chessMovement Module Explanation

chessMovement库说明

Chinese:

1. createBoard(): 创建一个棋盘，并返回一个表示棋盘的二维数组（如：[['BRook', 'BKnight', 'BBishop', 'BQueen', 'BKing', 'BBishop', 'BKnight', 'BRook'], ['BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn'], ['', '', '', '', '', '', '', ''], ['', '', '', '', '', '', '', ''], ['', '', '', '', '', '', '', ''], ['', '', '', '', '', '', '', ''], ['WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn'], ['WRook', 'WKnight', 'WBishop', 'WKing', 'WQueen', 'WBishop', 'WKnight', 'WRook']]， 其中“B”指black， “W”指white）

2. checkPlace(Board, Piece): Board变量是一个表示棋盘的二维数组（由函数createBoard()创建），Piece变量是一个用于表示棋子种类的字符串（可选值："Pawn", "Rook", "Bishop", "Knight", "Queen", "King"）将返回一个包含棋子位置坐标的二维数组，前一半表示黑棋，后一半表示白棋，如：[[10, 11, 12, 13, 14, 15, 16, 17], [60, 61, 62, 63, 64, 65, 66, 67]]

1. pawnMovement(Board, Colour, Place): Board变量如上。Colour变量是int型整数，可取值为0或1，0表示black，1表示white，Place表示棋子坐标，如3（第一行第四个）， 72（第八行第三个）（需要注意，Place最大值为77， 最小值为0，且其个位的值不能超过7）

这个函数会返回一个一维数组，表示给定的兵在棋盘上所有可以到达的位置坐标（如[21, 31]）

1. rookMovement(Board, Colour, Place)：类似于pawnMovement，但判断的是车的行动
2. bishopMovement(Board, Colour, Place): 类似于pawnMovement，但是判断的是象的行动
3. queenMovement(Board, Colour, Place): 类似于pawnMovement，但是判断的是女王的行动，且会给出一个二维数组，上半表示横向及纵向的移动，下半表示斜线移动（如：[[35, 36, 37, 33, 32, 31, 30, 44, 54, 64, 24], [45, 56, 67, 23, 43, 52, 61, 25]]）
4. knightMovement(Board, Colour, Place): 类似于pawnMovement，但判断的是马的行动
5. kingMovement(Board, Colour, Place): 类似于pawnMovement，但是判断的是王的行动
6. pieceMovement(Board, P1, P2): P1，P2为象棋中标准坐标表示方法（如a2，b5等），Board为函数createBoard()创建的二维数组。 这个函数是前面六个函数的整合，如果棋子P1可以到达P2，则返回True，反之返回False
7. kingCanNotGo(Board, Colour)：Board为函数createBoard()创建的二维数组，Colour指需要判断的颜色（为int型整数，值为为0或1，0表示black，1表示white）。这个函数将返回一个一维数组，用于表示Colour所表示的颜色中王无法前往（即前往将会被将军）的区域
8. checkKing(Board, Colour): Board为函数createBoard()创建的二维数组，Colour指需要判断的颜色。这个函数返回一个布尔值，用于判断Colour所表示的颜色的王是否被将军
9. castling(Board, P1, P2): 判断了在棋盘Board（由函数createBoard()创建）上棋子P1和P2能否进行王车易位，P1 和P2是两子的坐标，个位及十位均在0到7之间，将返回一个布尔值。
10. pieceMove(Board, P1, P2): P1，P2是标准的国际象棋坐标（如a1，b3等），Board是由函数createBoard()创建的棋盘，这个函数将首先判定是否被将军（如果被将军返回"Move your king first"），再判定该棋子（P1）是否可以移动到P2位置（如果不行返回"Can't move"），如果可以移动将返回一个经过更改后的二维数组Board用于表示棋盘上各棋子的位置
11. hasValidMoves(Board, Colour): Board及Colour定义如上，将返回一个布尔值，如果颜色为Colour的棋子还能行动（即任意子行动后国王不会被将军），将返回True，反之返回False
12. pawnChange(Board, Colour, Place, Kind): Board，Colour及Place定义如上，Kind为棋子的种类（可能值："Pawn", "Rook", "Bishop", "Knight", "Queen", "King"）。将将一个到达对方底线的兵变换为Kind类型棋子，如果Place上的棋子不是兵，或未到达对方底线，则不进行任何改动。将返回经过上述操作的棋盘Board
13. gameSituation(Board): Board定义如上。给出现在棋局情况，可能返回值有”White wins”, ”Black wins”, ”Draw” , ”Continue”

English:

1. createBoard(): Create a chessboard and return a two-dimensional array representing the board, e.g.: [['BRook', 'BKnight', 'BBishop', 'BQueen', 'BKing', 'BBishop', 'BKnight', 'BRook'], ['BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn', 'BPawn'], ['', '', '', '', '', '', '', ''], ['', '', '', '', '', '', '', ''], ['', '', '', '', '', '', '', ''], ['', '', '', '', '', '', '', ''], ['WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn', 'WPawn'], ['WRook', 'WKnight', 'WBishop', 'WKing', 'WQueen', 'WBishop', 'WKnight', 'WRook']], where "B" represents black and "W" represents white.
2. checkPlace(Board, Piece): The Board variable is a two-dimensional array representing the chessboard (created by the createBoard() function), and the Piece variable is a string representing the type of chess piece (possible values: "Pawn", "Rook", "Bishop", "Knight", "Queen", "King"). It will return a two-dimensional array containing the coordinates of the chess pieces, with the first half representing the black pieces and the second half representing the white pieces, e.g.: [[10, 11, 12, 13, 14, 15, 16, 17], [60, 61, 62, 63, 64, 65, 66, 67]].
3. pawnMovement(Board, Colour, Place): The Board variable is as described above. The Colour variable is an integer that can take values 0 or 1, where 0 represents black and 1 represents white. The Place variable represents the coordinates of the chess piece, such as 3 (fourth square in the first row) or 72 (third square in the eighth row) (note that Place has a maximum value of 77 and a minimum value of 0, and the units digit cannot exceed 7). This function will return a one-dimensional array representing all the reachable positions on the chessboard for the given pawn, e.g.: [21, 31].
4. rookMovement(Board, Colour, Place): Similar to pawnMovement, but it checks the movement of a rook.
5. bishopMovement(Board, Colour, Place): Similar to pawnMovement, but it checks the movement of a bishop.
6. queenMovement(Board, Colour, Place): Similar to pawnMovement, but it checks the movement of a queen. It returns a two-dimensional array, where the upper half represents horizontal and vertical movements, and the lower half represents diagonal movements, e.g.: [[35, 36, 37, 33, 32, 31, 30, 44, 54, 64, 24], [45, 56, 67, 23, 43, 52, 61, 25]].
7. knightMovement(Board, Colour, Place): Similar to pawnMovement, but it checks the movement of a knight.
8. kingMovement(Board, Colour, Place): Similar to pawnMovement, but it checks the movement of a king.
9. pieceMovement(Board, P1, P2): P1 and P2 are standard coordinates used in chess notation (e.g., a2, b5). Board is a two-dimensional array created by the createBoard() function. This function combines the functionality of the previous six functions. It returns True if the chess piece at position P1 can move to position P2, and False otherwise.
10. kingCanNotGo(Board, Colour): Board is a two-dimensional array created by the createBoard() function, and Colour indicates the color to be checked (an integer value of 0 or 1, where 0 represents black and 1 represents white). This function returns a one-dimensional array indicating the areas where the king of the specified color cannot move to (i.e., moving to those areas would result in check).
11. checkKing(Board, Colour): Board is a two-dimensional array created by the createBoard() function, and Colour indicates the color to be checked. This function returns a boolean value indicating whether the king of the specified color is in check.
12. castling(Board, P1, P2): This function checks whether the pieces P1 and P2 on the chessboard Board (created by the createBoard() function) can perform a castling move. P1 and P2 are the coordinates of the two pieces, with both the tens and units digits ranging from 0 to 7. It returns a boolean value.
13. pieceMove(Board, P1, P2): P1 and P2 are standard coordinates used in international chess notation (e.g., a1, b3). Board is the chessboard created by the createBoard() function. This function first checks for check (if the king is in check, it returns "Move your king first"), then checks if the chess piece at P1 can move to position P2 (if not, it returns "Can't move"). If the move is valid, it returns a modified two-dimensional array (Board) representing the updated positions of the chess pieces.
14. hasValidMoves(Board, Colour): Board and Colour are defined as above. This function returns a boolean value. If there are still valid moves for the pieces of the specified color (i.e., no move would result in the king being in check), it returns True; otherwise, it returns False.
15. pawnChange(Board, Colour, Place, Kind): Board, Colour, and Place are defined as above. Kind represents the type of chess piece (possible values: "Pawn", "Rook", "Bishop", "Knight", "Queen", "King"). This function promotes a pawn to a chess piece of the specified Kind if the pawn has reached the opponent's baseline. If the chess piece at Place is not a pawn or has not reached the opponent's baseline, no changes are made. It returns the modified chessboard (Board) after the pawn promotion.
16. gameSituation(Board): Board are defined as above。Show the chess situation at this time，Its possible returns are”White wins”, ”Black wins”, ”Draw” , ”Continue”