**Project-2 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**ABHI RAJ**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

char marks[3][3];

int counter = 0;

void initialize\_rdm(void);

void initialize\_grid(void);

void grid(void);

int checker(char c, int player);

void result(void);

void retry(void);

void ai\_opponent(int n);

void singleplayer(int n);

void multiplayer(void);

int main(void){

int op, x;

initialize\_rdm();

initialize\_grid();

grid();

printf("\tGame Options:\n\n\t1 - Want to Play with Smart Computer.");

printf("\n\n\t2 - Want to Play With Evil Computer.\n\n\t3 - Want to Play With Computer.\n\n\t4 - Exit Game.");

printf("\n\nSelect an option: ");

for(;;){

    scanf("%d", &op);

    switch(op){

        case 1:

            multiplayer();

        case 2:

            grid();

            printf("\tHow would you like to play?.\n\n");

            printf("\t1 - Play using 'X' (Player 1).\n\n\t2 - Play using 'O' (Player 2).");

            printf("\n\n\nSelect an option: ");

            scanf("%d", &x);

            while(x != 1 && x != 2){

                printf("\nERROR! Please select a valid option: ");

                scanf("%d", &x);

            }

            singleplayer(x);

        case 3:

            for(;;){

                ai\_opponent(1);

                result();

                ai\_opponent(2);

                result();

            }

            case 4:

                printf("\nThanks for Playing! :A");

                printf("\n\nMade by:\n\nMe ABHISHEK.");

                exit (EXIT\_SUCCESS);

        default:

            printf("\nERROR! Please select a valid option: ");

    }

}

}

void initialize\_rdm(void){

srand((unsigned) time(NULL));

}

void initialize\_grid(void){

int i, j;

char k = '1';

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

    for(j = 0; j < 3; j++){

        marks[i][j] = k++;

    }

}

}

void grid(void){

                 system("CLS");

                 printf("Tic-Tac-Toe: The Game / Ta-Te-Ti: El Juego\n");

                 printf("\n\t        |       |\n");

                 printf("\t      %c |   %c   | %c\n", marks[0][0], marks[0][1], marks[0][2]);

                 printf("\t        |       |\n");

                 printf("\t-------------------------\n");

                 printf("\t        |       |\n");

                 printf("\t      %c |   %c   | %c\n", marks[1][0], marks[1][1], marks[1][2]);

                 printf("\t        |       |\n");

                 printf("\t-------------------------\n");

                 printf("\t        |       |\n");

                 printf("\t      %c |   %c   | %c\n", marks[2][0], marks[2][1], marks[2][2]);

                 printf("\t        |       |\n\n");

}

int checker(char c, int player){

int i,j;

if(c < '1' || c > '9'){

    printf("\nERROR! Please select a valid grid: ");

    return 0;

}

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

    for(j = 0; j < 3; j++){

        if(marks[i][j] == c){

            switch(player){

                case 1:

                    counter++;

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

                    return 1;

                case 2:

                    counter++;

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

                    return 1;

            }

        }

    }

}

printf("\nGRID ALREADY FILLED!\n\nPlease select another grid: ");

return 0;

}

void result(void){

float condition;

int i, j, winner = 3;

grid();

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

    for(j = 0, condition = 0; j < 3; j++){ /\*checks rows\*/

        if(marks[i][j] == 'X' || marks[i][j] == 'O'){

            condition += marks[i][j];

        }

        if((condition / 'X') == 3.0){

            winner = 1;

        } else if((condition / 'O') == 3.0){

            winner = 2;

        }

    }

}

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

    for(i = 0, condition = 0; i < 3; i++){ /\*checks columns\*/

        if(marks[i][j] == 'X' || marks[i][j] == 'O'){

            condition += marks[i][j];

        }

        if((condition / 'X') == 3.0){

            winner = 1;

        } else if((condition / 'O') == 3.0){

            winner = 2;

        }

    }

}

for(i = 0, j = 0, condition = 0; i < 3; i++, j++){ /\*checks diagonally\*/

    if(marks[i][j] == 'X' || marks[i][j] == 'O'){

        condition += marks[i][j];

    }

    if((condition / 'X') == 3.0){

        winner = 1;

    } else if((condition / 'O') == 3.0){

        winner = 2;

    }

}

for(i = 2, j = 0, condition = 0; j < 3; i--, j++){ /\*checks diagonally\*/

    if(marks[i][j] == 'X' || marks[i][j] == 'O'){

        condition += marks[i][j];

    }

    if((condition / 'X') == 3.0){

        winner = 1;

    } else if((condition / 'O') == 3.0){

        winner = 2;

    }

}

if(counter >= 9 && winner == 3)

    winner = 0;

switch(winner){

    case 0:

        printf("\a\nIT'S A DRAW!");

        retry();

    case 1:

        printf("\aPLAYER 1 WINS!");

        retry();

    case 2:

        printf("\aPLAYER 2 WINS!");

        retry();

    default: return;

    }

}

void retry(void){

char c;

counter = 0;

printf("\n\nWould you like to play again?(Y/N): ");

scanf(" %c", &c);

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

    main();

} else{

    printf("\n\nThanks for Playing! :)");

    printf("\n\nMade by:\n\nInsert Students names xd.");

    exit(EXIT\_SUCCESS);

}

}

void ai\_opponent(int n){

int a, b, i;

for(;;){

    a = rand() % 3;

    b = rand() % 3;

    if(marks[a][b] != 'X' && marks[a][b] != 'O'){

        switch(n){

            case 1:

                marks[a][b] = 'X';

                counter++;

                return;

            case 2:

                marks[a][b] = 'O';

                counter++;

                return;

        }

    }

}

}

void singleplayer(int n){

char c;

grid();

for(;;){

    if(n == 1){

        printf("\nPlease select a grid: ");

        do{

            scanf(" %c", &c);

        } while(checker(c, n) != 1);

        result();

        ai\_opponent(2 / n);

    } else if(n == 2){

        ai\_opponent(2 / n);

        result();

        printf("\nPlease select a grid: ");

        do{

            scanf(" %c", &c);

        } while(checker(c, n) != 1);

    }

    result();

}

}

           void multiplayer(void){

           char c;

           grid();

for(;;){

    printf("\nPlayer 1: Please select a grid: ");

    do{

        scanf(" %c", &c);

    } while(checker(c, 1) != 1);

    result();

    printf("\nPlayer 2: Please select a grid: ");

    do{

        scanf(" %c", &c);

    } while(checker(c, 2) != 1);

    result();

}

}