Ain Shams University
Faculty Of Computer And Information

Chess Game Project

DR :

Abdelaziz A. Abdelhamid .

TA :

Ibrahim Amer .

Team Members :

1. Muhammad Fouad Ahmed Badr .
   Section : 15
2. Muhammad Ali Muhammad Sayed .
   Section : 15
3. Amr Ayman Ayoub Abd El Shafy .
   Section : 12
4. Muhammad Ahmed Ramzy Khalil .
   Section : 14
5. Yasmeen Amr Abd El Moniem Ghareeb .
   Section : 20

Team Mentor :

Muhammad Ahmed Ramzy .

Chess-Game Documentation

Chess is a two-player strategy board game played on a chessboard, a checkered GameBoard with 64 squares arranged in an eight-by-eight grid . Chess is played by millions of people worldwide, both amateurs and professionals .

Each player begins the game with 16 pieces : one King , one Queen , two Rooks, two knights , two Bishops , and eight Pawns . Each of the six piece types moves differently . The most powerful piece is the Queen and the least powerful piece is 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 their own . In addition to checkmate, the game can be won by voluntary resignation by the opponent, which typically occurs when too much material is lost, or if checkmate appears unavoidable . A game may also result in a draw in several ways .

Our game is a two-player game Each player has 16 pieces : one King, one Queen, two Rooks, two knights, two Bishops, and eight Pawns . Each of the six pieces types moves differently according to the rule of the game .
Our game has 64 buttons which implement the chessboard .
Our game has different classes that implement the game
the first class is Chess-Game class that contain :
1- Static array of strings which save the type of the piece on it's button number
as there are 6 types pieces of on chessboard (Pawn , Bishop , knight , Rook , Queen and King) .
2- Static array of integers which save the type of the player (1 or 2) .

The Pawn (♙ ♟) is the most numerous piece in the game of chess, and in most circumstances, also the weakest . It historically represents

infantry, or more particularly , armed peasants or pike-men . Each player begins a game of chess with eight Pawns, one on each square of the rank immediately in front of the other pieces .
A Pawn that advances all the way to the opposite side of the board (the opposing player's first rank) is promoted to another piece of that player's choice : a Queen, Rook, Bishop, or knight of the same color .

Our game implement this piece as class called Pawn which contain all possible Features in Pawn .
this class contain some methods like :

1- All possible moves which fill array with all possible moves for any Pawn
this function in all pieces (knight , Bishop , . . .)
2- Convert which can convert any Pawn to any piece (Queen , Rook , knight , Bishop) if it reach to the end of the chessboard by open new window to the player to choose his new piece .
3- move_OR_Kills that only kills by Pawn if it can kill or move if it can move .

The knight (♘ ♞) is a piece in the game of chess, representing a knight (armored cavalry) . It is normally represented by a horse's head and neck . Each player starts with two knights, which begin on the row closest to the player, one square from each corner .
The knight move is unusual among chess pieces . When it moves, it can move to a square that is two squares horizontally and one square vertically, or two squares vertically and one square horizontally . The complete move therefore looks like the letter L . Unlike all other standard chess pieces, the knight can 'jump over' all other pieces (of either color) to its destination square . It captures an enemy piece by replacing it on its square .

All these are implemented in our game as we implement Knight class which contain all possible Features in knight .

This class contain some methods like :

1- All possible moves which fill array with all possible moves for any knight
this function in all pieces (knight , Bishop , . .) .
2-move_OR_Killss that only kill by knight if it can kill or move if it can move .

A Bishop (♗,♝) is a piece in the board game of chess . Each player begins the game with two Bishops . One starts between the King's knight and the King, the other between the Queen's knight and the Queen .
The Bishop has no restrictions in distance for each move, but is limited to diagonal movement . Bishops, like all other pieces except the knight, cannot jump over other pieces . A Bishop captures by occupying the square on which an enemy piece sits .

Our game implements this piece as class called Bishop which contain all possible Features in Bishop .
this class contain some methods like :

1- all possible moves which fill array with all possible moves for any Bishop
this function in all pieces (knight , Bishop , . .) .
2-move_OR_Killss that only kill by Bishop if it can kill or move if it can move .

A Rook (♖ ♜ borrowed from Persian رخ rokh , Sanskrit रथ ratha , "chariot") is a piece in the strategy board game of chess . Formerly the piece was called the tower, marquess , rector, and comes (Sunnucks 1970) . The term castle is considered informal, incorrect, or old-fashioned .

Each player starts the game with two Rooks, one in each of the corner squares on their own side of the board .

In the opening, the Rooks are blocked in by other pieces and cannot immediately participate in the game : so it is usually desirable to connect one's Rooks on the first rank by clearing all pieces except the King and Rooks from the first rank and then castling . In that position, the Rooks support each other, and can more easily move to occupy and control the most favorable files .

Our game implements this piece as class called Rook which contain all possible Features in Rook .
this class contain some methods like :

1- all possible moves which fill array with all possible moves for any Rook
this function in all pieces (knight , Bishop , ...) .
2-move_OR_Killss that only kill by Pawn if it can kill or move if it can move .

The Queen (♕,♛) is the most powerful piece in the game of chess, able to move any number of squares vertically, horizontally or diagonally . Each player starts the game with one Queen, placed in the middle of the first rank next to the King . Because of the value of a Queen, it is sometimes used as bait to lure an opponent into a trap by a Queen sacrifice . Another tactic is to use the Queen to threaten the opponent's Queen, to either retreat or to exchange the Queen (losing both of them) to reduce the game to less powerful pieces . The Queen is often used in conjunction with another piece, such as teamed with a Bishop or Rook, where the pieces could guard each other while threatening the opponent pieces .
With the chessboard oriented correctly, the white Queen starts on a white square and the black Queen starts on a black square .

We implement this piece as class Queen which only contians two objects one from class , Rook and the other form object Bishop as we can define one Queen by union of one Rook and one Bishop .
this class contains some methods like :

1- all possible moves which fill array with all possible moves for any Queen
only by call function all possible moves by Rook object and call function all possible moves by Bishop object .
2-move_OR_Killss that only kill by Queen if it can kill or move if it can move .

In chess , the King (♔, ♚) is the most important piece . The object of the game is to trap the opponent's King so that its escape is not possible (checkmate) . If a player's King is threatened with capture , it is said to be in check, and the player must remove the threat of capture on the next move . If this cannot be done , the King is said to be in checkmate . Although the King is the most important piece, it is usually the weakest piece in the game until a later phase , the endgame .
A King can move one square in any direction (horizontally, vertically, or diagonally) unless the square is already occupied by a friendly piece or the move would place the King in check . As a result, the opposing Kings may never occupy adjacent squares (see opposition), but the King can give discovered check by Unmasking a Bishop, Rook, or Queen . The King is also involved in the special move of castling .

We do our best to implement the King with all its Features according to the rules of the game .
We implement King class which contain attributes :
1- two integer static variables (King1,King2) one for each player that takes only two values 0 or 1
0 -> if the number of moves of the King = 0
so the King can any time make special move of castling .
1 -> if the number of moves of the King != 0
so the King can't make special move of castling .
2- objects from all other pieces to know all possible move for all enemies .
3- array of integers called enemy to save enemies positions .

and it has some methods like :

1- all possible moves which fill array with all possible moves for the King

2- move_OR_Kills that only kills by King if it can kill or move if it can move .

3- Special move which return true if the King can move the special move of castling and false otherwise .

4- test to test any piece before it's move to ensure the King can't attacked .

NOTES about Game :

1- You can click any piece to view it's all possible move (place lights with different color) .

2- You can make King special move according to the rule (castling) .

3- There is a classic music during the game .

4- Draw and win one of the two player implement as a two separated functions .

Chess-Game UML