```java

import Game.*;
import Pieces.*;
import java.util.Arrays;
import java.util.Scanner;
import java.util.regex.Pattern;

/**
 *
 * @author E
 */
public class Main {

    public Main() {
        Scanner sc;
        String[] columnSplit, rowSplit;
        String columnsA, columnsB, rowsA, rowsB;
        int cStart, rStart;
        int startR, startC, nextR, nextC;
        int searchDepth = 1; // need user input to change
        Colour player, opponent;
        sc = new Scanner(System.in);
        // Get start values for game
        while (true) {
            System.out.print("Input Player Colour (w/b): ");
            String playerColour = sc.next();
            if (playerColour.equalsIgnoreCase("w")
                    || playerColour.equalsIgnoreCase("white")) {
                player = Colour.White;
                opponent = Colour.Black;
                break;
            } else if (playerColour.equalsIgnoreCase("b")
                    || playerColour.equalsIgnoreCase("black")) {
                player = Colour.Black;
                opponent = Colour.White;
                break;
            } else {
                System.out.println("Invalid Input");
            }
        }

        // Get the depth
        while (true) {
            System.out.print("Input Search Depth (Minimum 1): ");
            String depth = sc.next();
            int depthVal = 0;
            try{
                depthVal = Integer.parseInt(depth);
            }catch(Exception ex){
                System.out.println("Invalid Depth");
            }
            if(depthVal > 0){
                searchDepth = depthVal;
                break;
            }
            else{
                searchDepth = 1;
                break;
            }
        }
        // create game
        Game game = new Game();

        game.getBoard().printBoard();
        while (!game.isGameEnd()) {
            // could have ai determine next move first on a separate thread
            // get player input
            if (game.getCurrentTurn() == player) {
                System.out.println("Player Making Move");
                // Get input from the user
                while (true) {
                    System.out.print("Input Position: ");
                    String userInput = sc.next();
                    /*
                    String columns = userInput.replaceAll("[^a-g]", "");
                    String rows = userInput.replaceAll("[^1-8]", "");
                     */
                    columnSplit = userInput.split("[^a-z]+");
                    rowSplit = userInput.split("[^0-9]+");
                    cStart = columnSplit[0].equals("") ? 1 : 0;
                    rStart = rowSplit[0].equals("") ? 1 : 0;

                    try {
                        if (columnSplit.length == 1) {
                            columnsA = columnSplit[cStart].replaceAll("[^a-h]", "");
                            rowsA = rowSplit[rStart].replaceAll("[^1-8]", "");

                            startC = Board.boardToIndexC(columnsA.charAt(0));
                            startR = Board.boardToIndexR(Character.getNumericValue(rowsA.charAt(0)));
                            nextC = Board.boardToIndexC(columnsA.charAt(1));
                            nextR = Board.boardToIndexR(Character.getNumericValue(rowsA.charAt(1)));
                            System.out.println("Start Position: " + columnsA.charAt(0) + "" + rowsA.charAt(0));
                            System.out.println("Next Position: " + columnsA.charAt(1) + "" + rowsA.charAt(1));
                        } else {
                            columnsA = columnSplit[cStart].replaceAll("[^a-h]", "");
                            columnsB = columnSplit[cStart + 1].replaceAll("[^a-h]", "");
                            rowsA = rowSplit[rStart].replaceAll("[^1-8]", "");
                            rowsB = rowSplit[rStart + 1].replaceAll("[^1-8]", "");

                            startC = Board.boardToIndexC(columnsA.charAt(0));
                            startR = Board.boardToIndexR(Character.getNumericValue(rowsA.charAt(0)));
                            nextC = Board.boardToIndexC(columnsB.charAt(0));
                            nextR = Board.boardToIndexR(Character.getNumericValue(rowsB.charAt(0)));
                            System.out.println("Start Position: " + columnsA.charAt(0) + "" + rowsA.charAt(0));
                            System.out.println("Next Position: " + columnsB.charAt(0) + "" + rowsB.charAt(0));
                        }
                        // Set the next board and change turn, if move is valid
                        Board next = game.nextBoard(startR, startC, nextR, nextC);
                        if (!next.equals(game.getBoard())) {
                            System.out.println("Changing Turn");
                            game.setBoard(next);
                            game.changeTurn();
                        }
                        game.getBoard().printBoard();
                        break;
                    } catch (Exception e) {
                        System.out.println("Invalid Input");
                    }
                }
            } else {
                System.out.println("AI Making Move");
                // game.setBoard(game.getBoard());
                // GAME TREE ALGORITHM
                // ai determines next move to perform (may be slow)
                GameTree gameTree = new GameTree(game, searchDepth);
                Node bestMove = gameTree.findBestMove(opponent, opponent,
                    gameTree.root, null);
                    System.out.println("Move: " + bestMove.move.getMove());

                    Board next = game.nextBoard(bestMove.move);
                    if (!next.equals(game.getBoard())) {
                        game.setBoard(next);
                        game.changeTurn();
                    }
                    game.getBoard().printBoard();
                }
            }

        }

    public static void main(String args[]) {
        Main m = new Main();
    }
}
```