1. **Title of the Project :** Chess Game
2. **About Chess Game :**

**Chess** is a [board game](https://simple.wikipedia.org/wiki/Board_game) for two players. It is played in a [square](https://simple.wikipedia.org/wiki/Square_(geometry)) [board](https://simple.wikipedia.org/wiki/Board_(wood)), made of 64 smaller squares, with eight squares on each side. Each player starts with sixteen [pieces](https://simple.wikipedia.org/wiki/Chess_piece): eight [pawns](https://simple.wikipedia.org/wiki/Pawn_(chess)), two [knights](https://simple.wikipedia.org/wiki/Knight_(chess)), two [bishops](https://simple.wikipedia.org/wiki/Bishop_(chess)), two [rooks](https://simple.wikipedia.org/wiki/Rook_(chess)), one [queen](https://simple.wikipedia.org/wiki/Queen_(chess)) and one [king](https://simple.wikipedia.org/wiki/King_(chess)). The goal of the game is for each player to try and [checkmate](https://simple.wikipedia.org/wiki/Checkmate) the king of the opponent. Checkmate is a threat ('check') to the opposing king which no move can stop. It ends the game.

During the game the two opponents take turns to move one of their pieces to a different square of the board. One player ('White') has pieces of a light [color](https://simple.wikipedia.org/wiki/Color); the other player ('Black') has pieces of a dark color. There are rules about how pieces move, and about taking the opponent's pieces off the board. The player with white pieces always makes the first move. Because of this, White has a [small advantage](https://simple.wikipedia.org/wiki/First-move_advantage), and wins more often than Black in tournament games.

**3.** **Rules:**

The rules of chess are governed by the [World Chess Federation](https://simple.wikipedia.org/wiki/FIDE), which is known by the initials FIDE, meaning *Fédération Internationale des Échecs*. The rules are in the section *Laws of Chess* of the *FIDE Handbook*. FIDE also give rules and guidelines for [chess tournaments](https://simple.wikipedia.org/wiki/Chess_tournament)

### **Setup**

Chess is played on a [square](https://simple.wikipedia.org/wiki/Square_(geometry)) board divided into [eight](https://simple.wikipedia.org/wiki/Eight) rows of squares called [ranks](https://simple.wikipedia.org/wiki/List_of_chess_terms#R) and eight columns called [files](https://simple.wikipedia.org/wiki/List_of_chess_terms#F), with a dark square in each player's lower left corner. This is altogether 64 squares. The colors of the squares are laid out in a checker (chequer) pattern in light and dark squares. To make speaking and writing about chess easy, each square has a name. Each rank has a number from 1 to 8, and each file a letter from *a* to *h*. This means that every square on the board has its own label, such as g1 or f5. The pieces are in [white](https://simple.wikipedia.org/wiki/White) and [black](https://simple.wikipedia.org/wiki/Black) sets. The players are called White and Black, and at the start of a game each player has 16 pieces. The 16 pieces are one king, one queen, two rooks, two bishops, two knights and eight pawns.in this game out can get up to a quadruple pawn, king, knight, queen, and also the king although it is very rare.

### **Movement**

Definitions: vertical lines are *files*; horizontal lines are *ranks*; lines at 45° are [*diagonals*](https://simple.wikipedia.org/wiki/Diagonal). Each piece has its own way of moving around the board. The *X* marks the squares where the piece can move.

* The knight is the only piece that can jump over another piece.
* No piece may move to a square occupied by a piece of the same color.
* All pieces capture the same way they move, except pawns.

**Moves of the king**  **Moves of the rook**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Start of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |
| https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png | a8 __ | b8 __ | c8 __ | d8 __ | e8 __ | f8 __ | g8 __ | h8 __ | https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png |
| a7 __ | b7 __ | c7 __ | d7 __ | e7 __ | f7 __ | g7 __ | h7 __ |
| a6 __ | b6 __ | c6 __ | d6 __ | [e6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [f6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [g6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | h6 __ |
| a5 __ | b5 __ | c5 __ | d5 __ | [e5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [f5 white king](https://simple.wikipedia.org/wiki/File:Chess_kll44.png) | [g5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | h5 __ |
| a4 __ | b4 __ | c4 __ | d4 __ | [e4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [f4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [g4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | h4 __ |
| a3 __ | b3 __ | c3 __ | d3 __ | e3 __ | f3 __ | g3 __ | h3 __ |
| a2 __ | b2 __ | c2 __ | d2 __ | e2 __ | f2 __ | g2 __ | h2 __ |
| a1 __ | b1 __ | c1 __ | d1 __ | e1 __ | f1 __ | g1 __ | h1 __ |
| [End of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Start of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |
| https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png | a8 __ | b8 __ | c8 __ | [d8 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | e8 __ | f8 __ | g8 __ | h8 __ | https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png |
| a7 __ | b7 __ | c7 __ | [d7 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | e7 __ | f7 __ | g7 __ | h7 __ |
| a6 __ | b6 __ | c6 __ | [d6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | e6 black king | f6 black king | g6 black king | h6 __ |
| [a5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [b5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [c5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [d5 black rook](https://simple.wikipedia.org/wiki/File:Chess_rdl44.png) | [e5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [f5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [g5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [h5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) |
| a4 __ | b4 __ | c4 __ | [d4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | e4 black king | f4 black king | g4 black king | h4 __ |
| a3 __ | b3 __ | c3 __ | [d3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | e3 __ | f3 __ | g3 __ | h3 __ |
| a2 __ | b2 __ | c2 __ | [d2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | e2 __ | f2 __ | g2 __ | h2 __ |
| a1 __ | b1 __ | c1 __ | [d1 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | e1 __ | f1 __ | g1 __ | h1 __ |
| [End of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Start of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |
| https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png | [a8 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | b8 __ | c8 __ | d8 black king | e8 __ | f8 __ | [g8 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | h8 __ | https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png |
| a7 __ | [b7 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | c7 __ | d7 black king | e7 __ | [f7 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | g7 __ | h7 __ |
| a6 __ | b6 __ | [c6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | d6 black king | [e6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | f6 black king | g6 black king | h6 __ |
| a5 black king | b5 black king | c5 black king | [d5 white bishop](https://simple.wikipedia.org/wiki/File:Chess_bll44.png) | e5 black king | f5 black king | g5 black king | h5 black king |
| a4 __ | b4 __ | [c4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | d4 black king | [e4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | f4 black king | g4 black king | h4 __ |
| a3 __ | [b3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | c3 __ | d3 black king | e3 __ | [f3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | g3 __ | h3 __ |
| [a2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | b2 __ | c2 __ | d2 black king | e2 __ | f2 __ | [g2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | h2 __ |
| a1 __ | b1 __ | c1 __ | d1 black king | e1 __ | f1 __ | g1 __ | [h1 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) |
| [End of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |

**Moves of the bishop Moves of the queen**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Start of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |
| https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png | a8 black king | b8 __ | c8 __ | [d8 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | e8 __ | f8 __ | g8 black king | [h8 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png |
| [a7 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | b7 __ | c7 __ | [d7 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | e7 __ | f7 __ | [g7 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | h7 __ |
| a6 __ | [b6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | c6 __ | [d6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | e6 __ | [f6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | g6 __ | h6 __ |
| a5 __ | b5 __ | [c5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [d5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [e5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | f5 black king | g5 black king | h5 __ |
| [a4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [b4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [c4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [d4 black queen](https://simple.wikipedia.org/wiki/File:Chess_qdd44.png) | [e4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [f4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [g4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [h4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) |
| a3 __ | b3 __ | [c3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | [d3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [e3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | f3 black king | g3 black king | h3 __ |
| a2 __ | [b2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | c2 __ | [d2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | e2 __ | [f2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | g2 __ | h2 __ |
| [a1 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | b1 __ | c1 __ | [d1 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | e1 __ | f1 __ | [g1 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | h1 __ |
| [End of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |

**Moves of the knight** **Moves of the pawn**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Start of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |
| https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png | a8 black king | b8 __ | c8 __ | d8 black king | e8 __ | f8 __ | g8 black king | h8 __ | https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png |
| a7 black king | b7 __ | c7 __ | d7 black king | e7 __ | f7 __ | g7 black king | h7 __ |
| a6 __ | b6 black king | [c6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | d6 black king | [e6 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | f6 black king | g6 __ | h6 __ |
| a5 __ | [b5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | c5 black king | d5 black king | e5 black king | [f5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | g5 black king | h5 __ |
| a4 black king | b4 black king | c4 black king | [d4 black knight](https://simple.wikipedia.org/wiki/File:Chess_ndd44.png) | e4 black king | f4 black king | g4 black king | h4 black king |
| a3 __ | [b3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | c3 black king | d3 black king | e3 black king | [f3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | g3 black king | h3 __ |
| a2 __ | b2 black king | [c2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | d2 black king | [e2 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | f2 black king | g2 __ | h2 __ |
| a1 black king | b1 __ | c1 __ | d1 black king | e1 __ | f1 __ | g1 black king | h1 __ |
| [End of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Start of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |
| https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png | a8 black king | b8 __ | c8 __ | [d8 white circle](https://simple.wikipedia.org/wiki/File:Chess_ood44.png) | [e8 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [f8 white circle](https://simple.wikipedia.org/wiki/File:Chess_ood44.png) | g8 black king | h8 __ | https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png |
| a7 black king | b7 __ | c7 __ | d7 black king | [e7 white pawn](https://simple.wikipedia.org/wiki/File:Chess_pld44.png) | f7 __ | g7 black king | h7 __ |
| a6 __ | b6 black king | c6 black king | d6 black king | e6 black king | f6 black king | g6 __ | h6 __ |
| [a5 white circle](https://simple.wikipedia.org/wiki/File:Chess_ood44.png) | [b5 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [c5 white circle](https://simple.wikipedia.org/wiki/File:Chess_ood44.png) | d5 black king | e5 black king | f5 black king | g5 black king | h5 __ |
| a4 black king | [b4 white pawn](https://simple.wikipedia.org/wiki/File:Chess_pld44.png) | c4 black king | d4 black king | e4 black king | [f4 cross](https://simple.wikipedia.org/wiki/File:Chess_xxd44.png) | g4 black king | h4 black king |
| a3 __ | b3 black king | c3 black king | d3 black king | [e3 white circle](https://simple.wikipedia.org/wiki/File:Chess_ood44.png) | [f3 cross](https://simple.wikipedia.org/wiki/File:Chess_xxl44.png) | [g3 white circle](https://simple.wikipedia.org/wiki/File:Chess_ood44.png) | h3 __ |
| a2 __ | b2 black king | c2 black king | d2 black king | e2 black king | [f2 white pawn](https://simple.wikipedia.org/wiki/File:Chess_pld44.png) | g2 __ | h2 __ |
| a1 black king | b1 __ | c1 __ | d1 black king | e1 __ | f1 __ | g1 black king | h1 __ |
| [End of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |

* The **king'**s move is one square in any direction. The king (K for short) may not move to any square where it is *threatened* by an opposing piece. However, the king can move to a square that is *occupied* by an opponent's piece and capture the piece, taking it off the board.
* The **queen** (Q) can move any distance in any direction on the ranks, files and diagonals.
* The **rooks** (R) move any distance on the *ranks* or *files*
* The **bishops** (B) move diagonally on the board. Since a bishop can only move diagonally, it will always be on the same color square
* The **knights** (Kt or N) move in an "L" shape. Each move must be either two squares along a rank and one square along a file, or two squares along a file and one square along a rank. It is the only piece that can jump over other pieces. Like the other pieces, it captures an opposing piece by landing on its square.
* The **pawns** can only move up the board. On its first move a pawn may move either one or two squares forward. A pawn captures one square diagonally, not as it moves: see white circles on its diagram. Besides, in some situations pawns can capture opponent's pawns in a special way called *en passant*, which means *in passing* in [French](https://simple.wikipedia.org/wiki/French_(language)) (see below)

### **Capturing**

Most pieces capture as they move. If a piece lands on an opponent's piece, the opposing piece is taken off the board. There are three special cases:

1. The king cannot be taken (see check and checkmate).
2. No piece can be taken while *castling* (see below).
3. Pawns take one square diagonally.

[**Check and Checkmate**](https://simple.wikipedia.org/wiki/Check_and_checkmate)

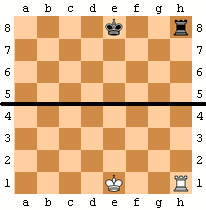
[](https://simple.wikipedia.org/wiki/File:Checkmate.jpg)

An example of checkmate

If a move is made which attacks the opposing king, that king is said to be 'in check'. The player whose king is checked must make a move to remove the check. The options are: moving the king, capturing the threatening piece, or moving another piece between the threatening piece and the king.If the player whose king is in danger cannot do any of these things, it is [checkmate](https://simple.wikipedia.org/wiki/Checkmate), and the player loses the game

**Special Moves:**

### **Castling**

[](https://simple.wikipedia.org/wiki/File:ChessCastlingMovie.gif)

A chess castling move

Once in every game, each king can make a special move, known as [*castling*](https://simple.wikipedia.org/wiki/Castling). When the King castles, it moves two squares to the left or right. When this happens, the Rook is moved to stand on the opposite side of the King. Castling is only allowed if all of these rules are kept:

* *Neither piece doing the castling may have been moved during the game*.
* There must be no pieces between the king and the rook.
* The king may not be currently in [check](https://simple.wikipedia.org/wiki/Check_(chess)), nor may the king pass through any square attacked by the opponent. As with any move, castling is not allowed if it would place the king in check

### **En passant**

En passant

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Start of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |
| https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png | a8 __ | b8 __ | c8 __ | d8 __ | e8 __ | f8 __ | g8 __ | h8 __ | https://upload.wikimedia.org/wikipedia/commons/4/4e/Chess_zver_22.png |
| a7 __ | b7 __ | c7 __ | d7 __ | e7 __ | f7 __ | g7 __ | h7 __ |
| a6 __ | b6 __ | c6 __ | d6 __ | e6 __ | f6 __ | [g6 white circle](https://simple.wikipedia.org/wiki/File:Chess_ool44.png) | h6 __ |
| a5 __ | b5 __ | c5 __ | d5 __ | e5 __ | [f5 white pawn](https://simple.wikipedia.org/wiki/File:Chess_pll44.png) | [g5 black pawn](https://simple.wikipedia.org/wiki/File:Chess_pdd44.png) | h5 __ |
| a4 __ | b4 __ | c4 __ | d4 __ | e4 __ | f4 __ | g4 __ | h4 __ |
| a3 __ | b3 __ | c3 __ | d3 __ | e3 __ | f3 __ | g3 __ | h3 __ |
| a2 __ | b2 __ | c2 __ | d2 __ | e2 __ | f2 __ | g2 __ | h2 __ |
| a1 __ | b1 __ | c1 __ | d1 __ | e1 __ | f1 __ | g1 __ | h1 __ |
| [End of chess board.](https://simple.wikipedia.org/wiki/File:Chess_zhor_22.png) | | | | | | | | | |

*En passant* ('in passing' in [French](https://simple.wikipedia.org/wiki/French_(language))) is a special capture. It is only available when a pawn moves forward two squares past an opposing pawn on an [adjacent](https://simple.wiktionary.org/wiki/adjacent) file. The opposing pawn must be on the 5th rank from its own side. Then the opponent's pawn can capture the double-mover *as if it had only moved one square forward*. This option is open on the next move only.

For example, if the black pawn has just moved up two squares from g7 to g5, then the white pawn on f5 can take it by *en passant* on g6. The *en passant* rule was developed when pawns were allowed to make their double move. The rule made it more difficult for players to avoid pawn exchanges and blockade the position. It kept the game more open.

### **Promotion**

When a pawn moves to its eighth rank, it must be changed for a piece: a queen, rook, bishop, or knight of the same color (player's choice).Normally, the pawn is *queened*, but in some advantageous cases another piece is chosen, called 'under-promotion'.

## Ways a game may end

Checkmates are rare in competitive chess. The most common ends are decisions made by one or both players.

### **Wins**

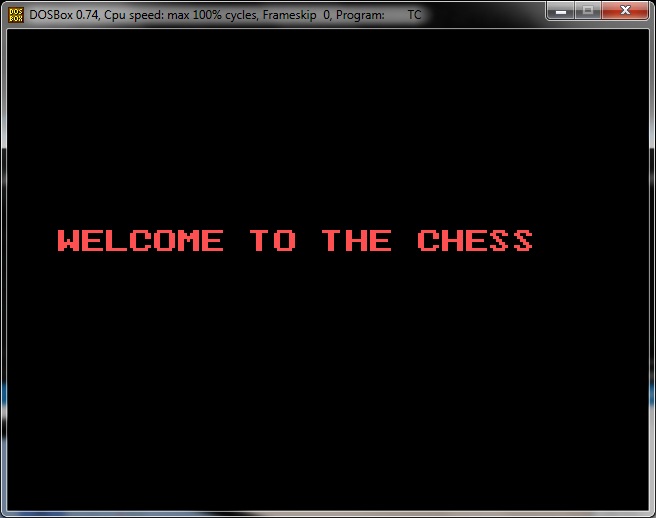
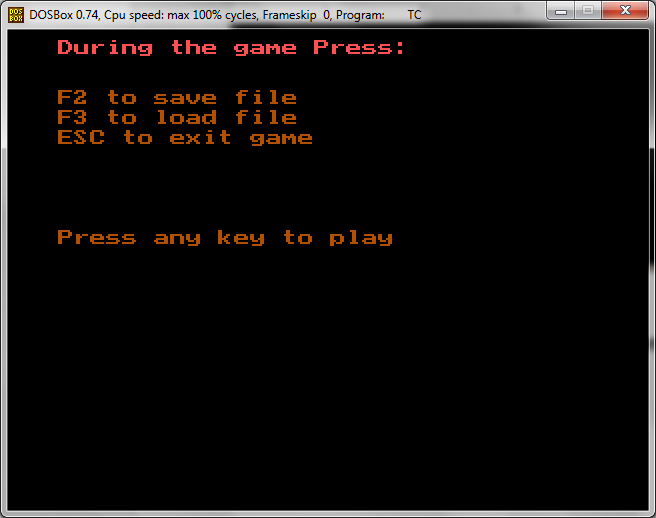
* **Checkmate**. When a king is in check, and cannot get out of it.
* **Resignation**. A player may resign at any time, usually because his/her position is hopeless. A losing player is able to resign by placing their king on its side on the chessboard.
* **Out of time**. If player's clock time is over (exceeding the time control). Strictly speaking, this is not part of the rules of the game, but part of the rules of tournament and match chess where chess clocks are used.

### **Draws**

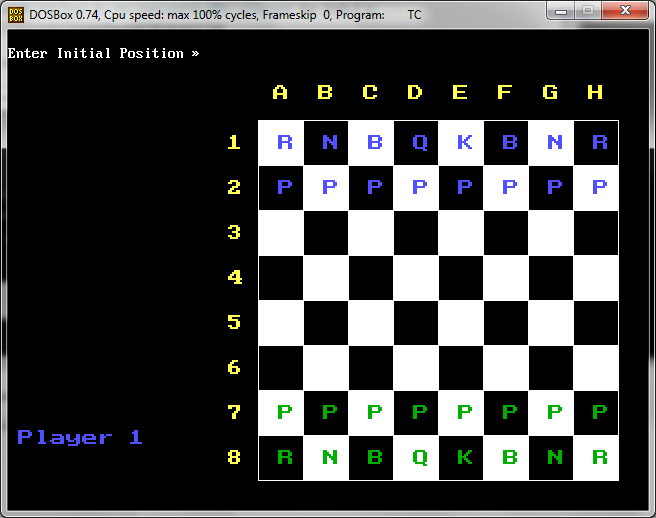
* **Draw agreed**. A game may end in a draw at any time if one player offers a draw and the other accepts.
* **Dead position**. A position where no series of legal moves could lead to a mate (example: K+B vs K). The game is drawn.
* **Stalemate**. If a player cannot make a move, and the player's king is not in check, this is also a draw. This kind of draw is called a *stalemate*, and is rare.
* **50-move rule**. A game will also end in a draw if no piece is captured and no pawn has moved after fifty moves. This is called the *fifty-move rule*, and happens late in the game.
* **Threefold repetition**. If the exactly same position is repeated three times during a game with the same player to move each time, the player next to move may claim a draw. The game is now drawn. This is called a *draw by threefold repetition*

## Display of game

**Welcome Screen Save/load file screen**

**** 

**ChessBoard Screen**



**Introduction**

There are lots of implementations of Chess' games available on the internet, most of them richer in features than this one. Nevertheless, there's no demerit on developing a simpler, lightweight piece of software, specially aiming for didactic purposes.

What this game is (or tries to be):

* Lightweight. The size of the version 1.0 of the application is 15.1 KB
* Implemented [fully on console](https://en.wikipedia.org/wiki/Console_application)

What this game **is not/does not have**:

* Does have a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) & Does not have artificial intelligence (AI)

## Background

This game runs in a console, i.e., that means no GUI is available to the user. All the input is taken from the keyboard, and for that, it uses the [Coordinate Notation](https://en.wikipedia.org/wiki/Chess_notation#Notation_systems_for_humans).

The white pieces are represented by capital letters and the black pieces are represented in lowercase letters. They are all represented by the first letter of their names, the only exception being the Knight, which is represented by an **N**, leaving the **K** for the king):

**P**awn  
**R**ook  
K**n**ight  
**B**ishop  
**Q**ueen  
**K**ing

## Functions Used

**//In game functions**

**void printboard();**

**void wrongStart();**

**void initialize();**

**void cleartext();**

**void check();**

**//function Prototypes for pieces**

**int pawn(int x,int y,int x1,int y1);**

**int knight(int x,int y,int x1,int y1);**

**int rook(int x,int y,int x1,int y1);**

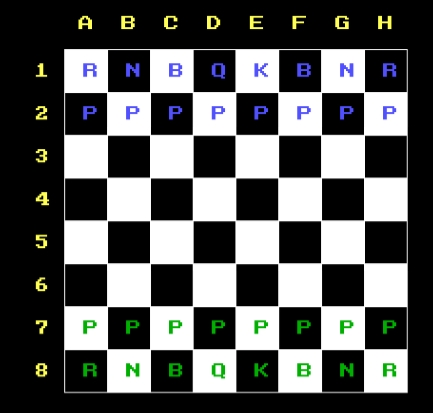
**int bishop(int x,int y,int x1,int y1);**

**int queen(int x,int y,int x1,int y1);**

**int king(int x,int y,int x1,int y1);**

## Drawing the Board

The following picture illustrates the options we have:



These are the functions that draw the board:

**void printboard(){**

**for(int i=0;i<8;i++){**

**for(int j=0;j<8;j++){**

**if((i+j)%2==0)**

**setcolor(WHITE);**

**else**

**setcolor(BLACK);**

**outtextxy(255+(j\*45),105+(i\*45),"ÛÛ");**

**if(board[i][j]<0)**

**setcolor(GREEN);**

**else if(board[i][j]>0)**

**setcolor(9);**

**if(abs(board[i][j])>=9 && abs(board[i][j])<=16){**

**outtextxy(270+(j\*45),105+(i\*45),"P");}**

**else if(abs(board[i][j])==1 || abs(board[i][j])==8){**

**outtextxy(270+(j\*45),105+(i\*45),"R");}**

**else if(abs(board[i][j])==2 || abs(board[i][j])==7){**

**outtextxy(270+(j\*45),105+(i\*45),"N");**

**}**

**else if(abs(board[i][j])==3 || abs(board[i][j])==6){**

**outtextxy(270+(j\*45),105+(i\*45),"B");**

**}**

**else if(abs(board[i][j])==4){**

**outtextxy(270+(j\*45),105+(i\*45),"Q");**

**}**

**else if(abs(board[i][j])==5){**

**outtextxy(270+(j\*45),105+(i\*45),"K");**

**}**

**}**

**}**

**}**

## Wrong Start

## When game is not legally start by user or by giving wrong inputs

**void wrongStart()**

**{**

**cout<<"\nWrong Input\nPress any key to continue...";**

**char ch=getch();**

**if(ch==0){**

**getch();**

**}**

**for(int i=1;i<=30;i++)**

**for(int j=1;j<=4;j++)**

**{**

**gotoxy(i,j);**

**cout<<" ";**

**}**

**gotoxy(1,1);**

**cout<<"\nEnter Initial Position ¯ ";**

**}**

## Initialize

## Starting of the game

**void initialize(){**

**setcolor(WHITE);**

**rectangle(250,90,610,450);**

**// Board Printing**

**for (int i=250;i<560;i+=90)**

**for (int j=90;j<450;j+=90)**

**{**

**bar(i,j,i+45,j+45);**

**bar(i+45,j+45,i+90,j+90);**

**}**

**//Asigning the Values in the empty spaces**

**for(i=2;i<=5;i++)**

**for(j=0;j<=7;j++)**

**board[i][j] = 0;**

**for(j=0;j<=7;j++)**

**{**

**board[0][j] = j+1;**

**board[1][j] = j+9; //Pawns for player 1**

**board[7][j] = 0-(j+1);**

**board[6][j] = 0-(j+9); //Pawns for player 2**

**}**

**printboard();**

**setcolor(YELLOW);**

**outtextxy(265,55,"A");**

**outtextxy(310,55,"B");**

**outtextxy(355,55,"C");**

**outtextxy(400,55,"D");**

**outtextxy(445,55,"E");**

**outtextxy(490,55,"F");**

**outtextxy(535,55,"G");**

**outtextxy(580,55,"H");**

**outtextxy(220,105,"1");**

**outtextxy(220,150,"2");**

**outtextxy(220,195,"3");**

**outtextxy(220,240,"4");**

**outtextxy(220,285,"5");**

**outtextxy(220,330,"6");**

**outtextxy(220,375,"7");**

**outtextxy(220,420,"8");**

**}**

**Cleartext**

Clears the text

**void cleartext(){**

**for(int i=1;i<=27;i++)**

**for(int j=1;j<=8;j++){**

**gotoxy(i,j);**

**cout<<" ";**

**}**

**gotoxy(1,1);**

**cout<<"\nEnter Initial Position ¯ ";**

}

**Check**

The position where your/opponent king is in checked position and has to be move to safe (unchecked) position.

**void check()**

**{**

**int t=0,i,j,x1,y1;**

**if(turn==0){**

**for(i=0;i<=7;i++)**

**{**

**for(j=0;j<=7;j++)**

**{**

**if(board[i][j]==-5)**

**{**

**y1=i; x1=j;**

**break;**

**}**

**}**

**}**

**for(j=0;j<=7;j++){**

**for(i=0;i<=7;i++){**

**if (t==1){**

**setcolor(9);**

**rectangle(20,335,100,365);**

**outtextxy(30,340,"Check");**

**break;**

**}**

**if(board[j][i]>=9 && board[j][i]<=16)**

**t = pawn(i,j,x1,y1);**

**else if(board[j][i]==2 || board[j][i]==7)**

**t = knight(i,j,x1,y1);**

**else if(board[j][i]==4)**

**t = queen(i,j,x1,y1);**

**else if(board[j][i]==1 || board[j][i]==8)**

**t = rook(i,j,x1,y1);**

**else if(board[j][i]==3 || board[j][i]==6)**

**t = bishop(i,j,x1,y1);**

**}// for**

**}// for**

**}//endif turn**

**else {**

**for(i=0;i<=7;i++){**

**for(j=0;j<=7;j++){**

**if(board[i][j]==5){**

**y1=i; x1=j;**

**break;**

**}**

**}**

**}**

**for(j=0;j<=7;j++){**

**for(i=0;i<=7;i++){**

**if (t==1){**

**setcolor(GREEN);**

**outtextxy(30,340,"Check");**

**break;**

**}**

**if(board[j][i]>=(-9) && board[j][i]<=(-16))**

**t = pawn(i,j,x1,y1);**

**else if(board[j][i]==(-2) || board[j][i]==(-7))**

**t = knight(i,j,x1,y1);**

**else if(board[j][i]==(-4))**

**t = queen(i,j,x1,y1);**

**else if(board[j][i]==(-1) || board[j][i]==(-8))**

**t = rook(i,j,x1,y1);**

**else if(board[j][i]==(-3) || board[j][i]==(-6))**

**t = bishop(i,j,x1,y1);**

**}// for**

**}// for**

**}// end else**

**}**

**Gameover**

The position of your/opponent king where it is checkmated and unable to go at safe position or your/opponent king get killed.

**int gameover(){**

**int i,j,a=0,b=0;**

**for(i=0;i<=7;i++)**

**{**

**for(j=0;j<=7;j++)**

**{**

**if(board[i][j]==5)**

**b=1;**

**if(board[i][j]==-5)**

**a=1;**

**}**

**}**

**if(b==0)**

**{**

**setcolor(12);**

**outtextxy(30,340,"Game Over");**

**setcolor(GREEN);**

**outtextxy(1,100,"Player 2 Wins");**

**getch();**

**return 0;**

**}**

**else if(a==0)**

**{**

**setcolor(9);**

**outtextxy(30,340,"Game Over");**

**outtextxy(1,100,"Player 1 Wins");**

**getch();**

**return 0;**

**}**

**return 1;**

**}**

**Functions for all pieces**

**////knight//////**

**int knight(int x,int y,int x1,int y1)**

**{**

**int a=0;**

**if((y1==y+2 && x1==x+1)||(y1==y+2 && x1==x-1)||(y1==y+1 &&**

**x1==x+2)||(y1==y+1 && x1==x-2)||(y1==y-1 && x1==x+2)||(y1==y-1 &&**

**x1==x-2)||(y1==y-2 && x1==x+1)||(y1==y-2 && x1==x-1)){**

**a = 1;**

**}**

**return a;**

**}**

**/////KING////**

**int king(int x,int y,int x1,int y1)**

**{**

**int a = 0;**

**if((y1==y+1 && x1==x)||(y1==y-1 && x1==x)||(y1==y+1 &&**

**x1==x+1)||(y1==y+1 && x1==x-1)||(y1==y-1 && x1==x+1)||(y1==y-1 &&**

**x1==x-1)||(y1==y && x1==x+1)||(y1==y && x1==x-1)){**

**a = 1;**

**}**

**/////CASTLING//////**

**else**

**if(board[y][x+1]==0&&board[y][x+2]==0&&(abs(board[y1][x1])**

**==abs(board[y][x+3])==1||abs(board[y1][x1])==**

**abs(board[y][x+3])==8)){**

**return 2;**

**}**

**if(a==1)**

**{**

**if(board[y][x]==5)**

**count1=1;**

**if(board[y][x]==-5)**

**count2=1;**

**}**

**return a;**

**}**

**//////BISHOP/////**

**int bishop(int x,int y,int x1,int y1)**

**{**

**int a=1,i;**

**if(abs(x1-x)!=abs(y1-y))**

**{**

**a=0;**

**}**

**if((x<x1)&&(y<y1))**

**{**

**for(i=1;(i+x)<x1;i++)**

**{**

**if(board[y+i][x+i]!=0)**

**a=0;**

**}**

**}**

**else if((x>x1)&&(y>y1)){**

**for(i=1;(x-i)>x1;i++){**

**if(board[y-i][x-i]!=0)**

**a=0;**

**}**

**}**

**else if((x>x1)&&(y<y1)){**

**for(i=1;(x-i)>x1;i++){**

**if(board[y+i][x-i]!=0)**

**a=0;**

**}**

**}**

**else if((x<x1)&&(y>y1)){**

**for(i=1;(y-i)>y1;i++){**

**if(board[y-i][x+i]!=0)**

**a=0;**

**}**

**}**

**return a;**

**}**

**////////QUEEN///////**

**int queen(int x,int y,int x1,int y1){**

**if(x==x1||y==y1){**

**//if queen moves in + direction**

**return rook(x,y,x1,y1);**

**}**

**else if(abs(x1-x)==abs(y-y1)){**

**//if queen moves in diagnol direction**

**return bishop(x,y,x1,y1);**

**}**

**else**

**return 0;**

**}**

**///ROOK////**

**int rook(int x,int y,int x1,int y1){**

**int a=1;**

**if(y1==y)**

**{**

**for(int i=x+1;i<x1;i++)**

**{**

**if(board[y1][i]!=0)**

**{**

**a = 0;**

**break;**

**}**

**}**

**for(i=x-1;i>x1;i--)**

**{**

**if(board[y1][i]!=0)**

**{**

**a = 0;**

**break;**

**}**

**}**

**}**

**else if(x1==x)**

**{**

**for(int i=y+1;i<y1;i++)**

**{**

**if(board[i][x1]!=0)**

**{**

**a = 0;**

**break;**

**}**

**}**

**for(i=y-1;i>y1;i--)**

**{**

**if(board[i][x1]!=0)**

**{**

**a = 0;**

**break;**

**}**

**}**

**}**

**else**

**{**

**a=0;**

**}**

**return a;**

**}**

**///PAWN/////**

**int pawn(int x,int y,int x1,int y1){**

**int a=0;**

**if(turn==0){**

**if(y==1){**

**if(y1==(y+2) && x1==x){**

**if(board[y1][x1]==0&&board[y+1][x]==0){**

**a = 1;**

**}**

**}**

**}**

**if(y1==y+1 && x1==x){**

**if(board[y1][x1]==0){**

**a = 1;**

**}**

**}**

**else if(y1==(y+1) && (x1==(x+1)||x1==(x-1))){**

**if(board[y1][x1]<0){**

**a = 1;**

**}**

**}**

**}**

**else if(turn==1){**

**if(y==6){**

**if( y1==(y-2) && x1==x){**

**if(board[y1][x1]==0&&board[y-1][x]==0){**

**a = 1;**

**}**

**}**

**}**

**if( y1==(y-1) && x1==x){**

**if(board[y1][x1]==0){**

**a = 1;**

**}**

**}**

**else if(y1==(y-1) && (x1==(x-1)||x1==(x+1))){**

**if(board[y1][x1]>0){**

**a = 1;**

**}**

**}**

**}**

**if(a==1)**

**{**

**if(turn==0)**

**{**

**if(y1==7)**

**return 2;**

**}**

**else**

**{**

**if(y1==0)**

**return 2;**

**}**

**}**

**return a;**

**}**

**Main function/ printing welcome texts/ save & load of game/ Legal moves for all pieces**

**void main()**

**{**

**//Graphics Initialization**

**int gdriver = DETECT;**

**int gmode;**

**initgraph(&gdriver,&gmode,"c:\\turboc3\\bgi");**

**//Loading Intro**

**for(float i=0; i<=4;i=i+0.1)**

**{**

**setcolor(12);**

**settextstyle(0,HORIZ\_DIR,i);**

**outtextxy(50,200,"WELCOME TO THE CHESS");**

**delay(30);**

**cleardevice();**

**}**

**outtextxy(50,200,"WELCOME TO THE CHESS");**

**settextstyle(0,HORIZ\_DIR,2);**

**getch();**

**cleardevice();**

**//Displaying Group Members**

**setcolor(CYAN);**

**outtextxy(50,220,"Designed by:");**

**setcolor(6);**

**outtextxy(50,250,"Sahil Sao");**

**setcolor(WHITE);**

**outtextxy(180,350,"MCA 2nd Semester");**

**//Displaying Roll Numbers**

**setcolor(6);**

**outtextxy(350,250,"Roll NO. 22");**

**getch();**

**cleardevice();**

**cleardevice();**

**settextstyle(0,HORIZ\_DIR,2);**

**setcolor(12);**

**outtextxy(50,10,"During the game Press:");**

**setcolor(6);**

**settextstyle(0,HORIZ\_DIR,2);**

**outtextxy(50,60,"F2 to save file");**

**outtextxy(50,80,"F3 to load file");**

**outtextxy(50,100,"ESC to exit game");**

**outtextxy(50,120,"U to undo selection");**

**outtextxy(50,200,"Press any key to play ");**

**getch();**

**cleardevice();**

**cleartext();**

**initialize();**

**char ch;**

**int x=-1,y=-1,x1=-1,y1=-1;**

**setcolor(9);**

**outtextxy(10,400,"Player 1");**

**for(;;)**

**{**

**int game;**

**game=gameover();**

**if(game==0)**

**{**

**break;**

**}**

**FILE \*fp;**

**ch=getch();**

**if(ch==0){**

**ch=getch();**

**if(ch=='<'){**

**//////SAVE/////**

**for(;;)**

**{**

**cout<<"\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\**

**\b\b\bEnter File name to save \n";**

**cin>>strFileName; //Enter File Name**

**fp=fopen(strFileName,"r+");**

**if(fp!=NULL)**

**{**

**cout<<"File already exist\n";**

**}**

**else**

**{**

**fp=fopen(strFileName,"w+");**

**if(fp==NULL)**

**{**

**cout<<"Invalid File Name\n";**

**}**

**else**

**break;**

**}**

**}//end for**

**fprintf(fp,"%d\n",turn);**

**for(int i=0;i<8;i++)**

**{**

**for(int j=0;j<8;j++)**

**{**

**fprintf(fp,"%d\n",board[i][j]);**

**}**

**}**

**fclose(fp);**

**cleartext();**

**x=-1;**

**y=-1;**

**x1=-1;**

**y1=-1;**

**continue;**

**}**

**if(ch=='=')**

**{**

**////Load/////**

**for(;;)**

**{**

**cout<<"\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\**

**\b\b\b\b\bEnter File name to Load \n";**

**cin>>strFileName;**

**fp=fopen(strFileName,"r+");**

**if(fp==NULL)**

**{**

**cout<<"File does not exist\n";**

**}**

**else**

**{**

**break;**

**}**

**}//end for**

**fscanf(fp,"%d",&turn);**

**for(int i=0;i<8;i++){**

**for(int j=0;j<8;j++){**

**fscanf(fp,"%d",&board[i][j]);**

**}**

**}**

**fclose(fp);**

**printboard();**

**cleartext();**

**x=-1;**

**y=-1;**

**x1=-1;**

**y1=-1;**

**setcolor(0);**

**outtextxy(10,400,"Player Û");**

**if(turn!=0){**

**check();**

**setcolor(2);**

**outtextxy(10,400,"Player 2");**

**}**

**else{**

**check();**

**setcolor(9);**

**outtextxy(10,400,"Player 1");**

**}**

**continue;**

**}**

**}**

**if(ch==27){**

**break;**

**}**

**else if(ch=='u'||ch=='U'){**

**x=-1;**

**x1=-1;**

**y1=-1;**

**y=-1;**

**cleartext();**

**continue;**

**}**

**if(x==-1){**

**ch=toupper(ch);**

**if(ch>=65 && ch<=72){**

**cout<<ch;**

**x=ch-65;**

**}**

**}**

**else if(y==-1)**

**{**

**if(ch>='1' && ch<='8')**

**{**

**cout<<ch;**

**y=ch-'1';**

**if(board[y][x]==0){**

**wrongStart();**

**x=-1;**

**y=-1;**

**}**

**else**

**{**

**if(turn==0)**

**{**

**if(board[y][x]<0)**

**//if Player 1 has selected a piece of Player 2**

**{**

**wrongStart();**

**x=-1;**

**y=-1;**

**}**

**}**

**else if(turn!=0)**

**{**

**if(board[y][x]>0)**

**{**

**wrongStart();**

**x=-1;**

**y=-1;**

**}**

**}**

**}**

**}**

**if(x!=-1 && y!=-1)**

**cout<<"\n\nEnter Final Position ¯ ";**

**}**

**else if(x1==-1)**

**{ //Converting lower to upper case**

**ch=toupper(ch);**

**if(ch>=65 && ch<=72)**

**{**

**cout<<ch;**

**x1=ch-65;**

**}**

**}**

**else if(y1==-1)**

**{**

**if(ch>='1' && ch<='8')**

**{**

**cout<<ch;**

**y1=ch-'1';**

**}**

**int b=0,killed=0;**

**/////pawn/////**

**if(abs(board[y][x])>=9 &&**

**abs(board[y][x])<=16){ //if piece selected has value greater than 8 and less than 17**

**b = pawn(x,y,x1,y1);**

**}//end pawn**

**//////knight//////**

**else if(abs(board[y][x])==2 ||**

**abs(board[y][x])==7){ //if piece selscted has value 2 and 7**

**b = knight(x,y,x1,y1);**

**}//end knight**

**////queen///**

**else if(abs(board[y][x])==4){ ////if piece selected has value 4**

**b = queen(x,y,x1,y1);**

**}//end queen**

**///king///**

**else if(abs(board[y][x])==5){ //if piece selected has value 5**

**b = king(x,y,x1,y1);**

**}//end king**

**///rook///**

**else if(abs(board[y][x])==1 ||**

**abs(board[y][x])==8){ //if piece selscted has value 1 and 8**

**b = rook(x,y,x1,y1);**

**}//end rook**

**///bishop///**

**else if(abs(board[y][x])==3 ||**

**abs(board[y][x])==6){ //if piece selscted has value 3 and 6**

**b = bishop(x,y,x1,y1);**

**}//end bishop**

**if(b==2){**

**if(abs(board[y][x])>=9 && abs(board[y][x])<=16){**

**char pp;**

**cout<<"\n\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\**

**\bwhich piece Q,R,N,B";**

**pp=getch();**

**if(turn==0){**

**if(pp=='r')**

**board[y1][x1]=1;**

**else if(pp=='n')**

**board[y1][x1]=2;**

**else if(pp=='b')**

**board[y1][x1]=3;**

**else if(pp=='q')**

**board[y1][x1]=4;**

**}**

**else{**

**if(pp=='r')**

**board[y1][x1]=-1;**

**else if(pp=='n')**

**board[y1][x1]=-2;**

**else if(pp=='r')**

**board[y1][x1]=-3;**

**else if(pp=='q')**

**board[y1][x1]=-4;**

**}**

**board[y][x]=0;**

**killed =1;**

**}**

**else if(abs(board[y][x])==5){**

**if(board[y][x]==5){**

**if(count1==1);**

**wrongStart();}**

**else if(board[y][x]==-5){**

**if(count1==1);**

**wrongStart();}**

**else{**

**board[y][x+1]=board[y1][x1];**

**board[y][x+2]=board[y][x];**

**board[y1][x1]=0;**

**board[y][x]=0;**

**killed=1;**

**}**

**}**

**printboard();**

**}**

**else if(b==1)**

**{**

**if(turn==0)**

**{**

**if(board[y1][x1]<0)**

**{**

**board[y1][x1]=board[y][x];**

**board[y][x] = 0;**

**killed=1;**

**}**

**}**

**if(turn!=0)**

**{**

**if(board[y1][x1]>0)**

**{**

**board[y1][x1]=board[y][x];**

**board[y][x] = 0;**

**killed=1;**

**}**

**}**

**if(board[y1][x1]==0)**

**{**

**int temp=board[y][x];**

**board[y][x]=board[y1][x1];**

**board[y1][x1]=temp;**

**killed=1;**

**}**

**printboard();**

**}**

**x=-1;**

**y=-1;**

**x1=-1;**

**y1=-1;**

**setcolor(0);**

**outtextxy(30,340,"Check");**

**if(killed==0){**

**wrongStart();**

**continue;**

**}**

**cleartext();**

**setcolor(0);**

**outtextxy(10,400,"Player Û");**

**if(turn==0){**

**check();**

**turn=1;**

**setcolor(2);**

**outtextxy(10,400,"Player 2");**

**}**

**else{**

**check();**

**turn=0;**

**setcolor(9);**

**outtextxy(10,400,"Player 1");**

**}**

**}//end legal move**

**}//end for**

## Playing the Game

### **Starting a New Game**

Start the app and press **any key**, followed by **ENTER**, to start a new game. The board is shown and it's WHITE turn.

### **Make a Move**

You will be prompted to choose a piece to be moved. Do it by entering two characters (uppercase or lowercase will give the same results) describing first the column, then the row where the piece you want to be moved currently is. For example, the white pawn in front of the king is the **E2** square.

Next, you'll be prompted for the destination square. One of the most common moves is moving the pawn from **E2** to **E4**.

You will be warned if the move is invalid.

## Saving / Loading a Game

Save a game is useful if you want to finish it later. By pressing **F2** to save the game or **F3** to load the game from saved position.

**Bugs**

This application is certainly not bug-free. The castling feature is not happening sometimes.

**Bibliography**

* 1. <https://en.wikipedia.org/wiki/Chess>
  2. <http://www.vyomworld.com/source/code.asp?id=03&l=C_Projects&t=Chess%20Game%20Developed%20in%20C>++.
  3. [https://www.codeproject.com/Articles/1214018/Chess-Console-Game-in-Cplusplus (article](https://www.codeproject.com/Articles/1214018/Chess-Console-Game-in-Cplusplus%20(article) by Jerome Vonk)
  4. <https://stackoverflow.com/questions/47566243/simple-console-chess-game>
  5. <https://www.go4expert.com/articles/chess-using-bgi-graphics-c-cpp-t598/>