

Game
<ul style="list-style-type: none"> - moveHistory: Array<Board>:status: - moveCount: int - status: int - statusText: String - statusDict: Dictionary
<ul style="list-style-type: none"> + getMoveCount(): int + setStatus(status): void + checkMove(piece, square): int + undo(): void

Board
<ul style="list-style-type: none"> - squares: Array<Square> - whitePiecesCaptured: Array<Piece> - blackPiecesCaptures: Array<Piece>
<ul style="list-style-type: none"> + isCaptured(piece): Boolean + setSquare(name, piece)

Square
<ul style="list-style-type: none"> - name: String - x: int - y: int - occupant: Piece
<ul style="list-style-type: none"> + getOccupant(): Piece + setOccupant(piece): void + isEmpty(): Boolean + clear(): void + coordinatesToName(x, y): String + nameToCoordinates(String): Object

MateChecker
<ul style="list-style-type: none"> - whiteKingChecked: Boolean = false - blackKingChecked: Boolean = false - piecesCheckingKing: Array<Piece>
<ul style="list-style-type: none"> findPiecesCheckingKing(): Array<Piece> isPlayerCheckedmated(playerIsWhite): Boolean isPlayerStalemated(playerIsWhite): Boolean

Piece
<ul style="list-style-type: none"> # location: Square # isWhite: Boolean
<ul style="list-style-type: none"> - capture(piece): void + move(square): void + getLocation(): Square + isWhite(): Boolean + computeMoves(): Array<Square>

MoveSensitivePiece
hasMoved: Boolean = false
+ hasMoved(): Boolean

Queen
+ computeMoves(): Array<Square> <<override>>

King
+ computeMoves(): Array<Square> <<override>>

Bishop
+ computeMoves(): Array<Square> <<override>>

Rook
+ computeMoves(): Array<Square> <<override>>

Knight
+ computeMoves(): Array<Square> <<override>>

Pawn
<ul style="list-style-type: none"> + computeMoves(): Array<Square> <<override>> - promote(): void

How do I deal with checking if the king is being checked/mated? Particularly discovered checks and mates will be tricky as they depend on other pieces. Bubble that responsibility up to the board, which checks all pieces to see if they are checking/mating a king under a proposed move?

Perform the move regardless, then check validity, then revert if invalid? Should be fine as long as we only have app.js redraw board AFTER this is all done.

How to detect stalemate? check for any moves available among opponent's pieces.