public class Hashset {

int x;

int y;

public Hashset(int x, int y) {

this.x=x;

this.y=y;

}

public String toString() {

return "["+x+","+y+"]";

}

}

public class Hashsetandscoring{

int score;

Hashset hashset;

public Hashsetandscoring(int score, Hashset hashset)

{

this.score=score;

this.hashset=hashset;

}

}

import java.util.ArrayList;

import java.util.List;

public class Domainbase {

int NoPlayer=0;

int comp=1;

int user=2;

int [][] domainbase=new int [3][3];

public Hashset cmpmovement;

public boolean GameisOver() {

return victory(comp) || victory(user) || getavilibleCels().isEmpty();

//get avilibleCels optional he....

}

public boolean victory(int player) {

if((domainbase[0][0]==domainbase[1][1] && domainbase[0][0]==domainbase[2][2] && domainbase[0][0]==player) || (domainbase[0][2]==domainbase[1][1] && domainbase[0][2]==domainbase[2][0] && domainbase[0][2]==player )) {

return true;

}

for(int i =0; i<3; i++) {

if((domainbase[i][0]==domainbase[i][1] && domainbase[i][0]==domainbase[i][2] && domainbase[i][0]==player)

|| (domainbase[0][i]==domainbase[1][i] && domainbase[0][i]==domainbase[2][i] && domainbase[0][i]==player)

) {

return true;

}

}

return false;

}

public List<Hashset> getavilibleCels(){

List<Hashset> avilibleCels= new ArrayList<>(); //for available cells

for(int i=0 ; i <3 ; i++) {

for(int j=0; j<3; j++) {

if (domainbase[i][j]==NoPlayer)

avilibleCels.add(new Hashset(i,j)); //other available cells

}

}

return avilibleCels;

}

//specific value for particular cells agr usme phele se he tu false dega else true

public boolean placemove(Hashset hashset, int player) {

if(domainbase[hashset.x][hashset.y]!= NoPlayer) //mtlb isme value he noplayer 0 he to ye iske barabr nahi he

return false;

domainbase[hashset.x][hashset.y] = player; //player move krega

return true;

}

public void displaydomainbase() {

System.out.println();

for(int i=0 ; i<3 ; i++) {

for(int j=0 ; j<3 ; j++) {

String value="-";

if (domainbase[i][j]==comp)

value ="X";

if(domainbase[i][j]==user)

value="O";

System.out.print(value+" ");

}

System.out.println();

}

System.out.println();

}

public int minimax(int dpt, int trn) {

if(victory(comp))

return 1;

if(victory(user))

return -1;

List<Hashset>avilibleCels= getavilibleCels();

if(avilibleCels.isEmpty())

return 0;

int min = Integer.MAX\_VALUE;

int max=Integer.MIN\_VALUE;

for (int i=0 ; i<avilibleCels.size();i++) {

Hashset hashset = avilibleCels.get(i);

if (trn == comp) {

placemove(hashset, comp);

int CurntValue = minimax(dpt + 1, user); //dpt 1 for user player

max=Math.max(CurntValue, max);

if(dpt==0)

System.out.println("Computer score for postion" + hashset+ "=" +CurntValue);

if(CurntValue>=0)

if(dpt==0)

cmpmovement=hashset;

if(CurntValue==1) {

domainbase[hashset.x][hashset.y]= NoPlayer;

break;

}

if(i==avilibleCels.size()-1 && max<0)

if(dpt == 0)

cmpmovement=hashset;

}

else if(trn== user) {

placemove(hashset, user);

int cuurentscore= minimax(dpt+1, comp);

min=Math.min(cuurentscore, min);

if (min==-1) {

domainbase[hashset.x][hashset.y]=NoPlayer;

break;

}

}

domainbase[hashset.x][hashset.y]=NoPlayer;

}

return trn== comp ? max:min; //agr comp ki trn he to max value otherwise min value

}

}

import java.util.Random;

import java.util.Scanner;

public class TicCross {

public static Random random= new Random();

public static void main(String[] args) {

Domainbase b = new Domainbase();

Scanner scanner=new Scanner(System.in);

b.displaydomainbase();

System.out.println("Select trn:\n 1. Computer(X)/ 2. User(O) ");

int choice= scanner.nextInt();

if(choice==b.comp) {

Hashset p = new Hashset(random.nextInt(3),random.nextInt(3));

b.placemove(p, b.comp);

b.displaydomainbase();

}

while(!b.GameisOver()) {

boolean moveok= true;

do {

if(!moveok) { //exception

System.out.println("Cell already filled !");

}

System.out.println("Your Move: ");

Hashset usermove=new Hashset(scanner.nextInt(),scanner.nextInt());

moveok=b.placemove(usermove, b.user);

}

while (!moveok);

b.displaydomainbase();

if(b.GameisOver())

break;

b.minimax(0, b.comp);

System.out.println("computer choose postion: "+ b.cmpmovement);

b.placemove(b.cmpmovement, b.comp);

b.displaydomainbase();

}

if(b.victory(b.comp))

System.out.println("You Lost! ");

else if(b.victory(b.user))

System.out.println("You win! ");

else

System.out.println("Draw!");

}

}