**MINI-PROJECT**

**CHESS GAME**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **SI.No** | **Content** | **Page Number** |
| **1.** | **Problem Statement** | **2** |
| **2.** | **Description** | **2** |
| **3.** | **Requirement Gathering** | **4** |
| **4.** | **Research about Problem Statement** | **4** |
| **5.** | **Design** | **7** |
| **6.** | **Test Plan** | **8** |
| **7.** | **Test Cases** | **8** |
| **8.** | **Expected Outcome** | **10** |
| **9.** | **Conclusion** | **10** |

**Problem Statement:**

I am trying to create a game of chess using C programming. The game should have good graphics and should allow two users to play the game using the keyboard based on the rules of chess.

**Description:**

Chess is based on some fairly simple rules. Each side has 16 pieces of 6 different types: Pawn, Rook, Knight, Bishop, Queen and King. The game is played on a board with 64 squares and the object of the game is to capture the other team's King. Each of the 6 types of piece have different rules for moving around the board, and can take pieces from the other team under certain conditions. Your job is to ensure that the two players playing your game of chess get to take turns (white always goes first), and cannot make a move that is invalid.

In order to ensure that only valid moves are made, you should allow the player whose turn it is to click on one of their own pieces on the board. This event should trigger your program to highlight all the valid moves for the piece that was selected. If the user then clicks on one of the highlighted squares, the piece will move. If the user clicks anywhere other than a highlighted square, the selection of the piece will be cancelled and no move will be made.

Only one chess piece can occupy a single square. Valid moves for each of the pieces are shown below:

***Pawn***: one their first move, pawns may move either one or two squares directly ahead of them (unless blocked by any other piece). After their first move, they may only move one square directly ahead of them per turn. They may only take pieces of the other color when the opposing piece is in a square diagonally in front of them.(Pawn moves from Chess is Fun.) The player will need to implement some functionality to allow a pawn to change to another piece if it reaches the other side of the board. You are not required to implement the "En Passant" rule for Pawns (although you are welcome to if you so desire).

***Rook***: Rooks may move horizontally or vertically as many squares as possible. The range of their movement is only terminated by the edge of the board or another piece. Rooks may take the first opposing piece that blocks their movement in any vertical or horizontal direction. (Rook moves from Chess is Fun.)

***Knight***: Knights always move in the shape of the letter "L". This means that they move two squares forward and one to the left or right in any direction. The Knight is the only piece that can jump over other pieces. (Knight moves from Chess is Fun.)

***Bishop***: The Bishop's movement is similar to the Rook's except that the Bishop can move diagonally in any direction. (Bishop moves from Chess is Fun.)

***Queen***: The Queen's movement is a combination of the Rook's and the Bishop's movement. This means that the Queen can move both horizontally, vertically and diagonally in any direction. (Queen moves from Chess is Fun.)

***King***: The King can only move a single square in any direction. The King cannot move into a 'check' position (a square in which he could be taken by an opposing piece). (King moves from Chess is Fun.) You are not required to implement "castling" for the King and Rook (but again, you may if you wish).

At the beginning of the game, each piece has a set position on the board. Each time a piece moves, you need to add the move to the history of the game.

***Check***: My project is required to display a message any time one of the players is in check. Also, the project should not allow any move that will leave the King in a check position.

***Checkmate and Stalemate***: Checkmate occurs when the moving player's King is in check, and there are no valid moves that can be used to remove the king from check. Stalemate occurs when the moving player's King is not in check, but there are no valid moves that would not put the king in check. In either case (checkmate or stalemate), the game is over. In the case of checkmate, the player whose king is in checkmate loses the game. In the case of stalemate, neither player wins and the game is declared a draw.

My program is required to detect when checkmate or stalemate has occurred, in which case your program should print out an appropriate message and not allow further moves to be taken.

However, even when checkmate or stalemate is reached, the undo function should still work.

**Requirement Gathering:**

* In order to create a game of chess, one must first understand the game of chess, how it is played, what are all the characters in the game.
* The next step is to create the chess board using graphics.
* The next step is to create the pieces and assign the location to them.
* The next step is to create two players and allow them to play the game of chess using keyboards.
* CodeBlocks IDE and C Programming Language.
* Header Files used :
* graphics.h - Used to create graphicalical operations.
* stdio.h - Used to perform input and output operations in C.
* stdlib.h - Perform standard utility.
* conio.h - Contains function for console input/output.
* ctype.h - Useful for testing and mapping characters.
* dos.h - Used for handling interrupts, producing sound, date and time functions.

**Research about problem Statement:**

Chess is a two-player strategy board game played on a checkerboard with 64 squares arranged in an 8×8 square grid. Played by millions of people worldwide, chess is believed to be derived from the Indian game chaturanga sometime before the 7th century. Chaturanga is also the likely ancestor of the East Asian strategy games xiangqi (Chinese chess), janggi (Korean chess), and shogi (Japanese chess). Chess reached Europe via Persia and Arabia by the 9th century, due to the Umayyad conquest of Hispania. The queen and bishop assumed their current powers in Spain in the late 15th century, and the modern rules were standardized in the 19th century.

Play involves no hidden information. Each player begins with 16 pieces: one king, one queen, two rooks, two knights, two bishops, and eight pawns. Each piece type moves differently, with the most powerful being the queen and the least powerful the pawn. The objective is to checkmate the opponent's king by placing it under an inescapable threat of capture. To this end, a player's pieces are used to attack and capture the opponent's pieces, while supporting one another. During the game, play typically involves exchanging pieces for the opponent's similar pieces, and finding and engineering opportunities to trade advantageously or to get a better position. In addition to checkmate, a player wins the game if the opponent resigns, or in a timed game, runs out of time. There are also several ways a game can end in a draw.

The rules of chess are published by FIDE (Fédération Internationale des Échecs), chess's international governing body, in its Handbook. Rules published by national governing bodies, or by unaffiliated chess organizations, commercial publishers, etc., may differ. FIDE's rules were most recently revised in 2018.

By convention, chess game pieces are divided into white and black sets. Each set consists of 16 pieces: one king, one queen, two rooks, two bishops, two knights, and eight pawns. The pieces are set out as shown in the diagram and photo. The players of the sets are referred to as White and Black, respectively.

The game is played on a square board of eight rows (called ranks, denoted 1 to 8 from bottom to top according to White's perspective) and eight columns (called files, denoted a to h from left to right according to White's perspective). The 64 squares alternate in color and are referred to as light and dark squares. The chessboard is placed with a light square at the right-hand corner nearest to each player. Thus, each queen starts on a square of its own color (the white queen on a light square; the black queen on a dark square).

***Movement:*** In competitive games, the colors are allocated by the organizers; in informal games, the colors are usually decided randomly, for example by a coin toss, or by one player concealing a white pawn in one hand and a black pawn in the other, and having the opponent choose. White moves first, after which players alternate turns, moving one piece per turn (except for castling, when two pieces are moved).

A piece is moved to either an unoccupied square or one occupied by an opponent's piece, which is captured and removed from play. With the sole exception of en passant, all pieces capture by moving to the square that the opponent's piece occupies.

Moving is compulsory; it is illegal to skip a turn, even when having to move is detrimental. A player may not make any move that would put or leave the player's own king in check. If the player to move has no legal move, the game is over; the result is either checkmate (a loss for the player with no legal move) if the king is in check, or stalemate (a draw) if the king is not.

Each piece has its own way of moving. In the diagrams, the dots mark the squares to which the piece can move if there are no intervening piece(s) of either color (except the knight, which leaps over any intervening pieces).

* The king moves one square in any direction. The king also has a special move called castling that involves also moving a rook.
* A rook can move any number of squares along a rank or file, but cannot leap over other pieces. Along with the king, a rook is involved during the king's castling move.
* A bishop can move any number of squares diagonally, but cannot leap over other pieces.
* A queen combines the power of a rook and bishop and can move any number of squares along a rank, file, or diagonal, but cannot leap over other pieces.
* A knight moves to any of the closest squares that are not on the same rank, file, or diagonal. (Thus the move forms an "L"-shape: two squares vertically and one square horizontally, or two squares horizontally and one square vertically.) The knight is the only piece that can leap over other pieces.
* A pawn can move forward to the unoccupied square immediately in front of it on the same file, or on its first move it can advance two squares along the same file, provided both squares are unoccupied (black dots in the diagram); or the pawn can capture an opponent's piece on a square diagonally in front of it on an adjacent file, by moving to that square (black "x"s). A pawn has two special moves: the en passant capture and promotion.

**Games can be won in the following ways**:

***Checkmate***: The player whose turn it is to move is in check and has no legal move to escape check.

***Resignation***: Either player may resign, conceding the game to the opponent.

Win on time: In games with a time control, a player wins if the opponent runs out of time, even if the opponent has a superior position, as long as the player has a theoretical possibility to checkmate the opponent were the game to continue.

***Forfeit***: A player who cheats, violates the rules, or violates the rules of conduct specified for the particular tournament, can be forfeited.

**Design:**

This gives a basic idea about how the program executes. When the user executes the program, first the chess board will be shown with all the components.

The user(white) is supposed to make the first move and the next move goes to black and so on. There are certain movements that are applicable to the components as explained above and can only be moved in that direction.

The different functions used in these program are:

* gameover() - The function tries to find whether the game is over or not.
* printboard() - The function displays the chess board with all the components.
* wrongstart() - The function tries to prevent wrong movements.
* Initialize() - The function initiates the program.
* check() - The function ensures that the game is over or if one player checks the other.
* Pawn() - The function used for the movement of pawn.
* Rook() - The function used for the movement of rook.
* Horse() - The function used for the movement of a horse.
* Bishop() - The function used for the movement of bishops.
* Queen() - The function used for the movement of the queen.
* King() - The function used for the movement of the king.

**Test Plan:**

All the features present in the program should be tested. There are three types of testing that should be conducted:

* Unit Testing - It focuses on the smallest unit of software design. In this we test an individual unit or group of interrelated units.It is often done by programmers by using sample input and observing its corresponding outputs.
* Integration Testing - The objective is to take unit tested components and build a program structure that has been dictated by design.Integration testing is testing in which a group of components are combined to produce output.
* System Testing - In this software is tested such that it works fine for different operating system.It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working.

Suspension Criterion - If there is 40% of the test cases are failed then the testing is suspended until the programmer fixes the mistakes

Exit Criterion - Specifies a criteria that denotes a successful completion of the test phase

The user should make sure to select the right option for the movement, it is quite easy for the user to move their components using the keyboard.

The test plan for this program is to check whether the right movement of the components are done so as to play a clean game.

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SI.No** | **Test Description** | **Test Steps** | **Test Data** | **Test Output** |
| **1.** | Whether the user wants to explore. | User decides to execute the program. | Execute the program and clickenter. | Go tothegame. |
| **2.** | Whether white pieces are moved first. | Player’shaving thewhite pieceshould make the first movement. | Usingkeys in the keyboard moves the componentneeded by whiteplayer. | White player starts the game without any error. |
| **3.** | Whether the correct sequence of players are playing. | First white components are moved, then black components are moved and so on. | The players should play simultaneously, at first white should play then black and so on. The movement of components are made using arrow keys. | The game is played properly. |
| **4.** | Whetherthe pieces are moved properly | All the pieces have a set of movements applicable to them and can only be moved in that direction. | Therules are explained above and all movements should be made according. The movement is made using arrow keys. | The pieces are moved properly. |
| **5.** | Whether the game reaches a point of checkmate. | If one player checks the other, the game is necessarily over. | If check happens, see if there are any other possible moves left, if not declare the player as winner. | Player is the winner. |
| **6.** | Whether any variations occur during play that are out of ordinary or the game abruptly stops. | The game should be played according to the rules and no rules should be broken. | The players are supposed to check each other's movements, the game will also check them. | Proper playing of games. |

**Expected Outcome:**

The outcome of this program is such that the user finds a good game. It should be very easy for the user, since it is clearly provided in the program. The game should have a proper graphical chess board with all the components and should abide by the rules of chess and should declare the player who is eligible as winner and should provide the users with a good time.

**Conclusion:**

The game of chess is made for all chess enthusiasts, The game is made using c programming and has the necessary graphics included so as to provide an aspiring and entertaining game.

The game could be played by two players using keyboards. The easy and friendly design makes it a good way to enjoy your time with others.