Name: Shravani Bahulekar

TY-CS-D     BATCH 2

ROLL NO: 32

Assignment 1-B: Tic tac toe game with AI approach (using minmax algorithm)

**CODE -**

import java.util.Scanner;  
  
public class TicTacToeMinimax {  
 private static final char *PLAYER* = 'X';  
 private static final char *AI* = 'O';  
 private static final char *EMPTY* = '\_';  
  
 private static boolean isFull(char[][] board) {  
 for (char[] row : board) {  
 for (char cell : row) {  
 if (cell == *EMPTY*) {  
 return false;  
 }  
 }  
 }  
 return true;  
 }  
  
 private static boolean checkWinner(char[][] board, char player) {  
 for (int i = 0; i < 3; i++) {  
 if (board[i][0] == player && board[i][1] == player && board[i][2] == player) {  
 return true; // Row check  
 }  
 if (board[0][i] == player && board[1][i] == player && board[2][i] == player) {  
 return true; // Column check  
 }  
 }  
 if (board[0][0] == player && board[1][1] == player && board[2][2] == player) {  
 return true; // Diagonal from top-left to bottom-right  
 }  
 if (board[0][2] == player && board[1][1] == player && board[2][0] == player) {  
 return true; // Diagonal from top-right to bottom-left  
 }  
 return false;  
 }  
  
 private static void printBoard(char[][] board) {  
 for (char[] row : board) {  
 for (char cell : row) {  
 System.*out*.print(cell + " ");  
 }  
 System.*out*.println();  
 }  
 }  
  
 /\*private static int evaluate(char[][] board) {  
 if (checkWinner(board, AI)) {  
 return 10;  
 }  
 if (checkWinner(board, PLAYER)) {  
 return -10;  
 }  
 return 0;  
 }\*/  
  
 private static int minimax(char[][] board, int depth, boolean maximizingPlayer) {  
 if (*checkWinner*(board, *AI*)) {  
 return 10 - depth;  
 }  
 if (*checkWinner*(board, *PLAYER*)) {  
 return depth - 10;  
 }  
 if (*isFull*(board)) {  
 return 0;  
 }  
  
 if (maximizingPlayer) {  
 int maxEval = Integer.*MIN\_VALUE*;  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 if (board[i][j] == *EMPTY*) {  
 board[i][j] = *AI*;  
 int eval = *minimax*(board, depth + 1, false);  
 board[i][j] = *EMPTY*;  
 maxEval = Math.*max*(maxEval, eval);  
 }  
 }  
 }  
 return maxEval;  
 } else {  
 int minEval = Integer.*MAX\_VALUE*;  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 if (board[i][j] == *EMPTY*) {  
 board[i][j] = *PLAYER*;  
 int eval = *minimax*(board, depth + 1, true);  
 board[i][j] = *EMPTY*;  
 minEval = Math.*min*(minEval, eval);  
 }  
 }  
 }  
 return minEval;  
 }  
 }  
  
 private static int[] findBestMove(char[][] board) {  
 int[] bestMove = {-1, -1};  
 int bestEval = Integer.*MIN\_VALUE*;  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 if (board[i][j] == *EMPTY*) {  
 board[i][j] = *AI*;  
 int eval = *minimax*(board, 0, false);  
 board[i][j] = *EMPTY*;  
 if (eval > bestEval) {  
 bestEval = eval;  
 bestMove[0] = i;  
 bestMove[1] = j;  
 }  
 }  
 }  
 }  
 return bestMove;  
 }  
  
 public static void main(String[] args) {  
 char[][] board = new char[3][3];  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 board[i][j] = *EMPTY*;  
 }  
 }  
  
 Scanner scanner = new Scanner(System.*in*);  
 while (!*isFull*(board) && !*checkWinner*(board, *PLAYER*) && !*checkWinner*(board, *AI*)) {  
 *printBoard*(board);  
  
 // Player's move  
 System.*out*.print("Enter your move (row and column, separated by space): ");  
 int row = scanner.nextInt();  
 int col = scanner.nextInt();  
 while (board[row][col] != *EMPTY*) {  
 System.*out*.println("Invalid move. Cell is already occupied.");  
 System.*out*.print("Enter your move (row and column, separated by space): ");  
 row = scanner.nextInt();  
 col = scanner.nextInt();  
 }  
 board[row][col] = *PLAYER*;  
  
 // AI's move  
 if (!*isFull*(board) && !*checkWinner*(board, *AI*)) {  
 int[] aiMove = *findBestMove*(board);  
 board[aiMove[0]][aiMove[1]] = *AI*;  
 }  
 }  
  
 *printBoard*(board);  
  
 if (*checkWinner*(board, *PLAYER*)) {  
 System.*out*.println("Congratulations! You won!");  
 } else if (*checkWinner*(board, *AI*)) {  
 System.*out*.println("AI wins!");  
 } else {  
 System.*out*.println("It's a draw!");  
 }  
  
 scanner.close();  
 }  
}