Diagramme UML

Board + COLUMN_NOTATION : Character[] + PIECE_NOTATION : Character[] + PIECE PRINT : Character[] + ROW_NOTATION : Character[] + startFEN : String + testFEN : String + test2FEN : String + failSaveFEN : String + checkmateFEN : String + stalemateFEN : String + weirdPositionFEN : String + pawnDebudFEN : String + castleFEN : String + enPassantInitFEN : String + promotionFEN : String + ambitiousFEN : String + previousBoard : Board + previousMove : Move + notation : String + fen : String + board : Tile[][] + colorActive : int + castleAvailable : boolean[] + enPassantTileCoordinate : int[] + halfMoveCounter : int + fullMoveCounter : int + whiteMoves : LinkedList<Move> + blackMoves : LinkedList<Move> + whitePieces : LinkedList<Move> + blackPieces : LinkedList<Move> + whiteKingCoordinate : int[] + blackKingCoordinate : int[] + isWhiteInCheck : boolean + isBlackInCheck : boolean + isWhiteInCheckMate : boolean + isBlackInCheckMate : boolean + isWhiteInStaleMate : boolean + isBlackInStaleMate: boolean + castleCurrentAvailable : boolean[] + eliminateMove : boolean + searchDepth : int + boardStateEvaluation : double + treeStateEvaluation : double + optiMalMove : Move + nextBoardSet : TreeSet<Board> + Board(String fen)

+ Board(Board previousBoard, Move move, boolean eliminateMove, int searchDepth) + getPiece(int color) : LinkedList<Piece> getMove(LinkedList<Piece> listPiece) : LinkedList<Move> + getKingPosition(int color) : int[] + isInCheck(int color) : boolean + assignPiece(int[] destinationCoordinate, int currentId) : Piece + limitMoveResultInCheck(int color): void + isInCheckMate(int color) : boolean + isInStalemate(int color) : boolean + checkCastleAvailable(int color): void + fenToBoard(String fen) : void + BoardToFen() : String + calculateValue(): double + printBoard() : void + printBoard1() : void + getNotation() : String + compareTo(Board b) : int

Move + board : Board + notation : String + piece : Piece + affectedPiece : Piece + transformedPieceId : int + startingTile : Tile + destinationTile : Tile affectedStartingTile : Tile affectedDestinationTile : Tile + type : int + Move(Piece piece, Tile startingTile , Tile destinationTile) + Move(Piece piece, Piece affectedPiece, Tile startingTile, Tile destinationTile, Tile affectedStaringTile, Tile affectedDestinationTile) + Move(Piece piece, int transformedPieceId, Tile startingTile, Tile destinationTile) + equals(Move move) : boolean + toString() : String

Tile

+ Tile (int[] tileCoordinate)

+ equals (Tile tile): boolean

OccupiedTile

+ OccupiedTile (int[] tileCoordinate, Piece

pieceOnTile)

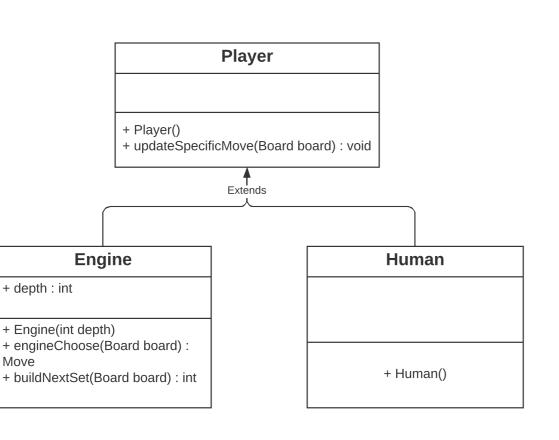
+ tileCoordinate : int[]

+ pieceOnTile : Piece

EmptyTile

+ EmptyTile (int[] tileCoordinate)

Loop + white : Player + black : Player + listBoard : List<Board> + Loop(Player white, Player black) + Loop() + gameLoop(Board board, Scanner sc): void + gameLoopInterface(Board board, Interface interFace) : void + gameLoopHumanMachine(Board board, Interface interFace, Player whitePlayer, Player blackPlayer): void + gameLoopHumanMachineBetter(Board board, Interface interFace, Player whitePlayer, Player blackPlayer): void + gameEnd(Board board) : int + getPosition(String s, Board board) : Tile



Client + scanner : Scanner + moveNumber : int + Client()

Server + PORT : int + scanner : Scanner + moveNumber : int + chosenMove : Move + Server()

+ updateInterface(Board board) : void

+ getTempPartie() : JComboBox

Interface Chessboard piecelcons : Image[] + chessBoard : ChessBoard + board : Board + history : History + startingCoordination : int [] + button : JButton + destinationCoordination : int [] + startingTextField : JTextField + tileChose : boolean + destinationTextField : JTextField + highlightTile : int [] + buttonActivated : boolean + timerAnimation: Timer + piecelconType : Image + startingInput : String + destinationInput : String + xPiece : int + Parameters : JPanel + yPiece : int + ChoixPage1 : JButton + xGoal : int + ChoixPage2 : JButton + yGoal : int + ChoixPage3 : JButton + xVelocity : int + BJouer : JButton + yVelocity : int + TempPartie : JComboBox + isAnimating : boolean + ElementsTempPartie : Object[] + affectedPieceIconType : Image + affectedXPiece : int + Recherche : JTextField + Page1 : JPanel + affectedYPiece : int + Page2 : JPanel + affectedXGoal : int + Page3 : JPanel + affectedYGoal : int + AfftimerB : JPanel + affectedXVelocity : int + AfftimerW : JPanel + affectedYVelocity : int + timer1 : ChessTime + isInGame : boolean + dottedTiles : LinkedList<Integer[]> + timer2 : ChessTime + TimerB : JLabel + definedMove : Move + TimerW : JLabel + theme : int + themeChessCom : Color[] + timerUpdate : Timer + menu1 : CustomizeMenu + themeLichess : Color [] + themePurple : Color [] + menu2 : CustomizeMenu + color1 : Color + changeTheme : JButton + color2 : Color + changeLanguage : JButton + color3 : Color + currentLanguage : JLabel + color4 : Color + languageList : String[] + languageChoice : int + color5 : Color + taskBar1 : JPanel + color6 : Color + taskBar2 : JPanel + Chessboard(Board board) + name1 : JLabel + Chessboard(Board board, int theme) + name2 : JLabel + paintComponent(Graphics g) : void + avatar1 : JLabel + loadImage(): void + avatar2 : JLabel + updateChessBoard(Board board) : void + draw : JButton + resetChessboard(): void + resign : JButton + drawCenteredCircle(Graphics2D g, int x, int y, int r): + flag : ImageIcon + drawlcon : Imagelcon + mouseClicked(MouseEvent e) : void + mousePressed(MouseEvent e): void Interface(Board board) + mouseReleased(MouseEvent e) : void + actionPerformed(ActionEvent e) : void + mouseEntered(MouseEvent e) : void

+ drawHollowCircle(Graphics2D g, int x, int y, int r) : void

+ mouseExited(MouseEvent e) : void

+ actionPerformed(ActionEvent e) : void

ChooseAvatar + frame : JFrame + panel1 : JPanel + panel2 : JPanel + text : JTextField + button1 : JButton + button2 : JButton + button3 : JButton + button4 : JButton + button5 : JButton + button6 : JButton + avatar1 : ImageIcon + avatar2 : ImageIcon + avatar3 : Imagelcon + avatar4 : Imagelcon + avatar5 : ImageIcon + avatar6 : Imagelcon + avatarChosen : ImageIcon + customizeMenu : CustomizeMenu + ChooseAvatar(CustomizeMenu customizeMenu) + ChooseAvatar() + actionPerformed(ActionEvent e) : void ChooseName

+ question : JButton

+ text : JTextField

+ panel : JPanel

+ name : String

+ ChooseName()

+ keyTyped(KeyEvent e) : void

+ keyPressed(KeyEvent e) : void

+ keyReleased(KeyEvent e) : void

+ actionPerformed(ActionEvent e) : void

ChessTime + minite : int seconde : int - Mytimer : Timer FinDeLaPartie: WindowTheEnd + task : TimerTask + paused : boolean + ChessTime(int m, int s) + start() : void restart() : void freezetime() : void History

- moves : LinkedList<String>

+ History()

CustomizeMenu

+ chooseAvatar : ChooseAvatar

+ checkForAvatar : Timer

+ avatarlcon : Imagelcon

+ CustomizeMenu(int player)

+ actionPerformed(ActionEvent e):

+ isInGame : boolean

+ player : int

+ avatar : JButton

+ label : JLabel

+ name : JTextField

WindowTheEnd + WindowTheEnd()

paintComponent(Graphics g) : void

