**Name – Sakshee Agrawal TY-CS-D Batch-1 Roll no -07**

*---------------------------------------------------------------------------*

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

**Code –**

#include <iostream>

using namespace std;

char board[3][3] = {{' ',' ',' '},{' ',' ',' '},{' ',' ',' '}};

char currentPlayer = 'O';

char choice;

int row,column;

void display\_board(){

cout<<"\n\t------------------------------------";

cout<<"\n\t\t Human[X]\t AI[O]\n";

cout<<"\t------------------------------------\n\n";

cout<<"\t\t | | \n";

cout<<"\t\t "<<board[0][0]<<" | "<<board[0][1]<<" | "<<board[0][2]<<" \n";

cout<<"\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n";

cout<<"\t\t | | \n";

cout<<"\t\t "<<board[1][0]<<" | "<<board[1][1]<<" | "<<board[1][2]<<" \n";

cout<<"\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n";

cout<<"\t\t | | \n";

cout<<"\t\t "<<board[2][0]<<" | "<<board[2][1]<<" | "<<board[2][2]<<" \n";

cout<<"\t\t | | \n";

}

bool checkWin(char letter) {

// Check for the win in rows and columns

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

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

(board[0][i] == letter && board[1][i] == letter && board[2][i] == letter)) {

return true;

}

}

// Check for the win in diagonal

if ((board[0][0] == letter && board[1][1] == letter && board[2][2] == letter) ||

(board[0][2] == letter && board[1][1] == letter && board[2][0] == letter)) {

return true;

}

return false;

}

bool checkDraw() {

//checking if any of them has won

if(checkWin('X')||checkWin('O')) return false;

//checking if moves are left as none of them have won

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

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

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

return false;

}

}

}

return true;

}

int heuristic(char b[3][3]){

// if AI wins return 10

if(checkWin('O')) return 10;

// if Human wins return -10

if (checkWin('X')) return -10;

// Else if none of them have won then return 0

return 0;

}

int minmax(char board[3][3], int depth, bool isMax){

int score = heuristic(board);

if (score == 10 || score == -10)

return score;

if (checkDraw())

return 0;

// This is maximizer's (AI's) move

if (isMax)

{

int best = -1000;

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

{

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

{

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

{

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

best = max( best,

minmax(board, depth+1, false));

board[i][j] = ' ';

}

}

}

return best;

}

// This is minimizer's (Human's) move

else

{

int best = 1000;

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

{

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

{

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

{

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

best = min(best,

minmax(board, depth+1, true));

board[i][j] = ' ';

}

}

}

return best;

}

}

void computerMove(){

int best = -1000;

int row = -1;

int col = -1;

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

{

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

{

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

{

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

int moveVal = minmax(board, 0,false);

board[i][j] = ' ';

if (moveVal >best)

{

row = i;

col = j;

best = moveVal;

}

}

}

}

board[row][col] = 'O';

display\_board();

}

void playerMove(){

cout << "\n\n\t Place your move [X] : ";

cin >> choice;

switch (choice) {

case '1': row = 0; column = 0; break;

case '2': row = 0; column = 1; break;

case '3': row = 0; column = 2; break;

case '4': row = 1; column = 0; break;

case '5': row = 1; column = 1; break;

case '6': row = 1; column = 2; break;

case '7': row = 2; column = 0; break;

case '8': row = 2; column = 1; break;

case '9': row = 2; column = 2; break;

default:

cout << "Invalid Move. Please choose a number between 1 and 9.\n";

playerMove();

return;

}

if (board[row][column] != 'X' && board[row][column] != 'O') {

board[row][column] = 'X';

} else {

cout << "That cell has been already filled! Please choose another!!\n\n";

playerMove();

return;

}

display\_board();

}

int main() {

cout << "\t\t\t\t W E L C O M E T O ";

cout << "\n\t\t\t I C -- T A C -- T O E -- G A M E\t\t\t";

cout << "\n\t\t\t\t HUMAN(X) vs AI(O)\n\t\t\t";

display\_board();

cout << "\n\t\tAI plays first\n\n";

while (true) {

if (currentPlayer == 'O') {

computerMove();

if (checkWin('O')) {

cout << "AI wins!" << endl;

break;

} else if (checkDraw()) {

cout << "Draw!" << endl;

break;

}

} else {

playerMove();

if (checkWin('X')) {

cout << "Player wins!" << endl;

break;

} else if (checkDraw()) {

cout << "Draw!" << endl;

break;

}

}

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

}

return 0;

}