**Name**: Suyash Shinde

**Class**: TY -CS **Div**: D

**Roll no**:28

**Tic-Tac-Toe AI approach**

**Code:**

#include <iostream>

using namespace std;

#define COMPUTER 1

#define HUMAN 2

#define COMPUTERMOVE 'O'

#define HUMANMOVE 'X'

void initialize(char board[3][3]){

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

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

board[i][j]='\*';

}

}

}

void showBoard(char board[3][3]){

cout<<board[0][0]<<" | "<< board[0][1]<<" | "<<board[0][2]<<endl;

cout<<"------------------------"<<endl;

cout<<board[1][0]<<" | "<< board[1][1]<<" | "<<board[1][2]<<endl;

cout<<"------------------------"<<endl;

cout<<board[2][0]<<" | "<< board[2][1]<<" | "<<board[2][2]<<endl;

}

bool rowCrossed(char board[3][3]){

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

if(board[i][0]==board[i][1]&&board[i][1]==board[i][2]&&board[i][0]!='\*')

return (true);

}

return false;

}

bool columnCrossed(char board[3][3]){for(int i=0;i<3;i++){

if(board[0][i]==board[1][i] &&board[1][i]==board[2][i] &&board[0][i]!='\*')

return true;

}

return false;

}

bool diagonalCrossed(char board[3][3]){

if(board[0][0]==board[1][1]&&board[1][1]==board[2][2]&&board[0][0]!='\*')

return true;

if(board[0][2]==board[1][1]&&board[1][1]==board[2][0]&&board[0][2]!='\*')

return true;

return false;

}

void declareWinner(int whoseTurn)

{

if (whoseTurn == COMPUTER)

printf("COMPUTER has won\n");

else

printf("HUMAN has won\n");

}

bool gameOver(char board[3][3]){

return (rowCrossed(board) || columnCrossed(board) ||diagonalCrossed(board));

}

int minimax(char board[3][3],int depth,bool isAI){

int score=0;

int bestScore=0;

if(gameOver(board)==true){

if(isAI==true)

return -10;

if(isAI==false)

return +10;

}else

{

if(depth<9){

if(isAI==true){

bestScore=-999;

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

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

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

board[i][j]=COMPUTERMOVE;

score=minimax(board,depth+1,false);

board[i][j]='\*';

if(score>bestScore){

bestScore=score;

}

}

}

}

return bestScore;

}else

{

bestScore=999;

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

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

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

board[i][j]=HUMANMOVE;

score=minimax(board,depth+1,true);

board[i][j]='\*';

if(score<bestScore){

bestScore=score;

}

}

}

}

return bestScore;

}

}

else{

return 0;

}

}

return 0;

}

int bestMove(char board[3][3],int moveIndex){

int x=-1;

int y=-1;

int score=0,bestScore=-999;

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

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

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

board[i][j]=COMPUTERMOVE;

score=minimax(board,moveIndex+1,false);

board[i][j]='\*';

if(score>bestScore){

bestScore=score;

x=i;

y=j;

}

}

}

}

return x\*3+y;

}

void playTicTacToe(int turn){

char board[3][3];

int moveIndex=0,x=0,y=0;

initialize(board);

while(gameOver(board)==false && moveIndex!=9){

int n;

if(turn==COMPUTER){

n=bestMove(board,moveIndex);

x=n/3;

y=n%3;

board[x][y]=COMPUTERMOVE;

cout<<"Computer has put a "<<COMPUTERMOVE<<" in cell"<<n+1<<endl;

showBoard(board);

moveIndex++;

turn=HUMAN;

}else if(turn=HUMAN)

{

cout<<"You can insert at the following postion "<<endl;

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

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

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

cout<<(i\*3+j)+1<<" ";

}

}

}

cout<<endl<<"Enter the postion"<<endl;

cin>>n;

n--;

x=n/3;

y=n%3;

if(board[x][y]=='\*' && n<9 && n>=0){

board[x][y]=HUMANMOVE;

cout<<endl<<"Human has put a "<<HUMANMOVE<<" in cell"<<n+1<<endl;

showBoard(board);

moveIndex++;

turn=COMPUTER;

}else if(board[x][y]!='\*' && n<9 &&n>=0)

{

cout<<endl<<"Postion is already occupied try anotherone";

}else if(n<0 || n>8)

{

cout<<endl<<"Invalid postion";

}

}

}

if (gameOver(board) == false && moveIndex == 3 \*3)

cout<<"It's a draw\n";

else

{

if (turn == COMPUTER)

turn = HUMAN;

else if (turn == HUMAN)

turn = COMPUTER;

declareWinner(turn);

}

}

int main(){

cout<<"Welcome to Tictactoe game"<<endl;

char choice;

cout<<"Are you interested to play first move(y/n)"<<endl;

cin>>choice;

if(choice=='y'){

playTicTacToe(HUMAN);

}else if(choice=='n')

{

playTicTacToe(COMPUTER);

}else

{

cout<<"Please select a valid choice";

}

}

**Output**:





