**DATE :** 12/4/2017

**Project:** Making a game using C++.

**S/W Requirements:** Operating system, Codeblocks.

**H/W Requirements:** Keyboard, Mouse, CPU, Motherboard, Hard disk, Printers.

**Title: Tic Tac Toe**

**Length: 1700+ lines.**

**Completed in: 5 days.**

**Project Description:**

**1) Moving cursor at respective X and Y coordinate:**

**#include “windows.h”**

void point(int px,int py)

{

COORD coord;

coord.X = px;

coord.Y = py;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

1. **Drawing of Horizontal line:**

void hline(int l)

{

cout.fill(196);

cout.width(l);

cout<<(char)196;

}

1. **Drawing of vertical line:**

void vline(int l,int lx,int ly)

{

while(l>0)

{

point(lx,ly++);

cout<<(char)179;

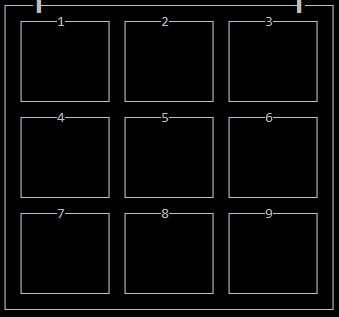
l--;

}

temp1=ly;

}

**4) Making of boards:**



int x,y,hx,hy;

void setposition(int start\_x,int start\_y)

{

x=start\_x;

y=start\_y;

}

**a) Drawing Rectangles:**

void structure(float a,float b)

{

hx=a,hy=b;

point(x,y);

cout<<(char)218;

hline(hx);

cout<<(char)191;

vline(hy,x+hx+1,y+1);

point(x,temp1);

cout<<(char)192;

hline(hx);

cout<<(char)217;

point(x,y+1);

vline(hy,x,y+1);

point(x,temp1+2);

}

**b) Making complete Board:**

void draw\_board(int dx,int dy,int l,int b)

{

setposition(dx,dy);

structure(l,b);

l=l/4;

b=b/4;

setposition(dx+2,dy+1);

structure(l,b);

point(dx+7,dy+1);

cout<<1;

setposition(dx+15,dy+1);

structure(l,b);

point(dx+20,dy+1);

cout<<2;

setposition(dx+28,dy+1);

structure(l,b);

point(dx+33,dy+1);

cout<<3;

setposition(dx+2,dy+7);

structure(l,b);

point(dx+7,dy+7);

cout<<4;

setposition(dx+15,dy+7);

structure(l,b);

point(dx+20,dy+7);

cout<<5;

setposition(dx+28,dy+7);

structure(l,b);

point(dx+33,dy+7);

cout<<6;

setposition(dx+2,dy+13);

structure(l,b);

point(dx+7,dy+13);

cout<<7;

setposition(dx+15,dy+13);

structure(l,b);

point(dx+20,dy+13);

cout<<8;

setposition(dx+28,dy+13);

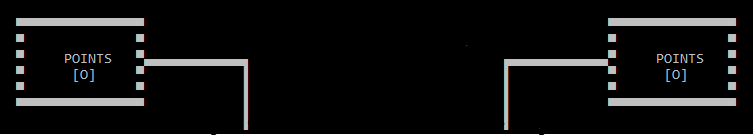
structure(l,b);

point(dx+33,dy+13);

cout<<9;

}

**5) Making of Score Panel:**



**a) Making of advanced horizontal line:**

void ahline(char design,int length,int position\_x,int position\_y,char direction)

{

Sleep(20);

for(int a=0;a<length;a++)

{

if(direction=='l'||direction=='L')

{

point(position\_x--,position\_y);

cout<<(char)design;

}

else if(direction=='r'||direction=='R')

{

point(position\_x++,position\_y);

cout<<(char)design;

}

}

}

**b) Making of advanced vertical line:**

void avline(char design,int length,int position\_x,int position\_y,char direction)

{

for(int a=0;a<length;a++)

{

Sleep(20);

if(direction=='d'||direction=='D')

{

point(position\_x,position\_y++);

cout<<(char)design;

}

else if(direction=='u'||direction=='U')

{

point(position\_x,position\_y--);

cout<<(char)design;

}

}

}

**c) Completing Score panel:**

void draw\_score()

{

int lx=34,ly=9,rx=67,ry=9;

char c2=220,c3=223;

point(lx,ly+1);

cout<<(char)222;

point(rx,ry+1);

cout<<(char)221;

avline(222,4,lx,ly,'u');

avline(221,4,rx,ry,'u');

ahline(c2,13,lx,ly-3,'l');

ahline(c2,13,rx,ry-3,'r');

avline(c3,2,rx+13,ry-3,'u');

ahline(c3,15,rx+13,ry-5,'r');

avline(c3,5,rx+28,ry-5,'d');

ahline(c3,15,rx+28,ry,'l');

avline(c3,5,rx+13,ry,'u');

avline(c3,2,lx-13,ly-3,'u');

ahline(c3,15,lx-13,ly-5,'l');

avline(c3,5,lx-28,ly-5,'d');

ahline(c3,15,lx-28,ly,'r');

avline(c3,5,lx-13,ly,'u');

}

**6) Making of Zero and Cross:**

C:\Users\ashish\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cross.png 

void printing\_tic\_and\_toe(int value,int position\_x,int position\_y)

{

int bx=position\_x,by=position\_y;

if(value==1) // for Cross

{

point(bx+1,by);

cout<<(char)254;

Sleep(150);

point(bx-1,by-1);

cout<<(char)254<<" "<<(char)254;

point(bx-1,by+1);

cout<<(char)254<<" "<<(char)254;

}

else if(value ==4) // for Zero

{

point(bx,by-1);

cout<<(char)254;

Sleep(40);

point(bx-2,by);

cout<<(char)254;

Sleep(40);

point(bx-2,by+1);

cout<<(char)254;

Sleep(40);

point(bx,by+2);

cout<<(char)254<<" ";

Sleep(40);

cout<<(char)254;

point(bx+4,by+1);

Sleep(40);

cout<<(char)254;

point(bx+4,by);

Sleep(40);

cout<<(char)254;

point(bx+2,by-1);

Sleep(40);

cout<<(char)254;

}

}

**7) Making of Title:**

C:\Users\ashish\AppData\Local\Microsoft\Windows\INetCache\Content.Word\title.png

void title(int wx,int wy)

{

// T

ahline(220,5,wx,wy-1,'r');

avline(219,2,wx+2,wy,'d');

// I

point(wx+4,wy);

cout<<(char)220;

avline(219,1,wx+4,wy+1,'d');

//C

ahline(220,2,wx+7,wy,'l');

avline(219,1,wx+6,wy+1,'d');

ahline(220,1,wx+7,wy+1,'r');

//T

ahline(220,3,wx+10,wy,'r');

avline(219,1,wx+11,wy+1,'d');

//A

avline(219,1,wx+14,wy+1,'u');

ahline(220,3,wx+14,wy,'r');

avline(219,1,wx+16,wy+1,'d');

point(wx+15,wy+1);

cout<<(char)205;

// C

ahline(220,2,wx+19,wy,'l');

avline(219,1,wx+18,wy+1,'d');

ahline(220,1,wx+19,wy+1,'r');

//T

ahline(220,3,wx+22,wy,'r');

avline(219,1,wx+23,wy+1,'d');

// O

ahline(220,2,wx+27,wy,'l');

avline(219,1,wx+26,wy+1,'d');

ahline(220,1,wx+27,wy+1,'r');

avline(219,1,wx+28,wy+1,'u');

point(wx+28,wy);

cout<<(char)220;

//E

ahline(220,3,wx+32,wy-1,'l');

avline(219,2,wx+30,wy,'d');

ahline(220,2,wx+31,wy+1,'r');

point(wx+31,wy);

cout<<(char)220;

}

**8) Making of Eraser:**

void Erase(int length,int line,int position\_x,int position\_y)

{

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

{

copy\_ahline(0,length,position\_x,position\_y++,'r');

}

}

**9) Printing with delay:**

void delay(char \*p,int time)

{

for(int i=0;i<strlen(p);i++)

{

cout<<\*(p+i);

Sleep(time);

}

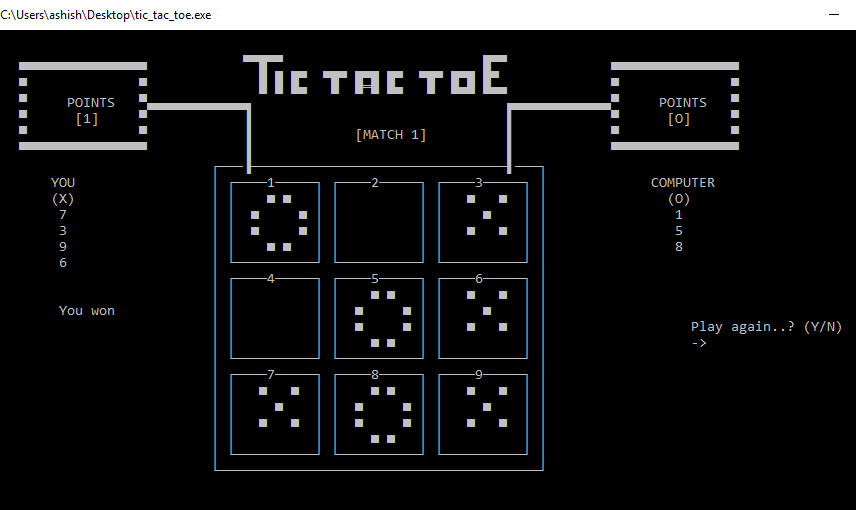
}

**10) Flashing of Score Panel:**

Score panel of the respective player flashes whenever there is an increment in the points. In case of Tie score panel of both the player flashes since both gets 1 point each.

The basic idea behind flashing of Score panel is putting null character itself while calling the function used for making score panel and then calling its original version again. The process is repeated to the number of times flash is required. In between calling, sleep function is used for observing time difference.

**11) Starting the game:**



**a)** **Creating 3\*3 matrix:**

int matrix[3][3];

**b)** **declaring necessary variables:**

int x,y;

int temp1;

int temp\_return;

int p1,p2,point1,point2,first\_move;

int score1[20],score2[20];

**c)** **Constructor:**

tictactoe()

{

p1=0;

p2=0;

first\_move=1;

}

**d)** **Initialising matrix and other variables:**

void initializer()

{

point1=0;

point2=0;

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

{

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

{

matrix[i][j] = 0;

}

}

temp\_return=0;

}

**e)** **Basic idea behind the game:**

I) The game has 9 entries which are 9 different elements of 3\*3 matrix.

II) The game should consists of exactly 2 players.

III) The game should go alternate, i.e If first chance of player 1 then second chance must be of player 2 and vice versa till all the entries were filled.

It was done using even – odd no concept.

IV) The game should terminate if all the entries were filled.

V) One player cannot renter or enter to that entry which is already filled.

VI) All the entries must be filled by taking input from the player in integer format and it must be from 1 to 9.

VII) Since there are two players, so in matrix value should be given accordingly.

for player 1 , I have given value = 1 and for player 2 value given is 4 so that both the values must not alter each other later while checking winning chances for both the players.

VIII) Deciding winner – For this there is a need to check sum of all the row, column and diagonal entries. If sum comes out to be 3 then player 1 is winner. If it comes out to be 12 then player 2 is winner.

With these ideas the root function was made.

**f)** **Checking for valid entry:**

int check()

{

int r1,r2,r3,c1,c2,c3,d1,d2;

r1=matrix[0][0]+matrix[0][1]+matrix[0][2];

r2=matrix[1][0]+matrix[1][1]+matrix[1][2];

r3=matrix[2][0]+matrix[2][1]+matrix[2][2];

c1=matrix[0][0]+matrix[1][0]+matrix[2][0];

c2=matrix[0][1]+matrix[1][1]+matrix[2][1];

c3=matrix[0][2]+matrix[1][2]+matrix[2][2];

d1=matrix[0][0]+matrix[1][1]+matrix[2][2];

d2=matrix[0][2]+matrix[1][1]+matrix[2][0];

if(r1==3||r2==3||r3==3||c1==3||c2==3||c3==3||d1==3||d2==3)

{

return 0;

}

else if(r1==12||r2==12||r3==12||c1==12||c2==12||c3==12||d1==12||d2==12)

{

return 1;

}

else

return 2;

}

**g)** **Checking for game end:**

int check2()

{

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

{

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

{

if(matrix[i][j]==0)

return 0; // atleast one entry is remaining

}

}

return 1; //all entries were filled

}

**h)** **Working on Single player mode:**

Since the game involve exactly two players, So the second player must be a computer. Since computer do not have a brain as human, So there is a need for making a logic such that probability of winning of player as well as computer remain 1/2.

* **Algorithm behind working of Computer:**

char logic()

{

int r1,r2,r3,c1,c2,c3,d1,d2;

r1=matrix[0][0]+matrix[0][1]+matrix[0][2];

r2=matrix[1][0]+matrix[1][1]+matrix[1][2];

r3=matrix[2][0]+matrix[2][1]+matrix[2][2];

c1=matrix[0][0]+matrix[1][0]+matrix[2][0];

c2=matrix[0][1]+matrix[1][1]+matrix[2][1];

c3=matrix[0][2]+matrix[1][2]+matrix[2][2];

d1=matrix[0][0]+matrix[1][1]+matrix[2][2];

d2=matrix[0][2]+matrix[1][1]+matrix[2][0];

if(r1==8)

{

if(matrix[0][0]==0)

return '1';

else if(matrix[0][1]==0)

return '2';

else if(matrix[0][2]==0)

return '3';

}

else if(r2==8)

{

if(matrix[1][0]==0)

return '4';

else if(matrix[1][1]==0)

return '5';

else if(matrix[1][2]==0)

return '6';

}

else if(r3==8)

{

if(matrix[2][0]==0)

return '7';

else if(matrix[2][1]==0)

return '8';

else if(matrix[2][2]==0)

return '9';

}

else if(c1==8)

{

if(matrix[0][0]==0)

return '1';

else if(matrix[1][0]==0)

return '4';

else if(matrix[2][0]==0)

return '7';

}

else if(c2==8)

{

if(matrix[0][1]==0)

return '2';

else if(matrix[1][1]==0)

return '5';

else if(matrix[2][1]==0)

return '8';

}

else if(c3==8)

{

if(matrix[0][2]==0)

return '3';

else if(matrix[1][2]==0)

return '6';

else if(matrix[2][2]==0)

return '9';

}

else if(d1==8)

{

if(matrix[0][0]==0)

return '1';

else if(matrix[1][1]==0)

return '5';

else if(matrix[2][2]==0)

return '9';

}

else if(d2==8)

{

if(matrix[0][2]==0)

return '3';

else if(matrix[1][1]==0)

return '5';

else if(matrix[2][0]==0)

return '7';

}

else if(r1==2)

{

if(matrix[0][0]==0)

return '1';

else if(matrix[0][1]==0)

return '2';

else if(matrix[0][2]==0)

return '3';

}

else if(r2==2)

{

if(matrix[1][0]==0)

return '4';

else if(matrix[1][1]==0)

return '5';

else if(matrix[1][2]==0)

return '6';

}

else if(r3==2)

{

if(matrix[2][0]==0)

return '7';

else if(matrix[2][1]==0)

return '8';

else if(matrix[2][2]==0)

return '9';

}

else if(c1==2)

{

if(matrix[0][0]==0)

return '1';

else if(matrix[1][0]==0)

return '4';

else if(matrix[2][0]==0)

return '7';

}

else if(c2==2)

{

if(matrix[0][1]==0)

return '2';

else if(matrix[1][1]==0)

return '5';

else if(matrix[2][1]==0)

return '8';

}

else if(c3==2)

{

if(matrix[0][2]==0)

return '3';

else if(matrix[1][2]==0)

return '6';

else if(matrix[2][2]==0)

return '9';

}

else if(d1==2)

{

if(matrix[0][0]==0)

return '1';

else if(matrix[1][1]==0)

return '5';

else if(matrix[2][2]==0)

return '9';

}

else if(d2==2)

{

if(matrix[0][2]==0)

return '3';

else if(matrix[1][1]==0)

return '5';

else if(matrix[2][0]==0)

return '7';

}

else

{

int r;

srand(time(0));

while(r=rand()%9+1)

{

if(matrix[0][0]==0 && r==1)

return '1';

else if(matrix[0][1]==0 && r==2)

return '2';

else if(matrix[0][2]==0 && r==3)

return '3';

else if(matrix[1][0]==0 && r==4)

return '4';

else if(matrix[1][1]==0 && r==5)

return '5';

else if(matrix[1][2]==0 && r==6)

return '6';

else if(matrix[2][0]==0 && r==7)

return '7';

else if(matrix[2][1]==0 && r==8)

return '8';

else if(matrix[2][2]==0 && r==9)

return '9';

}

}

}

**i)** **Root function( in class tictactoe):**

void fun(int mode)

{

char user,computer;

int ux=8,uy=13,cx=85,cy=13;

int a=first\_move; // change later

while(a)

{

if(a%2!=0)

{

if(a!=1)

{

point(cx,cy-1);

cout<<" "<<computer<<" ";

point(cx,cy);

cout<<" ";

}

point(ux,uy++);

if(mode==1)

cout<<"Your turn";

else cout<<"Player 1 turn";

point(ux,uy);

cout<<"-> ";

while((user=cin.get())=='\n'||user==32)

{

point(ux+3,uy);

}

fflush(stdin);

while(setinput(user,1)==0||temp\_return==1) //while(setinput(user,1)==0||!(user>=0 && user<=9))

{

point(ux,uy-1);

cout<<" ";

point(ux+3,uy);

if(setinput(user,1)==0)

cout<<"already entered";

else

cout<<"invalid input";

Sleep(1000);

point(ux,uy-1);

if(mode==1)

cout<<"Your turn";

else cout<<"Player 1 turn";

point(ux,uy);

cout<<"-> ";

point(ux+3,uy);

while((user=cin.get())=='\n'||user==32)

{

point(ux+3,uy);

}

fflush(stdin);

}

if(check()==0)

{

point(ux,uy-1);

cout<<" "<<user<<" ";

point(ux,uy);

cout<<" ";

point(ux,uy+2);

if(mode==1)

cout<<" You won";

else cout<<"Player 1 won";

a=0;

p1++;

point1++;

point(13,7);

cout<<"["<<p1<<"]";

int flash=3;

while(flash>0)

{

erase\_score(0,0,0,0,1);

point(0,15);

Sleep(500);

erase\_score(222,220,223,221,1);

point(0,15);

Sleep(500);

flash--;

}

break;

}

else if(check()==1)

{

point(cx,cy+2);

if(mode==1)

cout<<" Computer won";

else

cout<<"Player 2 won";

a=0;

p2++;

point2++;

point(87,7);

cout<<"["<<p2<<"]";

int flash=3;

while(flash>0)

{

erase\_score(0,0,0,0,2);

point(0,15);

Sleep(500);

erase\_score(222,220,223,221,2);

point(0,15);

Sleep(500);

flash--;

}

break;

}

else if(check()==2 && check2()==1)

{

point(ux,uy-1);

cout<<" "<<user<<" ";

point(ux,uy);

cout<<" ";

point(ux,uy+2);

cout<<" TIE";

p1++;

p2++;

point1++;

point2++;

point(13,7);

cout<<"["<<p1<<"]";

point(87,7);

cout<<"["<<p2<<"]";

int flash=3;

while(flash>0)

{

erase\_score(0,0,0,0,3);

point(0,15);

Sleep(500);

erase\_score(222,220,223,221,3);

point(0,15);

Sleep(500);

flash--;

}

a=0;

break;

}

}

else if(a%2==0)

{

point(ux,uy-1);

cout<<" "<<user<<" ";

point(ux,uy);

cout<<" ";

point(10,12);

cout<<"(X)";

point(cx,cy++);

if(mode==1)

cout<<"Computer turn";

else

cout<<"Player 2 turn";

point(cx,cy);

if(mode==1)

{

cout<<"-> ";

Sleep(30);

point(cx+3,cy);

cout<<"thinking";

point(cx+11,cy);

delay("...",500);

Sleep(20);

point(cx+11,cy);

cout<<" ";

point(cx+11,cy);

delay("...",300);

Sleep(20);

point(cx+3,cy);

cout<<" ";

}

else

cout<<"-> ";

if(mode==1)

{

computer=logic();

setinput(computer,4);

}

else

{

while((computer=cin.get())=='\n'||user==32)

{

point(cx+3,cy);

}

fflush(stdin);

while(setinput(computer,4)==0||temp\_return==1)

{

point(cx,cy-1);

cout<<" ";

point(cx+3,cy);

if(setinput(computer,4)==0)

cout<<"already entered";

else

cout<<"invalid input";

Sleep(1000);

point(cx,cy-1);

if(mode==1)

cout<<"Computer turn";

else

cout<<"Player 2 turn";

if(mode==1)

{

cout<<"-> ";

Sleep(30);

point(cx+3,cy);

cout<<"thinking";

point(cx+11,cy);

delay("...",500);

Sleep(20);

point(cx+11,cy);

cout<<" ";

point(cx+11,cy);

delay("...",300);

Sleep(20);

point(cx+3,cy);

cout<<" ";

}

else

{

point(cx+3,cy);

cout<<" ";

}

point(cx+3,cy);

while((computer=cin.get())=='\n'||user==32)

{

point(ux+3,uy);

}

fflush(stdin); //cin>>computer;

}

}

if(check()==0)

{

point(ux,uy+2);

if(mode==1)

cout<<" You won";

else

cout<<"Player 1 won";

a=0;

p1++;

point1++;

point(13,7);

cout<<"["<<p1<<"]";

int flash=3;

while(flash>0)

{

erase\_score(0,0,0,0,1);

point(0,15);

Sleep(500);

erase\_score(222,220,223,221,1);

point(0,15);

Sleep(500);

flash--;

}

break;

}

else if(check()==1)

{

point(cx,cy-1);

cout<<" "<<computer<<" ";

point(cx,cy);

cout<<" ";

point(cx,cy+2);

if(a=10)

point(cx,cy+1);

if(mode==1)

cout<<" Computer won";

else

cout<<"Player 2 won";

a=0;

p2++;

point2++;

point(87,7);

cout<<"["<<p2<<"]";

int flash=3;

while(flash>0)

{

erase\_score(0,0,0,0,2);

point(0,15);

Sleep(500);

erase\_score(222,220,223,221,2);

point(0,15);

Sleep(500);

flash--;

}

break;

}

else if(check()==2 && check2()==1)

{

point(cx,cy-1);

cout<<" "<<computer<<" ";

point(cx,cy);

cout<<" ";

point(cx,cy+2);

if(a=10)

point(cx,cy+1);

cout<<" TIE";

a=0;

p1++;

p2++;

point1++;

point2++;

point(13,7);

cout<<"["<<p1<<"]";

point(87,7);

cout<<"["<<p2<<"]";

int flash=3;

while(flash>0)

{

erase\_score(0,0,0,0,3);

point(0,15);

Sleep(500);

erase\_score(222,220,223,221,3);

point(0,15);

Sleep(500);

flash--;

}

break;

} }

a++;

} }

**12)** **Working on Scoreboard:**



**a)** **Making space free inside board:**

*\*here copyahline(...) and copyavline(...) are modified version of function ahline(...) and avline(...) to skip delaying in output.*

void erase\_score(char c1,char c2,char c3,char c4,int mode)

{

int lx=34,ly=9,rx=67,ry=9;

if(mode==1)

{

point(lx,ly+1);

cout<<(char)c1;

copy\_avline(c1,4,lx,ly,'u');

copy\_ahline(c2,13,lx,ly-3,'l');

copy\_avline(c3,2,lx-13,ly-3,'u');

copy\_ahline(c3,15,lx-13,ly-5,'l');

copy\_avline(c3,5,lx-28,ly-5,'d');

copy\_ahline(c3,15,lx-28,ly,'r');

copy\_avline(c3,5,lx-13,ly,'u');

}

else if(mode==2)

{

point(rx,ry+1);

cout<<(char)c4;

copy\_avline(c4,4,rx,ry,'u');

copy\_ahline(c2,13,rx,ry-3,'r');

copy\_avline(c3,2,rx+13,ry-3,'u');

copy\_ahline(c3,15,rx+13,ry-5,'r');

copy\_avline(c3,5,rx+28,ry-5,'d');

copy\_ahline(c3,15,rx+28,ry,'l');

copy\_avline(c3,5,rx+13,ry,'u');

}

else if(mode==3)

{

point(lx,ly+1);

cout<<(char)c1;

point(rx,ry+1);

cout<<(char)c4;

copy\_avline(c1,4,lx,ly,'u');

copy\_ahline(c2,13,lx,ly-3,'l');

copy\_avline(c3,2,lx-13,ly-3,'u');

copy\_ahline(c3,15,lx-13,ly-5,'l');

copy\_avline(c3,5,lx-28,ly-5,'d');

copy\_ahline(c3,15,lx-28,ly,'r');

copy\_avline(c3,5,lx-13,ly,'u');

copy\_avline(c4,4,rx,ry,'u');

copy\_ahline(c2,13,rx,ry-3,'r');

copy\_avline(c3,2,rx+13,ry-3,'u');

copy\_ahline(c3,15,rx+13,ry-5,'r');

copy\_avline(c3,5,rx+28,ry-5,'d');

copy\_ahline(c3,15,rx+28,ry,'l');

copy\_avline(c3,5,rx+13,ry,'u');

}

}

**b)** **printing on scoreboard:**

void scoreboard(int total\_matches,int mode)

{

Erase(40,17,31,11);

point(47,11);

delay("SCOREBOARD",100);

if(mode==1)

{

point(44,12);

delay("You vs Computer",100);

}

else if(mode==2)

{

point(44,12);

delay("Player 1 vs Player 2",100);

}

point(44,13);

delay("Total matches : ",100);

cout<<total\_matches;

point(48,15);

if(mode==1)

{

Sleep(300);

cout<<" YOU";

point(60,15);

Sleep(300);

cout<<"COMPUTER";

}

else if(mode==2)

{

Sleep(200);

cout<<"PLAYER 1";

point(60,15);

Sleep(200);

cout<<"PLAYER 2";

}

point(46,16);

int temp5=24;

while(temp5>0)

{

cout<<(char)196;

Sleep(20);

temp5--;

}

temp5=total\_matches+2;

int tx=57,ty=15;

while(temp5>0)

{

point(tx,ty++);

cout<<(char)179;

Sleep(50);

temp5--;

}

tx=35,ty=17;

for(int i=1;i<=total\_matches;i++)

{

point(tx,ty);

Sleep(200);

cout<<"-> ";

Sleep(400);

point(tx+3,ty);

cout<<"Match "<<i<<" : ";

Sleep(100);

point(tx,ty);

cout<<" ";

ty++;

}

for(int i=0;i<total\_matches;i++)

{

Sleep(400);

point(51,17+i);

cout<<score1[i];

Sleep(400);

point(64,17+i);

cout<<score2[i];

}

temp5=24;

point(46,ty);

while(temp5>0)

{

Sleep(20);

cout<<(char)196;

temp5--;

}

point(35,ty+1);

Sleep(200);

cout<<"-> ";

Sleep(400);

point(38,ty+1);

cout<<"Result : ";

Sleep(100);

point(35,ty+1);

cout<<" ";

point(51,ty+1);

Sleep(400);

cout<<p1;

point(64,ty+1);

Sleep(400);

cout<<p2;

int flash=7;

while(flash>0)

{

point(0,15);

if(flash<7)

Sleep(1000);

point(38,ty+4);

cout<<" ";

point(0,15);

Sleep(1000);

if(mode==1)

{

if(p1>p2)

{

point(46,ty+4);

cout<<"You Won the Series..!";

}

else if(p2>p1)

{

point(38,ty+4);

cout<<"Computer Won the Series..!";

}

else

{

point(51,ty+4);

cout<<"TIE....!";

}

}

else if(mode==2)

{

if(p1>p2)

{

point(38,ty+4);

cout<<"Player 1 Won the Series..!";

}

else if(p2>p1)

{

point(38,ty+4);

cout<<"Player 2 won the Series..!";

}

else

{

point(51,ty+4);

cout<<"TIE....!";

}

}

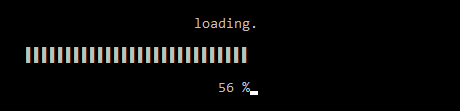
flash--;

}

}

**13)** **Working on Extra stuffs:**

**a)** **creating a fake loading page(to make the game more realistic):**



void load()

{

system("cls");

point(45,14);

cout<<"loading";

point(48,18);

cout<<"0 %";

point(52,14);

delay("...",600);

Sleep(200);

point(52,14);

cout<<" ";

int tempx=24,tempx2=52,temp3=1,t=2000;

for(int i=1;i<=50;i++)

{

if(i<4)

{

point(tempx2++,14);

cout<<".";

}

else

{

if(i%6==0)

{

point(52,14);

cout<<" ";

if(temp3==1)

{

point(52,14);

cout<<". ";

}

else if(temp3==2)

{

point(52,14);

cout<<".. ";

}

else

{

point(52,14);

cout<<"...";

temp3=0;

}

temp3++;

}

}

point(tempx++,16);

cout<<(char)221;

point(48,18);

cout<<2\*i<<" %";

Sleep(t);

if(t>1000)

t=t-300;

else if(t>200)

t=t-80;

else if(t>100)

t=t-10;

else

t=100;

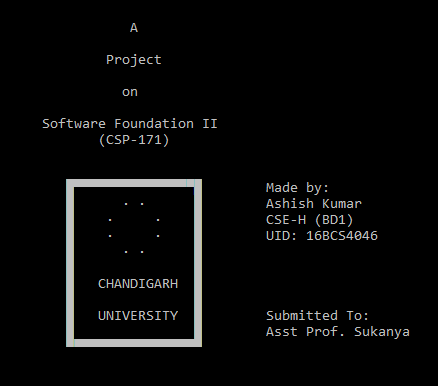
}

system("cls");

}

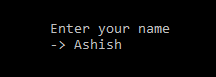
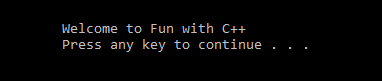
**b)** **Making of front page**

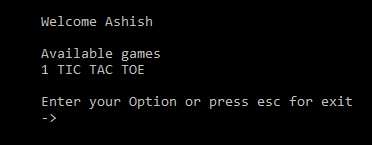
*\*code is not shown to save space.*

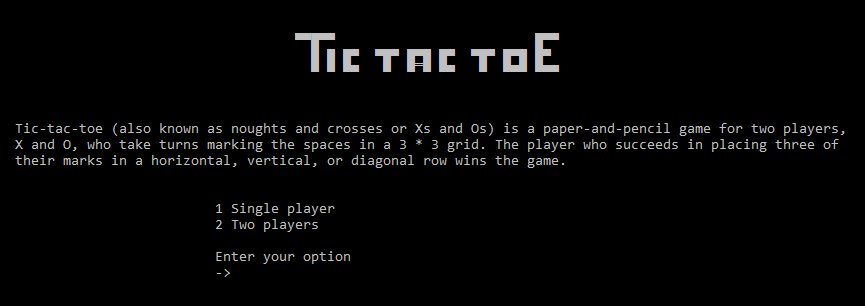


**c)** **Other pages:**

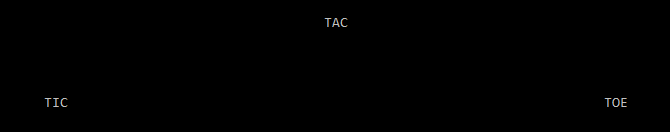
*\*code is not shown to save space.*







**d)** **Floating name – TIC(left) TAC(top) TOE(right)**



tictactoe r;

int fx=0,fy=7;

int d=53,e=0;

for(fx=0;fx<47;fx++)

{

r.point(fx,fy);

cout<<"TIC";

if(e%6==0)

{

r.point(51,e/6-1);

cout<<" ";

r.point(51,e/6);

cout<<"TAC";

}

if(d>6)

{

r.point(49+d,fy);

cout<<"TOE";

r.point(49+d+3,fy);

cout<<" ";

}

r.point(fx-1,fy);

cout<<" ";

r.point(0,0);

Sleep(100);

d--;

e++;

}

**e)** **Exiting:**

C:\Users\ashish\AppData\Local\Microsoft\Windows\INetCache\Content.Word\exit.pngC:\Users\ashish\AppData\Local\Microsoft\Windows\INetCache\Content.Word\see_you_soon.png

**Conclusion:**

The game of Tic Tac Toe was made using C++ programming language.