

PieceType «enumeration»
PAWN
ROOK
KNIGHT
BISHOP
QUEEN
KING

GameStatus «enumeration»
PLAYING
CHECKMATE
STALEMATE
MATERIL_LACK
REPETITION

PieceColor «enumeration»
WHITE
BLACK
VOID

BoardRenderer «strategy»
- whitePiecesChars : map - blackPiecesChars : map
+ BoardRenderer(whitePiecesChars : map, blackPiecesChars : map) + ~BoardRenderer() + PrintBoard(playerColor : PieceColor) : void - PieceToString(piece : shared_ptr, inverted : bool) : string_view - PieceVectorToString(pieceVector : vector, inverted : bool) string - {abstract} PrintWhiteBoard() : void - {abstract} PrintBlackBoard() : void

NormalBoardRenderer
+ NormalBoardRenderer() + ~NormalBoardRenderer() - PrintWhiteBoard() : void - PrintBlackBoard() : void

SimplifiedBoardRenderer
+ SimplifiedBoardRenderer() + ~SimplifiedBoardRenderer() - PrintWhiteBoard() : void - PrintBlackBoard() : void

ColoredBoardRenderer
+ ColoredBoardRenderer() + ~ColoredBoardRenderer() - PrintWhiteBoard() : void - PrintBlackBoard() : void

InvertedBoardRenderer
+ InvertedBoardRenderer() + ~InvertedBoardRenderer() - PrintWhiteBoard() : void - PrintBlackBoard() : void

VoidPiece
+ VoidPiece(pPosition : Coordinate) + ~VoidPiece() + IsMoveValid(endingPosition : Coordinate) : bool

Bishop
+ Bishop(pColor : PieceColor, pPosition : Coordinate) + ~Bishop() + IsMoveValid(endingPosition : Coordinate) : bool

Knight
+ Knight(pColor : PieceColor, pPosition : Coordinate) + ~Knight() + IsMoveValid(endingPosition : Coordinate) : bool

Queen
+ Queen(pColor : PieceColor, pPosition : Coordinate) + ~Queen() + IsMoveValid(endingPosition : Coordinate) : bool

Rook
- hasMoved : bool
+ Rook(pColor : PieceColor, pPosition : Coordinate, hasMoved : bool) + ~Rook() + IsMoveValid(endingPosition : Coordinate) : bool + CanCastle() : bool + Move(newPosition : Coordinate) : void

King
- hasMoved : bool
+ King(pColor : PieceColor, pPosition : Coordinate, hasMoved : bool) + ~King() + IsMoveValid(endingPosition : Coordinate) : bool + CanCastle() : bool + Move(newPosition : Coordinate) : void

Pawn
- hasMoved : bool - doubleAdvancementMoveNumber : int
+ Pawn(pColor : PieceColor, pPosition : Coordinate, hasMoved : bool) + ~Pawn() + IsMoveValid(endingPosition : Coordinate) : bool + Move(newPosition : Coordinate) : void + GetDoubleAdvancementMoveNumber() : int

BoardFactory «factory»
+ BoardFactory() + ~BoardFactory() + LoadFenPosition(fenString : string_view) : void + InitializeStartingBoard() : void + <u>makePiece(pieceChar : char, pPosition : Coordinate, hasRookMoved : bool = true) : shared_ptr</u> - LoadBoardPosition(boardString : string_view) : void

GameManager
- activePlayerColor : PieceColor - gameStatus : GameStatus - boardRenderer : BoardRenderer - boardFactory : BoardFactory - PastPositions : vector - welcomeFile : fstream - helpFile : fstream - settingsFile : fstream - endFile : fstream
+ GameManager() + ~GameManager() - HelpUser() : void - UserSettings() : void - GetUserMove() : void - GameLoop() : void - KillGame() : void - UpdateGameStatus() : void + StartGame() : void

Board «singleton»
- squaresMap : map - whitePieces : vector - blackPieces : vector - whiteCapturedPieces : vector - blackCapturedPieces : vector - whiteKing : shared_ptr - blackKing : shared_ptr - moveNumber : int
- Board() + <u>Instance()</u> : Board& - Castling(king : shared_ptr, kingEndingPosition : Coordinate) : void - EnPassant(pawn : shared_ptr, pawnEndingPosition : Coordinate) : void - IsSquareAttacked(square : Coordinate, attackerColor : PieceColor) : bool + UpdateSquare(position : Coordinate, piece : Piece) : void + UpdatePiecesVector() : void + AddKings(whiteKingPosition : Coordinate, blackKingPosition : Coordinate) : void + IsKingInCheck(kingColor : PieceColor) : bool + HasValidMoves(playerColor : PieceColor) : bool + IsMaterialLacking() : bool + NormalMove(movingPiece : shared_ptr, endingPosition : Coordinate) : void + Promotion(pawn : shared_ptr, promotionPiece : char, endingPosition : Coordinate) : void + GetPiece(position : Coordinate) : shared_ptr + ClearBoard() : void + ResetMoveNumber() : void + IncrementMoveNumber(increment : int) : void + GetMoveNumber() : int + GetCapturedPieces(pColor : PieceColor) : vector

Piece
# color : PieceColor # pieceType : Piecetype # position : Coordinate # literal : Char
+ ~Piece() + {abstract} IsMoveValid(endigPosition : Coordinate) : bool + GetColor() : PieceColor + GetType() : PieceType + GetPosition() : Coordinate + GetChar() : Char + {virtual} Move(newPosition : Coordinate) : void + {virtual} CanCastle() : bool + {virtual} GetDoubleAdvancementMoveNumber() : int

Coordinate
- <u>literalChars</u> : string - x : int - y : int
+ Coordinate(newX : int, newY : int) + Coordinate(literalExpression : string_view) + ~Coordinate() + Coordinate(newCoordinate : Coordinate) + GetX() : int + GetY() : int + operator=(newCoordinate : Coordinate) : Coordinate& + operator<(other : Coordinate) : bool + operator>(other : Coordinate) : bool + operator==(other : Coordinate) : bool + operator!=(other : Coordinate) : bool + operator<=(other : Coordinate) : bool + operator>=(other : Coordinate) : bool + operator+(movement : Movement) : Coordinate + operator+=(movement : Movement) : Coordinate& + SquaredDistance(other : Coordinate) : int + ToString() : string;

Movement
- x : int - y : int
+ Movement(newX : int = 0, newY : int = 0) + Movement(newMovement : Movement) + ~Movement() + GetX() : int + GetY() : int + operator=(newMovement : Movement) : Movement& + operator<(other : Movement) : bool + operator>(other : Movement) : bool + operator==(other : Movement) : bool + operator!=(other : Movement) : bool + operator<=(other : Movement) : bool + operator>=(other : Movement) : bool + operator*(multiplier : int) : Movement