//For testing purposes, makes the program infinite and completely disregards black

import java.util.Scanner;

import static java.lang.System.\*;

public class Multiplayer

{

private static String name1 = "";

private static String name2 = "";

private static boolean playerTurn = true;

private static boolean win = false;

private static Scanner keyboard = new Scanner(System.in);

private static String player1 = "";

private static String player2 = "";

public static void start()

{

int j = 0;

boolean playerOneWin = true;

boolean playerTwoWin = true;

out.print('\u000c');

out.print("Enter the name of player 1 --> ");

name1 = keyboard.nextLine();

out.print("Enter the name of player 2 --> ");

name2 = keyboard.nextLine();

while(j != 1) // Start of code by Ayaan Nazir

{

out.print("\n" + name1 + ": Black? (b) or White? (w) : ");

String color = keyboard.nextLine();

out.print('\u000c');

switch(color)

{

case "w" : out.println("Great! You start first\n\n"); player1 = name1; player2 = name2; j++; break;

case "W" : out.println("Great! You start first\n\n"); player1 = name1; player2 = name2; j++; break;

case "b" : out.println("Great! You start second\n\n");player2 = name1; player1 = name2; j++; break;

case "B" : out.println("Great! You start second\n\n");player2 = name1; player1 = name2; j++; break;

default : out.println("Invalid color. Please enter either black (b) or white (w)\n"); break;

}

}

Board.create(player1, player2); // End of code by Ayaan Nazir

while(playerOneWin && playerTwoWin)

{

callMove();

Board.update();

}

if(playerOneWin || playerTwoWin)

{

out.println("###########\n#Checkmate#\n###########");

if(playerOneWin)

out.println("Congratulations " + name1 + "! You have successfully checkmated " + name2 + "!");

else

out.println("Congratulations " + name2 + "! You have successfully checkmated " + name1 + "!");

}

else

{

out.println("###########\n#Stalemate#\n###########");

if(true)//not defined yet, but represents which player stalemated the other

out.println("Congratulations " + name1 + "! You have successfully outsmarted " + name2 + " and forced a stalemate!");

else

out.println("Congratulations " + name2 + "! You have successfully outsmarted " + name1 + " and forced a stalemate!");

}

}

public static void callMove()//Start of Evan

{

String piece = "";

String col = "";

String row = "";

int colm = 0;

int rowm = 0;

int colm1 = 0;

int rowm1 = 0;

if(playerTurn)//Checks to see if it is player one’s turn

{

out.print("\n\n" + player1 + " select your piece → "); //Start of Ayaan

int f = 0;

String input = "";

while(f!=1)

{

input = keyboard.nextLine();

if(checkValidity(input,3))

{

colm = (interpret(input,3)[0]);//equal to 2

rowm = (interpret(input,3)[1]);//equal to 6

piece = input.substring(0,1);

if(!(Board.board1[rowm][colm].getPieceType().equals("#") || Board.board1[rowm][colm].getPieceType().equals("/")))

{

out.print("\n\n" + player1 + " select where you would like the piece to go → "); //Start of Ayaan

int l = 0;

while(l!=1)

{

input = keyboard.nextLine();

if(checkValidity(input,2))

{

colm1 = (interpret(input,2)[0]);//equal to 2

rowm1 = (interpret(input,2)[1]);//equal to 4

l++;

}

else

out.println("Invalid syntax. Please input the new location as follows: \"letter\"\"number\"");

}

switch(piece)

{

case "p": ((Pawn)Board.board1[rowm][colm]).movePiece(rowm1,colm1,"w");

//case "k": ((King)array[rowm][colm]).movePiece(rowm1,colm1,"w");

//case "n": ((Knight)array[rowm][colm]).movePiece(rowm1,colm1,"w");

//case "b": ((Bishop)array[rowm][colm]).movePiece(rowm1,colm1,"w");

//case "q": ((Queen)array[rowm][colm]).movePiece(rowm1,colm1,"w");

//case "r": ((Rook)array[rowm][colm]).movePiece(rowm1,colm1,"w");

}

f++;

}

else

out.println("Invalid location. Please input a location that contains a piece");

}

else

out.println("Invalid syntax. Please input the first letter of the piece followed by the location: \"letter\"\"number\" without spaces");

}

}

else

{

playerTurn = true;//switches to player 1

out.print("\n\n" + player2 + " select your piece → "); //Start of Ayaan

int f = 0;

String input = "";

while(f!=1)

{

input = keyboard.nextLine();

if(checkValidity(input,3))

{

colm = (interpret(input,3)[0]);//equal to 2

rowm = (interpret(input,3)[1]);//equal to 6

piece = input.substring(0,1);

if(!(Board.board1[rowm][colm].getPieceType().equals("#") || Board.board1[rowm][colm].getPieceType().equals("/")))

{

out.print("\n\n" + player2 + " select where you would like the piece to go → "); //Start of Ayaan

int l = 0;

while(l!=1)

{

input = keyboard.nextLine();

if(checkValidity(input,2))

{

colm1 = (interpret(input,2)[0]);//equal to 2

rowm1 = (interpret(input,2)[1]);//equal to 4

l++;

}

else

out.println("Invalid syntax. Please input the new location as follows: \"letter\"\"number\"");

}

switch(piece)

{

case "p": ((Pawn)Board.board1[rowm][colm]).movePiece(rowm1,colm1,"b");

//case "k": ((King)array[rowm][colm]).movePiece(rowm1,colm1,"b");

//case "n": ((Knight)array[rowm][colm]).movePiece(rowm1,colm1,"b");

//case "b": ((Bishop)array[rowm][colm]).movePiece(rowm1,colm1,"b");

//case "q": ((Queen)array[rowm][colm]).movePiece(rowm1,colm1,"b");

//case "r": ((Rook)array[rowm][colm]).movePiece(rowm1,colm1,"b");

}

f++;

}

else

out.println("Invalid location. Please input a location that contains a piece");

}

else

out.println("Invalid syntax. Please input the first letter of the piece followed by the location: \"letter\"\"number\" without spaces");

}

}

}

public static int[] interpret(String move, int number)

{

String letCord = move.substring(0,(number-1));

if(letCord.length()!=1)

letCord = move.substring(1,number-1);

String numCord = move.substring(number-1);

int[] translated = new int[number];

switch(letCord)

{

case "a": translated[0] = 0;break;

case "b": translated[0] = 1;break;

case "c": translated[0] = 2;break;

case "d": translated[0] = 3;break;

case "e": translated[0] = 4;break;

case "f": translated[0] = 5;break;

case "g": translated[0] = 6;break;

case "h": translated[0] = 7;break;

}

switch(numCord)

{

case "1": translated[1] = 7;break;

case "2": translated[1] = 6;break;

case "3": translated[1] = 5;break;

case "4": translated[1] = 4;break;

case "5": translated[1] = 3;break;

case "6": translated[1] = 2;break;

case "7": translated[1] = 1;break;

case "8": translated[1] = 0;break;

}

return translated;

}

public static boolean checkValidity(String check, int length)

{

if(check.length() == length)

return true;

else

return false;

}

}