Command Design Pattern

HistoryManager

- undoStates : Stack<State>
- redoStates : Stack<State>

+ undoSave(game: ChessGame)
+ redoSave(game: ChessGame)
+ undo(game: ChessGame)
+ redo(game: ChessGame)
+ clearRedo()

«interface» Command

+ execute()

RedoCommand

- historyManager: HistoryManager

+ execute(game: ChessGame)

UndoCommand

- HistoryManager

+ execute(game: ChessGame)

ChessGUI

- panel: JPanel
- frame: JFrame
- historyManager: HistoryManager

+ panelSetup()
+ frameSetup()
+ boardSetup()
+ gameField()
+ keyActions()
+ MouseActions()
+ moveGUI(from: String, to: String, promoteTo: Char):

Piece

-moveNum: int
-position: String
-color: boolean
-isCaptured: boolean

+ isValidMove(from: String, to: String): boolean
+ increaseMoveNum()

Memento Design Pattern

1

0..*

State

- board : Pieces[ ][ ]
- lastMoved : int[ ]
- whiteTurn: boolean
- gameOver: boolean

ChessGame

- board : Pieces[ ][ ]
- lastMoved : int[ ]
- whiteTurn: boolean
- gameOver: boolean

+ save(): State
+ revert(prevState: State)
+ move(from: String, to: String, promote to : char): String
+ draw(): boolean
+ isStalemate(): boolean
+ promote(row: int, col: int, promote to : char):

Builder Design Pattern

<<Creates>>

BoardInitializer

+ boardInit(): Piece[ ] [ ]
+ generateBoard(): char[ ] [ ]

Pawn

+ isValidMove(from: String, to: String): boolean
+ toString(): String

Bishop

+ isValidMove(from: String, to: String): boolean
+ toString(): String

Knight

+ isValidMove(from: String, to: String): boolean
+ toString(): String

King

+ isValidMove(from: String, to: String): boolean
+ toString(): String

Queen

+ isValidMove(from: String, to: String): boolean
+ toString(): String

Knight

+ isValidMove(from: String, to: String): boolean
+ toString(): String