Packages

Files

board

Board

BoardUtils

Tile → EmptyTile

Tile → OccupiedTile

Move → MajorMove

Move → AttackMove

pieces

Piece → Rook

Piece → Knight

Piece → Bishop

Piece → Queen

Piece → King

Piece → Pawn

player

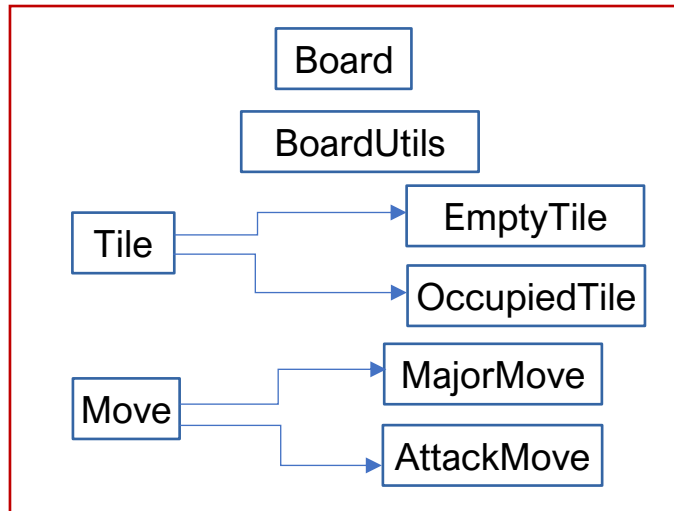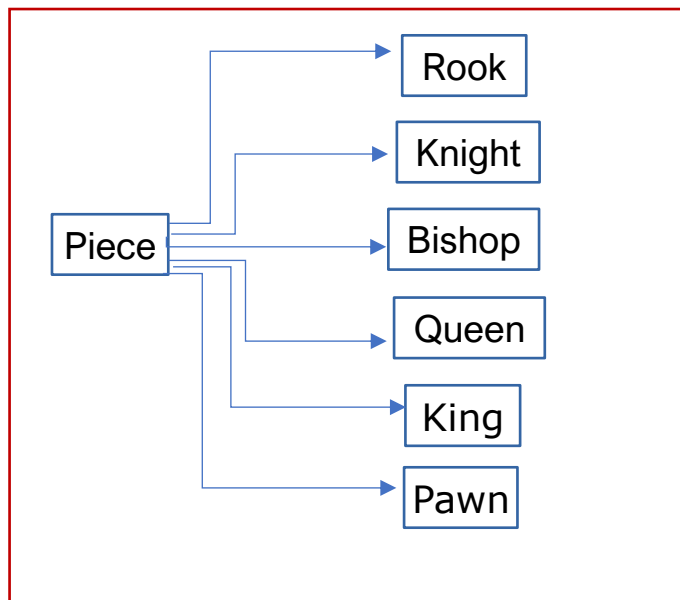Player → WhitePlayer

Player → BlackPlayer

MoveStatus

MoveTransition

gui

Table
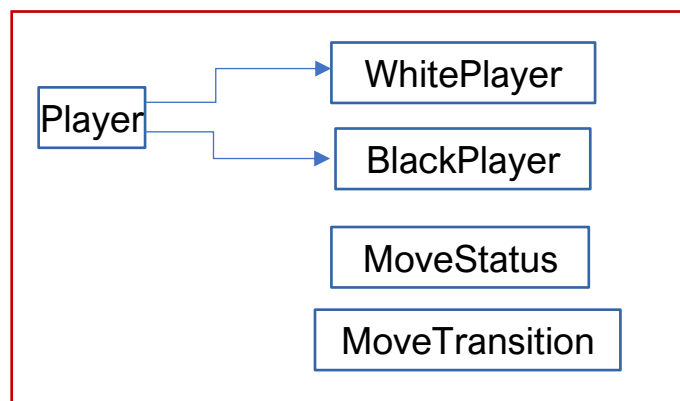
Outside packages files

Alliance

JChess

High Level View of the game

Class Board - It is a public class and provides the functionality to store the information about the game. It stores the tiles, whitePieces, blackPieces, whitePlayer, blackPlayer and their corresponding legal moves. It also stores the currentPlayer who has to make the next move which can be either the whitePlayer or the blackPlayer. It has a static function createStandardBoard which gets called first (initialization begins here) and creates a builder with the help of which all pieces are initialized and stored during the start of the game. It has getters method for all the data members and extra methods like getLegalMoves which return concatenation of all possible moves of white and black player and createGameBoard method which creates all the 64 tiles initializing them with the help of builder and returns this collection to the gameBoard variable.

Class Builder - This class is declared as static inside the Board class. As the name suggests it is the first class to fire up during the launch of game and sets up all the pieces. These pieces are stored using hashmap in the boardConfig variable with each piece mapped to its position in the board. This class also holds the alliance of the player who has to move next.

Class BoardUtils - This class holds utilities in the form of Booleans of size 64 which store true for the indices which lie in that particular row or column.

Class Tile - It is an abstract class providing the functionality to store the piece at a given tile or storing null if no piece is present. This class is extended by OccupiedTile and EmptyTile class. At any point, there are 64 instances of this class stored in the gameBoard variable of class Board. This class contains a static method createTile which creates a tile based on the parameters passed and returns this newly created tile.

Class Move – It is an abstract class which has MajorMove and AttackMove as its subclasses. Move class holds the state of the board i.e. object of Board class, the piece to be moved and the destination coordinate. MajorMove is the normal move whereas AttackMove additionally holds the piece which is attacked in the move. It also contains an execute method which is invoked when a move is to be made. This method creates a new builder which initializes the new board after the move has been made and this board is returned. Another method it has is the createMove method which on being given board, currentCoordinate and destinationCoordinate of the current move checks if this move exists, if it does, the matched move is returned otherwise a NULL_MOVE is returned. It does so by finding a match of the given move with every single move of all the available pieces on the board with the help of getLegalMoves method of the Board class. Another important method in this class is the execute method which gets called by an object of this class and it does most job. Execute method sets up the builder along with the pieces in their new positions for the move to be made and returns a new board.

Class Piece - It is an abstract class and provides the functionality of storing the respective piece positions of each of the pieces along with their alliance and their piece type. This class is extended by King, Queen, Bishop, Rook, Knight and Pawn classes. It has getters for each of its data members. Each class having their own MOVE_COORDINATES and own implementation of the method calculateLegalMoves which calculates their legal moves in the current state of the board and returns this list of moves. Each of the subclasses also implement a movePiece method which returns a new piece having new piece position from the move which is passed.

Class Table – This is the class is where things get linked up. It connects the gui with the logic. In this class, chessboard gets initialized along with building up of gui. The gui essentially contains a frame inside which there are two panels, BoardPanel and the ScorePanel. BoardPanel is a JPanel class consisting of 64 different panels as tiles arranged in a grid layout of 8X8. TilePanel class extends JPanel with each panel having assigned a tile color and a piece icon depending on the current state of the board. Each tile has a mouse listener connected to it. When a tile with a piece on it is clicked, all the tiles which can act as destination

coordinate for the selected piece get highlighted. When the piece is moved to any of these highlighted tiles, the gui board gets changed to a new board with this new configuration. This is done using drawBoard function of BoardPanel class.

ScorePanel contains three different sections with two containing the dismissed pieces of white and black player and the other providing the functionality of a timer which gets reset after every move. This section also displays info such as check state, illegal move and checkmate. Timer from javax.swing is used to which an action listener is connected which invokes itself after every second changing the text of timeLabel.

Class Player - It is an abstract class holding the current state of the player such as playerKing piece, legalMoves for all the pieces, the state of the board and Boolean isInCheck.

Method makeMove is defined in this class which receives move as argument and creates a transitionBoard and ensures that after the current move that the king of the player which made the move is not in danger. It returns object of MoveTransition class having a status depending on the validity of move. This MoveStatus determines whether to change the original board or not. WhitePlayer and BlackPlayer are subclasses of this class.

What happens when a player clicks on a destination tile?