messages

## On making an erroneous selection in the menu:

"That is not a valid option!" then a repeat of the menu selections.

## On making an illegal move during a game:

"Move not valid!" will be prompted to the user and will be asked to enter another move.

## On trying to open a log file that does not exist:

"Log file does not exist" message will be prompted to the user.

## On choosing an empty piece on the board to move:

"ERROR - Attempting to move a NULL piece!" will be warned to the player.

# Index