**Name: Sujal Samir Bokariya      TY-CS-D     BATCH 2      ROLL NO :45**

**Assignment 1-B :Tic tac toe game with AI approach**

**Code-**

package AI\_Lab;

import java.util.Scanner;

public class TicTacToeAi {

public static void displayBoard(char[][] board) {

System.***out***.println("\n-------------");

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

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

System.***out***.print("| " + board[i][j] + " ");

}

System.***out***.println("|");

System.***out***.println("-------------");

}

}

public 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;

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

return true;

}

if (board[0][0] == player && board[1][1] == player && board[2][2] == player)

return true;

if (board[0][2] == player && board[1][1] == player && board[2][0] == player)

return true;

return false;

}

public static boolean isBoardFull(char[][] board) {

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

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

if (board[i][j] == '-')

return false;

}

}

return true;

}

public static int minimax(char[][] board, boolean isMaximizing, char player) {

if (*checkWinner*(board, 'X'))

return -1;

if (*checkWinner*(board, 'O'))

return 1;

if (*isBoardFull*(board))

return 0;

if (isMaximizing) {

int maxEval = Integer.***MIN\_VALUE***;

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

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

if (board[i][j] == '-') {

board[i][j] = player;

int eval = *minimax*(board, false, player);

board[i][j] = '-';

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] == '-') {

board[i][j] = (player == 'X') ? 'O' : 'X';

int eval = *minimax*(board, true, player);

board[i][j] = '-';

minEval = Math.*min*(minEval, eval);

}

}

}

return minEval;

}

}

public static int[] findBestMove(char[][] board, char player) {

int bestVal = Integer.***MIN\_VALUE***;

int[] bestMove = {-1, -1};

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

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

if (board[i][j] == '-') {

board[i][j] = player;

int moveVal = *minimax*(board, false, player);

board[i][j] = '-';

if (moveVal > bestVal) {

bestVal = moveVal;

bestMove[0] = i;

bestMove[1] = j;

}

}

}

}

return bestMove;

}

public static void main(String[] args) {

char[][] board = new char[3][3];

char[][] board1 = new char[3][3];

int row, col, pos;

char player = 'X';

char ok;

int n= 1;

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

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

board[i][j] = '-';

board1[i][j] = (char) (n + '0');

n++;

}

}

Scanner scanner = new Scanner(System.***in***);

System.***out***.println("Welcome to Tic-Tac-Toe!\n");

System.***out***.println("\*\*\*\*\*\*\*Remember the positions\*\*\*\*\*\*");

*displayBoard*(board1);

System.***out***.print("Start?: ");

ok = scanner.next().charAt(0);

if (ok == 'Y' || ok == 'y') {

*displayBoard*(board);

while (true) {

System.***out***.print("\nPlayer " + player + "'s turn: ");

if (player == 'X') {

pos = scanner.nextInt();

if (pos > 9 || pos < 1) {

System.***out***.println("Invalid move. Try again.");

continue;

}

row = (pos - 1) / 3;

col = (pos - 1) % 3;

if (board[row][col] != '-') {

System.***out***.println("Invalid move. Try again.");

continue;

}

} else {

int[] aiMove = *findBestMove*(board, player);

row = aiMove[0];

col = aiMove[1];

pos = row \* 3 + col + 1;

}

row = (pos - 1) / 3;

col = (pos - 1) % 3;

board[row][col] = player;

*displayBoard*(board);

if (*checkWinner*(board, player)) {

System.***out***.println("Player " + player + " wins!");

break;

}

if (*isBoardFull*(board)) {

System.***out***.println("It's a tie!");

break;

}

player = (player == 'X') ? 'O' : 'X';

}

} else {

scanner.close();

return;

}

scanner.close();

}

}