Acknowledgement

My sincere thanks to Mrs .M. Lakshmi Radhika and

Mrs.M. Latha for their help during the course of my project. I thank them for giving me the freedom to explore different concepts, expand my boundaries and enhance my knowledge on Computer Science. I thank them for their valuable suggestions without which this would not have been possible.

I thank the school management for providing me with the necessary equipment to complete my project successfully.

I thank CBSE for giving me the opportunity to showcase my talents in the field of Programming.

Table of Contents

|  |  |  |
| --- | --- | --- |
| S.No | Topic | Page No |
| 1 | Project Description |  |
| 2 | Project Analysis |  |
| 3 | Source Code |  |
| 4 | Outputs |  |
| 5 | Scope and Limitation |  |
| 6 | Bibliography |  |

Project Description

The aim of our project was to bring the game known as ‘Ultimate Tic Tac Toe’ to this platform. The game consists of a large tic tac toe (3 x 3) grid, where each box has a smaller tic tac toe grid in it. The game is won where a player wins three consecutive boxes in the large tic tac toe – vertically, horizontally, or diagonally, or the game is drawn when it is established that neither player can win, or when no legal moves are left.

It is a menu driven programme which allows you to choose between:

* Multiplayer
* Single Player
* Instructions
* History
* Leaderboards
* Exit

Project Analysis

|  |  |  |
| --- | --- | --- |
| S.No | Name of the Function | Purpose |
| 1 | void print() | Displaying the whole tic tac toe grid |
| 2 | void instruction() | Dislplaying the instruction |
| 3 | void intro() | Introduction to the project |
| 4 | void endn() | Ending of the project |
| 5 | void square\_player2() | Holds all necessary information about the second player |
| 6 | void square\_player1() | Holds all necessary information about the first player |
| 7 | void comp() | Holds all necessary information about computer |
| 8 | int check() | Checks win of global and local boards |
| 9 | void reset() | Resets all arrays for every game |
| S.No | Name of the Function | Purpose |
| 10 | void histo() | Clears History |
| 11 | void filewrite() | Writes score of  winners into the a file |

SOURCE CODE

#include<fstream.h>

#include<dos.h>

#include<string.h>

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

#include<process.h>

int z[9],u,k1[9],k2[9],q,o,g,w=0,flag=0;

char player1[9][6],player2[9][6],grid\_check[9],grid\_check1[9],grid\_check2[9];

char player\_name[2][20];

void print(char a[9][3][3]);

void instruction();

void intro();

void endn();

void square\_player2(char a[9][3][3],char both\_players[9][10]);

void square\_player1(char a[9][3][3],char both\_players[9][10]);

void comp(char a[9][3][3],char both\_players[9][10]);

int check(char ch[5],int k);

void reset();

void histo();

void filewrite(int i);

void main(){

intro();

char ch;

do{

reset();

char both\_players[9][10];

int d;

clrscr();

cout<<"\t\t\t Welcome to Ultimate Tic Tac Toe \n\n";

cout<<"\t\t Player 1 : 'X' Player 2 : 'O' \n ";

cout<<"\n\t\t\t 1 for Multiplayer game \n";

cout<<"\n\t\t\t 2 for Computer vs human game \n";

cout<<"\n\t\t\t 3 for Instruction \n";

cout<<"\n\t\t\t 4 to History \n";

cout<<"\n\t\t\t 5 to Clear History \n";

cout<<"\n\t\t\t 6 to Exit \n";

d=getch();

clrscr();

char a[9][3][3]={'0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8','0','1','2','3','4','5','6','7','8'};

int g=0;

if(d==54)

endn();

else if(d==53)

histo();

else if(d==52)

filewrite(-1);

else if(d==51)

instruction();

else if(d==49){

print(a);

cout<<"Enter your name (1) : ";

cin>>player\_name[0];

cout<<"Enter your name (2) : ";

cin>>player\_name[1];

cout<<"Enter the global grid to start ,"<<player\_name[0]<<endl;

cin>>u;

while(w!=1){

square\_player1(a,both\_players);

g++;

if(w!=1)

square\_player2(a,both\_players);

if(g==41){

cout<<" Draw Game ";

delay(1000);

w=1;

}

}

}

else if(d==50){

print(a);

cout<<"Enter your name : ";

cin>>player\_name[0];

strcpy(player\_name[1],"Computer");

cout<<"Enter the global grid you want to start ,"<<player\_name[0]<<endl;

cin>>u;

while(w!=1){

square\_player1(a,both\_players);

g++;

if(w!=1)

comp(a,both\_players);

if(g==41){

cout<<" Draw Game ";

delay(1000);

w=1;

}

}

}

clrscr();

cout<<"Back to menu?(y/n)";

ch=getch();

}while(ch=='y');

endn();

}

void square\_player1(char a[9][3][3],char both\_players[9][10]){

clrscr();

int h=-1,temp,j,l;

textcolor(7);

print(a);

gotoxy(47,5);

cout<<" "<<player\_name[0]<<" has won global grids : ";

gotoxy(50,6);

for(int y=0;y<q;y++)

cout<<grid\_check1[y]<<",";

gotoxy(47,16);

cout<<" "<<player\_name[1]<<" has won global grids : ";

gotoxy(50,17);

for(y=0;y<o;y++)

cout<<grid\_check2[y]<<",";

gotoxy(1,18);

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

if(((char)u+48)==grid\_check[i]){

cout<<"\nEnter the global grid "<<player\_name[0];

cin>>u;

break;

}

if(g==9 ){

cout<<"Draw Game ";

delay(2000);

w=1;

goto again;

}

while(z[u]==9){

cout<<"The Global grid is fully filled \n";

cout<<"Enter a global grid "<<player\_name[0]<<'\n';

cin>>u;

}

textcolor(2+BLINK);

if(u==0){

gotoxy(1,3);cprintf("<");

gotoxy(16,3);cprintf(">");

}

else if(u==3){

gotoxy(1,9);cprintf("<");

gotoxy(16,9);cprintf(">");

}

else if(u==6){

gotoxy(1,15);cprintf("<");

gotoxy(16,15);cprintf(">");

}

else if(u==1){

gotoxy(17,3);cprintf("<");

gotoxy(32,3);cprintf(">");

}

else if(u==4){

gotoxy(17,9);cprintf("<");

gotoxy(32,9);cprintf(">");

}

else if(u==7){

gotoxy(17,15);cprintf("<");

gotoxy(32,15);cprintf(">");

}

else if(u==2){

gotoxy(33,3);cprintf("<");

gotoxy(48,3);cprintf(">");

}

else if(u==5){

gotoxy(33,9);cprintf("<");

gotoxy(48,9);cprintf(">");

}

else if(u==8){

gotoxy(33,15);cprintf("<");

gotoxy(48,15);cprintf(" >");

}

textcolor(7);

gotoxy(1,19);

cout<<"\nEnter a local grid "<<player\_name[0]<<'\n';

both\_players[u][z[u]]=getch();

if(both\_players[u][z[u]]=='q'){

w=1;

goto again;

}

both\_players[u][z[u]+1]='\0';

for(i=0;i<=z[u];i++)

for(j=0;j<=z[u];j++)

if((both\_players[u][i]==both\_players[u][j]&&i!=j)||(both\_players[u][j]>56&&both\_players[u][j]!=113)||both\_players[u][j]<48){

cout<<"Invalid entry "<<endl;

cout<<"Enter a local grid"<<player\_name[0]<<'\n';

both\_players[u][z[u]]=getch();

if(both\_players[u][z[u]]=='q'){

w=1;

goto again;

}

j--;

}

clrscr();

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

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

if(both\_players[u][z[u]]==a[u][i][j]){

a[u][i][j]='X';

temp=both\_players[u][z[u]];

}

}

print(a);

player1[u][k1[u]]=both\_players[u][z[u]];

k1[u]++;

if(check(player1[u],k1[u])){

for(l=0;l<=g;l++)

if(l!=g&&((char)u+48)==grid\_check[l])

h++;

if(h==-1){

grid\_check[g]=(char)u+48;

grid\_check1[q]=grid\_check[g];

q++;

g++;

if(check(grid\_check1,q)){

w=1;

cout<<player\_name[0]<<" has won the game! ";

delay(2000);

filewrite(1);

}

}

}

z[u]++;

u=temp-48;

again:;

}

void square\_player2(char a[9][3][3],char both\_players[9][10]){

int h=-1,temp,l,j;

textcolor(7);

gotoxy(47,5);

cout<<" "<<player\_name[0]<<" has won global grids : ";

gotoxy(50,6);

for(int y=0;y<q;y++)

cout<<grid\_check1[y]<<",";

gotoxy(47,16);

cout<<" "<<player\_name[1]<<" has won global grids : ";

gotoxy(50,17);

for(y=0;y<o;y++)

cout<<grid\_check2[y]<<",";

gotoxy(1,18);

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

if(((char)u+48)==grid\_check[i]){

cout<<"\nEnter the global grid "<<player\_name[1]<<'\n';

cin>>u;

break;

}

if(g==9){

cout<<"Draw Game";

delay(2000);

w=1;

goto back;

}

while(z[u]==9){

cout<<"The box is fully filled \n";

cout<<"Enter a global grid "<<player\_name[1]<<'\n';

cin>>u;

}

textcolor(2+BLINK);

if(u==0){

gotoxy(1,3);cprintf("<");

gotoxy(16,3);cprintf(">");

}

else if(u==3){

gotoxy(1,9);cprintf("<");

gotoxy(16,9);cprintf(">");

}

else if(u==6){

gotoxy(1,15);cprintf("<");

gotoxy(16,15);cprintf(">");

}

else if(u==1){

gotoxy(17,3);cprintf("<");

gotoxy(32,3);cprintf(">");

}

else if(u==4){

gotoxy(17,9);cprintf("<");

gotoxy(32,9);cprintf(">");

}

else if(u==7){

gotoxy(17,15);cprintf("<");

gotoxy(32,15);cprintf(">");

}

else if(u==2){

gotoxy(33,3);cprintf("<");

gotoxy(48,3);cprintf(">");

}

else if(u==5){

gotoxy(33,9);cprintf("<");

gotoxy(48,9);cprintf(">");

}

else if(u==8){

gotoxy(33,15);cprintf("<");

gotoxy(48,15);cprintf(" >");

}

textcolor(7);

gotoxy(1,19);

cout<<"\nEnter a local grid "<<player\_name[1]<<'\n';

both\_players[u][z[u]]=getch();

if(both\_players[u][z[u]]=='q'){

w=1;

goto back;

}

both\_players[u][z[u]+1]='\0';

for(i=0;i<=z[u];i++)

for(j=0;j<=z[u];j++)

if((both\_players[u][i]==both\_players[u][j]&&i!=j)||(both\_players[u][j]>56&&both\_players[u][j]!=113)||both\_players[u][j]<48){

cout<<"Invalid entry"<<endl;

cout<<"Enter a local grid "<<player\_name[1]<<'\n';

both\_players[u][z[u]]=getch();

if(both\_players[u][z[u]]=='q'){

w=1;

goto back;

}

j--;

}

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

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

if(both\_players[u][z[u]]==a[u][i][j])

a[u][i][j]='O';

temp=both\_players[u][z[u]];

}

print(a);

player2[u][k2[u]]=both\_players[u][z[u]];

k2[u]++;

if(check(player2[u],k2[u])){

for(l=0;l<=g;l++)

if(l!=g&&((char)u+48)==grid\_check[l])

h++;

if(h==-1){

grid\_check[g]=(char)u+48;

grid\_check2[o]=grid\_check[g];

g++;

o++;

if(check(grid\_check2,o)){

clrscr();

print(a);

w=1;

cout<<player\_name[1]<<" has won the game! ";

delay(2000);

filewrite(0);

}

}

}

z[u]++;

u=temp-48;

back:;

}

void comp(char a[9][3][3],char both\_players[9][10]){

randomize();

int q;

textcolor(7);

gotoxy(47,5);

cout<<" "<<player\_name[0]<<" has won global grids : ";

gotoxy(50,6);

for(int y=0;y<::q;y++)

cout<<grid\_check1[y]<<",";

gotoxy(47,16);

cout<<" "<<player\_name[1]<<" has won global grids : ";

gotoxy(50,17);

for(y=0;y<o;y++)

cout<<grid\_check2[y]<<",";

gotoxy(1,18);

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

if(((char)u+48)==grid\_check[i]){

q=random(9);

if(q!=u){

u=q;

break;

}

}

while(z[u]==9)

u=random(9);

textcolor(2+BLINK);

if(u==0){

gotoxy(1,3);cprintf("<");

gotoxy(16,3);cprintf(">");

}

else if(u==3){

gotoxy(1,9);cprintf("<");

gotoxy(16,9);cprintf(">");

}

else if(u==6){

gotoxy(1,15);cprintf("<");

gotoxy(16,15);cprintf(">");

}

else if(u==1){

gotoxy(17,3);cprintf("<");

gotoxy(32,3);cprintf(">");

}

else if(u==4){

gotoxy(17,9);cprintf("<");

gotoxy(32,9);cprintf(">");

}

else if(u==7){

gotoxy(17,15);cprintf("<");

gotoxy(32,15);cprintf(">");

}

else if(u==2){

gotoxy(33,3);cprintf("<");

gotoxy(48,3);cprintf(">");

}

else if(u==5){

gotoxy(33,9);cprintf("<");

gotoxy(48,9);cprintf(">");

}

else if(u==8){

gotoxy(33,15);cprintf("<");

gotoxy(48,15);cprintf(" >");

}

textcolor(7);

gotoxy(1,19);

cout<<"Computer is thinking ";

delay(2000);

q=random(9);

both\_players[u][z[u]]=(char)q+48;

both\_players[u][z[u]+1]='\0';

for(i=0;both\_players[u][i]!='\0';i++)

for(int j=0;both\_players[u][j]!='\0';j++)

if(both\_players[u][i]==both\_players[u][j]&&(i!=j)){ q=random(9);

both\_players[u][z[u]]=(char)q+48;

delay(1000);

both\_players[u][z[u]+1]='\0';

j--;

}

int k=0,temp;

clrscr();

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

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

if(both\_players[u][z[u]]==a[u][i][j])

a[u][i][j]='O';

}

print(a);

player2[u][k2[u]]=both\_players[u][z[u]];

k2[u]++;

int h=-1;

if(check(player2[u],k2[u])){

for(int l=0;l<=g;l++)

if(l!=g&&((char)u+48)==grid\_check[l])

h++;

if(h==-1){

grid\_check[g]=(char)u+48;

grid\_check2[o]=grid\_check[g];

g++;

o++;

if(check(grid\_check2,o)){

cout<<"Computer has won the game! ";

w=1;

delay(5000);

filewrite(0);

}

}

}

z[u]++;

u=q;

}

int check(char ch[5],int k){

char c[3],a[25]={'0','1','2','3','4','5','6','7','8','0','3','6','1','4','7','2','5','8','0','4','8','2','4','6'};

int i=0,j=0,count=0,l=0,q,z=0;

a[25]='\0';

for(int p=0;p<8;p++){

q=i+3;

for(i;i<q;i++,z++){

c[z]=a[i];

c[z+1]='\0';

}

for(j=0;j<k;j++)

for(l=0;l<3;l++)

if(ch[j]==c[l])

count++;

if(count==3)

return 1;

count=0;

z=0;

}

return 0;

}

void print(char a[9][3][3]){

int k=0;

int color\_X=0xC;

int color\_O=0xB;

for(int i=0;i<9;i++){

int decideBoxes = i/3;

int runJ = decideBoxes \* 3;

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

if(a[j][i%3][k]!='X'&&a[j][i%3][k]!='O')

cout<<" "<<a[j][i%3][k];

else if(a[j][i%3][k]=='X'){

cout<<" ";

textcolor(color\_X);

cprintf("X");

textcolor(7);

}

else{

cout<<" ";

textcolor(color\_O);

cprintf("O");

textcolor(7);

}

if(a[j][i%3][k+1]!='X'&&a[j][i%3][k+1]!='O')

cout<<" | "<<a[j][i%3][k+1];

else if(a[j][i%3][k+1]=='X'){

cout<<" | ";

textcolor(color\_X);

cprintf("X");

textcolor(7);

}

else{

cout<<" | ";

textcolor(color\_O);

cprintf("O");

textcolor(7);

}

if(a[j][i%3][k+2]!='X'&&a[j][i%3][k+2]!='O')

cout<<" | "<<a[j][i%3][k+2]<<" ";

else if(a[j][i%3][k+2]=='X'){

cout<<" | ";

textcolor(color\_X);

cprintf("X ");

textcolor(7);

}

else{

cout<<" | ";

textcolor(color\_O);

cprintf("O ");

textcolor(7);}

}

if(i%3!=2)

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

if(i%3==2)

cout<<endl;

cout<<endl;

}

}

void reset(){

u=q=o=g=w=0;

for(int i=0;i<9;i++)

for(int j=0;j<6;j++){

z[i]=0;

k1[i]=0;

k2[i]=0;

player1[i][j]=' ';

player2[i][j]=' ';

grid\_check[i]=' ';

grid\_check1[i]=' ';

grid\_check2[i]=' ';

}

}

void instruction(){

clrscr();

ifstream f;

char ch[2000];

cout<<"INSTRUCTIONS";

cout<<"\n\n->Each small 3-by-3 tic-tac-toe board is referred to as a local board, and the larger 3-by-3 board is referred to as the global board.";

cout<<"\n\n->The game starts with X playing wherever he wants in any of the 81 empty spots.";

cout<<"\n->This move 'sends' his opponent to its relative location. For example, if X played in the middle of his local board, then O needs to play next in the local board in the middle of the global board.";

cout<<"\n\n->If a move is played so that it is to win a local board by the rules of normal tic-tac-toe, then it wins that local board.";

cout<<"\n\n->Once the outcome of a local board is decided (win or draw), if a player is sent to such a board, then that player may play in any other board. Game play ends when a player wins the global board, or there are no legal moves left.";

cout<<"\n\nThe local grid goes this way:(Grids are numbered from 0-8)\n";

cout<<"\n 0 | 1 | 2\n";

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

cout<<" 3 | 4 | 5\n";

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

cout<<" 6 | 7 | 8\n";

cout<<"-->Press q to come back to menu while playing the game";

getch();

}

void histo(){

flag=0;

fstream f("history.txt",ios::trunc|ios::out);

}

void filewrite(int i){

char ch[2000],c;

if(i==1){

fstream f("history.txt",ios::out|ios::app);

f<<player\_name[0]<<" has won "<<player\_name[1]<<".";

f.close();

}

else if(i==0){

fstream f("history.txt",ios::out|ios::app);

f<<player\_name[1]<<" has won "<<player\_name[0]<<".";

f.close();

}

else{

flag=0;

clrscr();

fstream f2("history.txt",ios::in);

while(!f2.eof()){

flag++;

f2.getline(ch,2000,'.');

cout<<ch<<endl;

}

if(flag==1)

cout<<"No Records Found! ";

getch();

f2.close();

}

}

void intro(){

int gdriver=DETECT,gmode;

initgraph(&gdriver,&gmode,"C:\\TURBOC3\\BGI");

settextstyle(1,0,5);

setcolor(BLUE);

outtextxy(100,140,"Ultimate Tic Tac Toe");

settextstyle(1,0,4);

setcolor(BLUE);

outtextxy(160,300,"By Badri and Madhav 12F1");

getch();

setcolor(RED);

for(int i=0;i<200;i++){

circle(300,250,i);

delay(3);

}

closegraph();

}

void endn(){

int gdriver=DETECT,gmode;

initgraph(&gdriver,&gmode,"C:\\TURBOC3\\BGI");

settextstyle(8,0,5);

setcolor(RED);

outtextxy(100,150,"By Badri and Madhav");

getch();

setcolor(GREEN);

for(int i=0;i<200;i++){

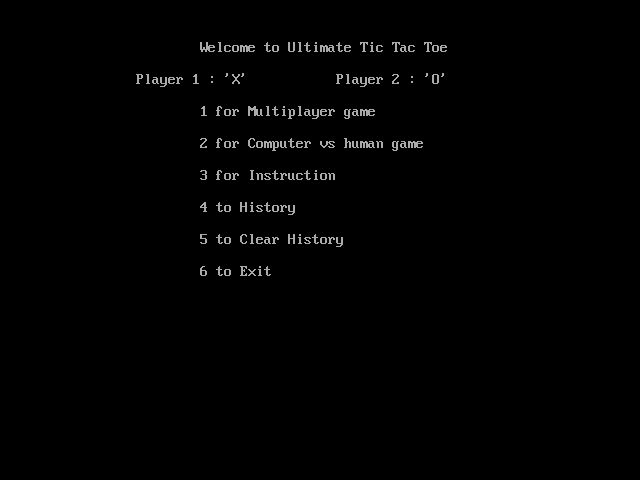
circle(300,250,i);

delay(3);

}

exit(0);

}

 Outputs

