## Piece

color : bool
name : str
pieceHeld : bool
pos : tuple
x
y

get_diagonal_moves(board)
get_ortogonal_moves(board)
get_pos()
move(to, board)
moveIsPossible(move, board)
set_pos(pos)

## Bishop

moves : list
name : str
x
y

get_valid_moves(board)

## King

canCastle : list
first_move : bool
inCheck : bool
moves : list
name : str
x
y

getNormalValidMoves(board)
get_valid_moves(board)

## Knight

moves : list
name : str
x
y

get_valid_moves(board)

## Pawn

canEnPasseant : NoneType
first_move : bool
moves : list
name : str
x
y

get_valid_moves(board)

## Queen

first_move : bool
moves : list
name : str
x
y

get_valid_moves(board)

## Rook

first_move : bool
moves : list
name : str
x
y

get_valid_moves(board)

## Board

blackGhostPawn : tuple, NoneType, bool
board : list
turn : bool
whiteGhostPawn : tuple, NoneType, bool

activateGhostPawn(pos, color)
board2str()
deactivateGhostPawn(color)
getControlledSquares(color)
getGhostPawn(color)
getKingPiece(color)
getRooks(color)
print_board()
vector()

## Chess

board : object
boardStates : dict
castlingRights : list
gameRunning : bool
legalMoves : list
movesList : list
movesWithoutCapturesOrPawnMovements : int
turn : bool

checkEndgameConditions()
checkMaterialDraw()
checkThreeFoldRepetition()
getLegalMoves()
getMovePlayer()
getMoveRandom()
kingsInCheck()
main()
moveGenerationTest(depth)
playBruteForce()
playCLI(getMove)
playGUI()
playMove(move)
printTurnDecorator()
uci2move(uci_move)

## GUI

board : object
captureVisualIndicator
cell : int
game : object
inCheckVisualIndicator
lastMoveFrom : NoneType
lastMoveTo : tuple, NoneType
pieceHeld : NoneType
piecesDict : dict
playable : bool
screen
spriteSheetPieceCoordinates : list
spritesheet
xoffset : int
yoffset : int

draw()
drop_piece()
getMousePos()
getPiecePixelPos(piece)
getPieceSpriteCoordinates(piece)
getSpriteSheet()
get_piece()
main()
mainUnplayable(tick)