My Project

Generated by Doxygen 1.8.2

Wed Sep 27 2017 16:28:38

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

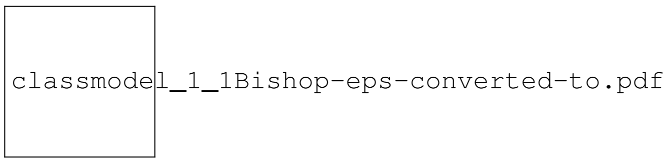
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 model.Bishop Class Reference

Inheritance diagram for model.Bishop:



Public Member Functions

- Bishop (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Bishop (Piece otherPiece)
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()

Additional Inherited Members

3.1.1 Constructor & Destructor Documentation

3.1.1.1 model.Bishop.Bishop (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of Bishop, inherited from abstract Piece constructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.1.1.2 model.Bishop.Bishop (Piece *otherPiece*) `[inline]`

copy constructor for Bishop

Parameters

otherPiece	

3.1.2 Member Function Documentation

3.1.2.1 String model.Bishop.getFullName ( ) `[inline]`, `[virtual]`

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.1.2.2 boolean model.Bishop.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[inline]`, `[virtual]`

determine whether the current move is valid by myPiece Bishop

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/Bishop.java

3.2 view.BoardView Class Reference

Inheritance diagram for view.BoardView:

classview_1_1BoardView-eps-converted-to.pdf

Public Member Functions

- BoardView (int width, int height, ChessBoard myChessBoard)
- void addBoard (ChessBoard currBoard)
- JPanel createBox (int i, int j)
- void addMouseControl (MouseListener listener)
- void highLightBut (int xPos, int yPos)
- void unhighLightBut (int xPos, int yPos)
- void addElement ()

Public Attributes

- JPanel **BoardSpace**
- JPanel[ ][ ] **grid**
- PieceView[ ][ ] **buttons**
- Vector< JButton > **controlButs**

3.2.1 Constructor & Destructor Documentation

3.2.1.1 view.BoardView.BoardView (int *width*, int *height*, ChessBoard *myChessBoard*) `[inline]`

default constructor for BoardView

Parameters

width	
height	
myChessBoard	

3.2.2 Member Function Documentation

3.2.2.1 void view.BoardView.addBoard (ChessBoard *currBoard*) `[inline]`

helper function to add BoardSpace

Parameters

currBoard	

3.2.2.2 void view.BoardView.addElement ( ) `[inline]`

generate click buttons on the boardView

3.2.2.3 void view.BoardView.addMouseControl (MouseListener *listener*) `[inline]`

to add mouse control unit to the board

Parameters

listener	

3.2.2.4 JPanel view.BoardView.createBox (int *i*, int *j*) `[inline]`

helper function to generate click button

Parameters

i	
j	

Returns

3.2.2.5 void view.BoardView.highLightBut (int *xPos*, int *yPos*) `[inline]`

high light current button with yPos and xPos

Parameters

xPos	
yPos	

3.2.2.6 void view.BoardView.unhighLightBut (int *xPos*, int *yPos*) `[inline]`

unhigh light current button with yPos and xPos

Parameters

xPos	
yPos	

The documentation for this class was generated from the following file:

- src/main/java/view/BoardView.java

3.3 view.BoardViewTest Class Reference

Inheritance diagram for view.BoardViewTest:

`classview_1_1BoardViewTest-eps-converted-to.pdf`

Public Member Functions

- void **testAddBoard** () throws Exception
- void **testCreateBox** () throws Exception
- void **testAddMouseControl** () throws Exception
- void **testHighLightBut** () throws Exception
- void **testUnhighLightBut** () throws Exception
- void **testAddElement** () throws Exception

The documentation for this class was generated from the following file:

- src/test/java/view/BoardViewTest.java

3.4 model.Cannon Class Reference

Inheritance diagram for model.Cannon:

classmodel_1_1Cannon-eps-converted-to.pdf

Public Member Functions

- Cannon (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Cannon (Piece otherPiece)
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()
- boolean isJumpPossible (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean mustCapture)

Additional Inherited Members

3.4.1 Constructor & Destructor Documentation

3.4.1.1 model.Cannon.Cannon (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of Cannon, inherited from abstract Piece constructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.4.1.2 model.Cannon.Cannon (Piece *otherPiece*) `[inline]`

copy constructor for Knight

Parameters

otherPiece	

3.4.2 Member Function Documentation

3.4.2.1 String model.Cannon.getFullName ( ) `[inline]`, `[virtual]`

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.4.2.2 boolean model.Cannon.isJumpPossible (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *mustCapture*) `[inline]`

check whether it's valid to move in linear directions like a rook

Parameters

myPiece	
myChessBoard	
destX	
destY	
mustCapture	

Returns

3.4.2.3 boolean model.Cannon.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[inline]`, `[virtual]`

determine whether the current move is valid by myPiece Cannon

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/Cannon.java

3.5 model.ChessBoard Class Reference

Public Member Functions

- ChessBoard (boolean isCustom)
- ChessBoard (ChessBoard otherBoard)
- boolean isEqual (ChessBoard otherBoard)
- Piece copyPiece (Piece otherPiece)
- boolean isPiece (int xPos, int yPos)

- void emptyBoard ()
- Piece getPiece (int xPos, int yPos)
- void addPiece (int xPos, int yPos, Piece addPiece)
- Piece removePiece (int xPos, int yPos)
- void moveChessPiece (int origX, int origY, int destX, int destY)
- Vector< Piece > getAllPieces (Color sideColor)

3.5.1 Constructor & Destructor Documentation

3.5.1.1 model.ChessBoard.ChessBoard (boolean *isCustom*) `[inline]`

default constructor fro ChessBoard

3.5.1.2 model.ChessBoard.ChessBoard (ChessBoard *otherBoard*) `[inline]`

copy constructor for ChessBoard

Parameters

otherBoard	

3.5.2 Member Function Documentation

3.5.2.1 void model.ChessBoard.addPiece (int *xPos,* int *yPos,* Piece *addPiece*) `[inline]`

add the model with xPos and yPos to the current chessBoard

Parameters

xPos	
yPos	
addPiece	

3.5.2.2 Piece model.ChessBoard.copyPiece (Piece *otherPiece*) `[inline]`

make a deep copy of otherPiece

Parameters

otherPiece	

Returns

3.5.2.3 void model.ChessBoard.emptyBoard ( ) `[inline]`

determine whether the current chessBoard is empty

3.5.2.4 Vector<Piece> model.ChessBoard.getAllPieces (Color *sideColor*) `[inline]`

get all pieces from one side

Parameters

sideColor	

Returns

3.5.2.5 Piece model.ChessBoard.getPiece (int *xPos*, int *yPos*) `[inline]`

get the model with xPos and yPos from the current ChessBoard

Parameters

xPos	
yPos	

Returns

3.5.2.6 boolean model.ChessBoard.isEqual (ChessBoard *otherBoard*) `[inline]`

determine whether otherBoard object is "approximately" equal to current ChessBoard

Parameters

otherBoard	

Returns

3.5.2.7 boolean model.ChessBoard.isPiece (int *xPos*, int *yPos*) `[inline]`

determine whether there exits a model on chessBoard with xPos and yPos

Parameters

xPos	
yPos	

Returns

3.5.2.8 void model.ChessBoard.moveChessPiece (int *origX*, int *origY*, int *destX*, int *destY*) `[inline]`

move the model from original position to destination position

Parameters

origX	
origY	
destX	
destY	

3.5.2.9 Piece model.ChessBoard.removePiece (int *xPos*, int *yPos*) `[inline]`

remove the model with xPos and yPos to the current chessBoard

Parameters

xPos	
yPos	

Returns

The documentation for this class was generated from the following file:

- src/main/java/model/ChessBoard.java

3.6 controller.ChessBoardTest Class Reference

Inheritance diagram for controller.ChessBoardTest:

`classcontroller_1_1ChessBoardTest-eps-converted-to.pdf`

Public Member Functions

- void **testCopyPiece** () throws Exception
- void **testIsPiece** () throws Exception
- void **testGetPiece** () throws Exception
- void **testRemovePiece** () throws Exception
- void **testMoveChessPiece** () throws Exception
- void **testGetAllPieces** () throws Exception

The documentation for this class was generated from the following file:

- src/test/java/controller/ChessBoardTest.java

3.7 model.Elephant Class Reference

Inheritance diagram for model.Elephant:

classmodel_1_1Elephant-eps-converted-to.pdf

Public Member Functions

- Elephant (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Elephant (Piece otherPiece)
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()

Additional Inherited Members

3.7.1 Constructor & Destructor Documentation

3.7.1.1 model.Elephant.Elephant (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of Elephant, inherited from abstract Piece constructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.7.1.2 model.Elephant.Elephant (Piece *otherPiece*) `[inline]`

copy constructor for Elephant

Parameters

otherPiece	

3.7.2 Member Function Documentation

3.7.2.1 String model.Elephant.getFullName ( ) `[inline], [virtual]`

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.7.2.2 boolean model.Elephant.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[inline], [virtual]`

determine whether the current move is valid by myPiece Elephant

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/Elephant.java

3.8 controller.Frame Class Reference

Public Member Functions

- Frame (ChessBoard currBoard, Color currColor)
- Status analyzeMove (ChessBoard currBoard, Color currColor)
- Status selectOrig (PieceView origBut)
- Status selectDest (PieceView destBut)
- Vector< Integer > availableMoves ()
- ChessBoard nextFrame ()

Public Attributes

- ChessBoard **currBoard**
- Color **currColor**
- Move **currMove**
- PieceView **origBut**
- PieceView **destBut**
- int **origX**
- int **origY**
- int **destX**
- int **destY**

3.8.1 Constructor & Destructor Documentation

3.8.1.1 controller.Frame.Frame (ChessBoard *currBoard,* Color *currColor*) `[inline]`

constructor for Frame

Parameters

currBoard	
currColor	

3.8.2 Member Function Documentation

3.8.2.1 Status controller.Frame.analyzeMove (ChessBoard *currBoard,* Color *currColor*) `[inline]`

analyze current move and return game end condition

Parameters

currBoard	
currColor	

Returns

3.8.2.2 Vector<Integer> controller.Frame.availableMoves ( ) `[inline]`

Returns

all the available positions of current piece

3.8.2.3 ChessBoard controller.Frame.nextFrame ( ) `[inline]`

Returns

the chessboard condition of next frame

3.8.2.4 Status controller.Frame.selectDest (PieceView *destBut*) `[inline]`

select destination piece, return the status(successful or fail)

Parameters

destBut	

Returns

3.8.2.5 Status controller.Frame.selectOrig (PieceView *origBut*) `[inline]`

select original piece, return the status(successful or fail)

Parameters

origBut	

Returns

The documentation for this class was generated from the following file:

- src/main/java/controller/Frame.java

3.9 controller.GamePlay Class Reference

Public Member Functions

- GamePlay ()
- ChessBoard proceedGame (Color currColor)
- Status currGamePlay (Scanner reader, ChessBoard currBoard, Color currColor)
- int safeScan (Scanner reader)
- Status analyzeMove (Scanner reader, ChessBoard currBoard, Color currColor)

Public Attributes

- Scanner **reader**
- ChessBoard **currBoard**
- Piece **currPiece**
- Color **currColor**
- Move **currMove**
- int **destX**
- int **destY**

3.9.1 Constructor & Destructor Documentation

3.9.1.1 controller.GamePlay.GamePlay ( ) `[inline]`

default constructor for GamePlay

3.9.2 Member Function Documentation

3.9.2.1 Status controller.GamePlay.analyzeMove (Scanner *reader,* ChessBoard *currBoard,* Color *currColor*) `[inline]`

help function for current move

Parameters

reader	
currBoard	
currColor	

Returns

3.9.2.2 Status controller.GamePlay.currGamePlay (Scanner *reader,* ChessBoard *currBoard,* Color *currColor*) `[inline]`

get the designate move by commandline or file, check whether it's valid and controller end condition, you have 3 chances to enter correct moves or the controller will end;

Parameters

reader	
currBoard	
currColor	

Returns

3.9.2.3 ChessBoard controller.GamePlay.proceedGame (Color *currColor*) `[inline]`

proceed controller with stored move

Parameters

currColor	

Returns

3.9.2.4 int controller.GamePlay.safeScan (Scanner *reader*) `[inline]`

a safe version to use Scanner.nextIn

Parameters

reader	

Returns

The documentation for this class was generated from the following file:

- src/main/java/controller/GamePlay.java

3.10 controller.GameTest Class Reference

Inheritance diagram for controller.GameTest:

classcontroller_1_1GameTest-eps-converted-to.pdf

Public Member Functions

- void **testPrintBoard** () throws Exception
- void **testMain** () throws Exception
- void **testMain2** () throws Exception

The documentation for this class was generated from the following file:

- src/test/java/controller/GameTest.java

3.11 model.King Class Reference

Inheritance diagram for model.King:

classmodel_1_1King-eps-converted-to.pdf

Public Member Functions

- King (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- King (Piece otherPiece)
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()

Additional Inherited Members

3.11.1 Constructor & Destructor Documentation

3.11.1.1 model.King.King (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of King, inherited from abstract Piece constructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.11.1.2 model.King.King (Piece *otherPiece*) `[inline]`

copy constructor for King

Parameters

otherPiece	

3.11.2 Member Function Documentation

3.11.2.1 String model.King.getFullName ( ) `[inline]`, `[virtual]`

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.11.2.2 boolean model.King.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[inline]`, `[virtual]`

determine whether the current move is valid by myPiece King

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/King.java

3.12 model.Knight Class Reference

Inheritance diagram for model.Knight:

classmodel_1_1Knight-eps-converted-to.pdf

Public Member Functions

- Knight (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Knight (Piece otherPiece)
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()

Additional Inherited Members

3.12.1 Constructor & Destructor Documentation

3.12.1.1 model.Knight.Knight (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of Knight, inherited from abstract Piece constructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.12.1.2 model.Knight.Knight (Piece *otherPiece*) `[inline]`

copy constructor for Knight

Parameters

otherPiece	

3.12.2 Member Function Documentation

3.12.2.1 String model.Knight.getFullName ( ) `[inline]`, `[virtual]`

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.12.2.2 boolean model.Knight.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[inline]`, `[virtual]`

determine whether the current move is valid by myPiece Knight

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/Knight.java

3.13 controller.MainControl Class Reference

Static Public Member Functions

- static void **main** (String[] args) throws IOException

The documentation for this class was generated from the following file:

- src/main/java/controller/MainControl.java

3.14 model.Move Class Reference

Public Member Functions

- Move ()
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- Vector< Piece > findPiece (char type, Color sideColor, ChessBoard myChessBoard)
- boolean isBeingChecked (ChessBoard myChessBoard, Color myColor)
- boolean isBeingStalemate (ChessBoard myChessBoard, Color myColor)
- boolean isCheckmate (ChessBoard myChessBoard, Color myColor)
- boolean willbeChecked (Piece myPiece, ChessBoard myChessBoard, int destX, int destY)
- Vector< Integer > availableMoves (Piece myPiece, ChessBoard myChessBoard, boolean conCheck)
- boolean isInBound (int destX, int destY)

3.14.1 Constructor & Destructor Documentation

3.14.1.1 model.Move.Move ( ) `[inline]`

default constructor for Move

3.14.2 Member Function Documentation

3.14.2.1 Vector<Integer> model.Move.availableMoves (Piece *myPiece,* ChessBoard *myChessBoard,* boolean *conCheck*) `[inline]`

get all available moves from myPiece

Parameters

myPiece	
myChessBoard	
conCheck	

Returns

3.14.2.2 Vector<Piece> model.Move.findPiece (char *type,* Color *sideColor,* ChessBoard *myChessBoard*) `[inline]`

find specific type of model with side color

Parameters

type	
sideColor	
myChessBoard	

Returns

3.14.2.3 boolean model.Move.isBeingChecked (ChessBoard *myChessBoard,* Color *myColor*) `[inline]`

check whether myColor side is being Checked

Parameters

myChessBoard	
myColor	

Returns

3.14.2.4 boolean model.Move.isBeingStalemate (ChessBoard *myChessBoard,* Color *myColor*) `[inline]`

check whether current controller is a stalemate

Parameters

myChessBoard	
myColor	

Returns

3.14.2.5 boolean model.Move.isCheckmate (ChessBoard *myChessBoard,* Color *myColor*) `[inline]`

check whether myColor side is lost

Parameters

myChessBoard	
myColor	

Returns

3.14.2.6 boolean model.Move.isInBound (int *destX,* int *destY*) `[inline]`

check whether the destination position is inside the chessboard

Parameters

destX	
destY	

Returns

3.14.2.7 boolean model.Move.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[inline]`

determine whether it's valid to move myPiece to destination position with the option of consideration of check.

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

3.14.2.8 boolean model.Move.willbeChecked (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY*) `[inline]`

check whether next move will cause check condition

Parameters

myPiece	
myChessBoard	
destX	
destY	

Returns

The documentation for this class was generated from the following file:

- src/main/java/model/Move.java

3.15 model.MoveTest Class Reference

Inheritance diagram for model.MoveTest:

`classmodel_1_1MoveTest-eps-converted-to.pdf`

Public Member Functions

- void **testIsValidMove** () throws Exception

- void **testFindPiece** () throws Exception
- void **testIsBeingChecked** () throws Exception
- void **testIsBeingStalemate** () throws Exception
- void **testIsCheckmate** () throws Exception
- void **testWillbeChecked** () throws Exception
- void **testAvailableMoves** () throws Exception
- void **testIsInBound** () throws Exception

The documentation for this class was generated from the following file:

- src/test/java/model/MoveTest.java

3.16 model.Pawn Class Reference

Inheritance diagram for model.Pawn:

classmodel_1_1Pawn-eps-converted-to.pdf

Public Member Functions

- Pawn (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Pawn (Piece otherPiece)
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()

Additional Inherited Members

3.16.1 Constructor & Destructor Documentation

3.16.1.1 model.Pawn.Pawn (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of Pawn, inherited from abstract Piece contructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.16.1.2 model.Pawn.Pawn (Piece *otherPiece*) `[inline]`

copy constructor for Pawn

Parameters

otherPiece	

3.16.2 Member Function Documentation

3.16.2.1 String model.Pawn.getFullName ( ) [inline], [virtual]

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.16.2.2 boolean model.Pawn.isValidMove (Piece *myPiece*, ChessBoard *myChessBoard*, int *destX*, int *destY*, boolean *conCheck*) [inline], [virtual]

determine whether the current move is valid by myPiece Pawn

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/Pawn.java

3.17 model.Piece Class Reference

Inheritance diagram for model.Piece:

Public Member Functions

- Piece (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Piece (Piece otherPiece)
- abstract String getFullName ()
- abstract boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- boolean isBoxPossible (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean mustCapture)
- boolean isLinearPossible (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean mustCapture)
- boolean isDiagnalPossible (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean mustCapture)

Public Attributes

- Color **pieceColor**
- int **xPos**
- int **yPos**
- boolean **isFirst**
- char **type**

3.17.1 Constructor & Destructor Documentation

3.17.1.1 model.Piece.Piece (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor for Piece

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.17.1.2 model.Piece.Piece (Piece *otherPiece*) `[inline]`

copy constructor for Piece

Parameters

otherPiece	

3.17.2 Member Function Documentation

3.17.2.1 abstract String model.Piece.getFullName ( ) `[pure virtual]`

Returns

name of the piece eg: black_bishop

Implemented in model.Pawn, model.King, model.Knight, model.Queen, model.Cannon, model.Elephant, model.-Bishop, and model.Rook.

3.17.2.2 boolean model.Piece.isBoxPossible (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *mustCapture*) `[inline]`

check whether it's valid to set the current destination (only consider that box)

Parameters

myPiece	
myChessBoard	
destX	
destY	
mustCapture	

Returns

3.17.2.3 boolean model.Piece.isDiagnalPossible (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *mustCapture*) `[inline]`

check whether it's valid to move in diagnal direction like a bishop

Parameters

myPiece	
myChessBoard	
destX	
destY	
mustCapture	

Returns

3.17.2.4 boolean model.Piece.isLinearPossible (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *mustCapture*) `[inline]`

check whether it's valid to move in linear directions like a rook

Parameters

myPiece	
myChessBoard	
destX	
destY	
mustCapture	

Returns

3.17.2.5 abstract boolean model.Piece.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[pure virtual]`

determine whether it's valid to move myPiece to destination position with the option of consideration of being checked. consider move rule of model only

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implemented in model.Cannon, model.Elephant, model.Queen, model.Bishop, model.King, model.Rook, model.-Knight, and model.Pawn.

The documentation for this class was generated from the following file:

- src/main/java/model/Piece.java

3.18 controller.PieceControl Class Reference

Inheritance diagram for controller.PieceControl:

Public Member Functions

- PieceControl (Frame currFrame, BoardView currBoardView)
- void highLightAvaiPosi ()
- void unhighLightAvaiPosi ()
- void clickControl (JButton currBtn)
- void clickPiece (JButton currBtn)
- void mouseClicked (MouseEvent e)
- void mousePressed (MouseEvent e)
- void mouseReleased (MouseEvent e)
- void mouseEntered (MouseEvent e)
- void mouseExited (MouseEvent e)

Public Attributes

- Status **result**
- PieceView **seleBtn**
- Frame **currFrame**
- BoardView **currBoardView**
- PrintFormat **PF** = new PrintFormat()

3.18.1 Constructor & Destructor Documentation

3.18.1.1 controller.PieceControl.PieceControl (Frame *currFrame*, BoardView *currBoardView*) `[inline]`

constructor for PieceControl

Parameters

currFrame	
currBoardView	

3.18.2 Member Function Documentation

3.18.2.1 void controller.PieceControl.clickControl (JButton *currBtn*) `[inline]`

add mouse control unit

Parameters

currBtn	

3.18.2.2 void controller.PieceControl.clickPiece (JButton *currBtn*) `[inline]`

add some reaction for pieces on chessboard

Parameters

currBtn	

3.18.2.3 void controller.PieceControl.highLightAvaiPosi ( ) `[inline]`

high light current available position on the chess board

3.18.2.4 void controller.PieceControl.mouseClicked (MouseEvent *e*) `[inline]`

Invoked when the mouse button has been clicked (pressed and released) on a component.

Parameters

e	

3.18.2.5 void controller.PieceControl.mouseEntered (MouseEvent *e*) `[inline]`

Invoked when the mouse enters a component.

Parameters

e	

3.18.2.6 void controller.PieceControl.mouseExited (MouseEvent *e*) `[inline]`

Invoked when the mouse exits a component.

Parameters

e	

3.18.2.7 void controller.PieceControl.mousePressed (MouseEvent *e*) `[inline]`

Invoked when a mouse button has been pressed on a component.

Parameters

e	

3.18.2.8 void controller.PieceControl.mouseReleased (MouseEvent *e*) `[inline]`

Invoked when a mouse button has been released on a component.

Parameters

e	

3.18.2.9 void controller.PieceControl.unhighLightAvaiPosi ( ) `[inline]`

unhigh light current available postion on the chess board

The documentation for this class was generated from the following file:

- src/main/java/controller/PieceControl.java

3.19 model.PieceTest Class Reference

Inheritance diagram for model.PieceTest:

`classmodel_1_1PieceTest-eps-converted-to.pdf`

Public Member Functions

- void **testIsValidMove** () throws Exception
- void **testIsBoxPossible** () throws Exception
- void **testIsLinearPossible** () throws Exception
- void **testIsDiagnalPossible** () throws Exception

The documentation for this class was generated from the following file:

- src/test/java/model/PieceTest.java

3.20 view.PieceView Class Reference

Inheritance diagram for view.PieceView:

`classview_1_1PieceView-eps-converted-to.pdf`

Public Member Functions

- void addPieceIcon (Piece piece)
- void highLightCurr ()
- void unhighLightCurr ()

3.20.1 Member Function Documentation

3.20.1.1 void view.PieceView.addPieceIcon (Piece *piece*) `[inline]`

add Piece Image from fileSystem

Parameters

piece	

3.20.1.2 void view.PieceView.highLightCurr ( ) `[inline]`

high light current button

3.20.1.3 void view.PieceView.unhighLightCurr ( ) `[inline]`

unhigh light current button

The documentation for this class was generated from the following file:

- src/main/java/view/PieceView.java

3.21 utility.PrintFormat Class Reference

Public Member Functions

- void printAvalPosi (Piece myPiece, ChessBoard currBoard)

Static Public Member Functions

- static void **printBoard** (ChessBoard myChessBoard)

3.21.1 Detailed Description

print the current ChessBoard

3.21.2 Member Function Documentation

3.21.2.1 void utility.PrintFormat.printAvalPosi (Piece *myPiece,* ChessBoard *currBoard*) `[inline]`

print the chessboard with all available positions of current model by star("$*$")

Parameters

myPiece	
currBoard	

The documentation for this class was generated from the following file:

- src/main/java/utility/PrintFormat.java

3.22 model.Queen Class Reference

Inheritance diagram for model.Queen:

classmodel_1_1Queen-eps-converted-to.pdf

Public Member Functions

- Queen (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Queen (Piece otherPiece)

- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()

Additional Inherited Members

3.22.1 Constructor & Destructor Documentation

3.22.1.1 model.Queen.Queen (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of Queen, inherited from abstract Piece constructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.22.1.2 model.Queen.Queen (Piece *otherPiece*) `[inline]`

copy constructor for Queen

Parameters

otherPiece	

3.22.2 Member Function Documentation

3.22.2.1 String model.Queen.getFullName ( ) `[inline]`, `[virtual]`

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.22.2.2 boolean model.Queen.isValidMove (Piece *myPiece,* ChessBoard *myChessBoard,* int *destX,* int *destY,* boolean *conCheck*) `[inline]`, `[virtual]`

determine whether the current move is valid by myPiece Queen

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/Queen.java

3.23 model.Rook Class Reference

Inheritance diagram for model.Rook:

```
classmodel_1_1Rook-eps-converted-to.pdf
```

Public Member Functions

- Rook (Color pieceColor, int xPos, int yPos, Boolean isFirst, char type)
- Rook (Piece otherPiece)
- boolean isValidMove (Piece myPiece, ChessBoard myChessBoard, int destX, int destY, boolean conCheck)
- String getFullName ()

Additional Inherited Members

3.23.1 Constructor & Destructor Documentation

3.23.1.1 model.Rook.Rook (Color *pieceColor,* int *xPos,* int *yPos,* Boolean *isFirst,* char *type*) `[inline]`

constructor of Rook, inherited from abstract Piece constructor

Parameters

pieceColor	
xPos	
yPos	
isFirst	
type	

3.23.1.2 model.Rook.Rook (Piece *otherPiece*) `[inline]`

copy constructor for Rook

Parameters

otherPiece	

3.23.2 Member Function Documentation

3.23.2.1 String model.Rook.getFullName ( ) `[inline]`, `[virtual]`

Returns

name of the piece eg: black_bishop

Implements model.Piece.

3.23.2.2 boolean model.Rook.isValidMove (Piece *myPiece*, ChessBoard *myChessBoard*, int *destX*, int *destY*, boolean *conCheck*) `[inline]`, `[virtual]`

determine whether the current move is valid by myPiece Rook

Parameters

myPiece	
myChessBoard	
destX	
destY	
conCheck	

Returns

Implements model.Piece.

The documentation for this class was generated from the following file:

- src/main/java/model/Rook.java

3.24 utility.Status Enum Reference

Public Attributes

- **SUCCESS**
- **FAIL**
- **DEFAULT**
- **VALID_MOVE**
- **LOST**
- **DRAW**
- **NO_SUCH_PIECE**
- **NOT_YOUR_PIECE**
- **INVALID_DESTINATION**
- **RANDOM_ERROR**

The documentation for this enum was generated from the following file:

- src/main/java/utility/Status.java