



Tic Tac Toe

Computer Science Project

Kartik Singhal

XI - A

Contents

- ✓ *Certificate*
- ✓ *Acknowledgments*
- ✓ *Source Code*
- ✓ *Bibliography*

Certificate

*This is to certify that **Kartik Singhal** of Class **XI A** has completed his project under my guidance and supervision. He has worked successfully under my provision and has shown utmost sincerity in completion of this project.*

Mrs. Bindu Gupta
(Computer Sc. Teacher)

Acknowledgements

*I want to thank my Computer Science teacher **Mrs. Bindu Gupta** for her constant encouragement and support and for her interest in the project.*

*My thanks are also due to **Mr. Manoj Kumar**, our Lab. Assistant for his support in the laboratory.*


```
//Initialization of graphics
initgraph(&driver,&mode,"c:\\tc\\bgi");

//Store screensize in terms of x and y
maxx=getmaxx();
maxy=getmaxy();

setlinestyle(0,0,3);

/*Storing images in memory :start*/
//Green Circle
setcolor(2);
circle(maxx/2+150,maxy/2,55);
areao=imagesize(maxx/2+90,maxy/2-
60,maxx/2+210,maxy/2+60);
buffo=(char*)malloc(areao);
getimage(maxx/2+90,maxy/2-60,maxx/2+210,maxy/2+60,buffo);

//Lightblue Cross
setcolor(9);
line(maxx/2-49,maxy/2-49,maxx/2+49,maxy/2+49);
line(maxx/2-49,maxy/2+49,maxx/2+49,maxy/2-49);
areax=imagesize(maxx/2-50,maxy/2-50,maxx/2+50,maxy/2+50);
buffx=(char*)malloc(areax);
getimage(maxx/2-50,maxy/2-50,maxx/2+50,maxy/2+50,buffx);

//Red Tick Mark
setcolor(RED);
line(maxx/2,maxy/2+110,maxx/2+10,maxy/2+120);
line(maxx/2+10,maxy/2+120,maxx/2+40,maxy/2+100);
areach=imagesize(maxx/2-
1,maxy/2+99,maxx/2+41,maxy/2+121);
buffch=(char*)malloc(areach);
getimage(maxx/2-1,maxy/2+99,maxx/2+41,maxy/2+121,buffch);
/*Storing images in memory :end*/

cleardevice();
start();
mainscreen();
}

//Shows the first screen
void start(void)
{
    setcolor(15);
    settextstyle(7,0,9);

    setfillpattern(pattern,1);
```

```

    bar(0,0,maxx,maxy);
    outtextxy(maxx/2-(textwidth("Tic Tac Toe")/2),maxy/2-
(textheight("Tic Tac Toe")/2)-100,"Tic Tac Toe");

    settextstyle(2,HORIZ_DIR,7);
    outtextxy(maxx-textwidth("Developed by:")+30,maxy-
7*textheight("Developed by:"),"Developed by:");
    outtextxy(maxx-textwidth("Kartik Singhal")+30,maxy-
5.5*textheight("Kartik Singhal"),"Kartik Singhal");
    outtextxy(maxx-textwidth("XI A")+30,maxy-4*textheight("XI
A"),"XI A");
    outtextxy(maxx-textwidth("DAV Public School")+30,maxy-
2.5*textheight("DAV Public School"),"DAV Public School");
    boundary();

    settextstyle(0,HORIZ_DIR,1);
    music(3);
}

//Makes boundary over the current screen
void boundary(void)
{
    setcolor(6);
    rectangle(0,0,maxx,maxy);
    setcolor(15);
}

//Shows the main menu
void mainscreen(void)
{
    cleardevice();

    setfillpattern(pattern,1);
    bar(0,0,maxx,maxy);

    setcolor(15);
    settextstyle(7,0,9);
    outtextxy(maxx/2-(textwidth("Tic Tac Toe")/2),maxy/2-
(textheight("Tic Tac Toe")/2)-150,"Tic Tac Toe");
    boundary();

    settextstyle(3,HORIZ_DIR,5);
    outtextxy(maxx/2-100,maxy/2-textheight("P"),"Play Game");
    outtextxy(maxx/2-
100,maxy/2+textheight("I"),"Instructions");
    outtextxy(maxx/2-100,maxy/2+textheight("E")*3,"Exit");

    y=maxy/2-textheight("P")/2;

```

```
putimage(maxx/2-160,y,buffch,XOR_PUT);

//To move the red tick mark to choose the main options
while(1)
{
    if(kbhit())
    {
        putimage(maxx/2-160,y,buffch,XOR_PUT);
        delay(0);

        ch=getch();
        if(ch==80 || ch=='s' || ch=='S')
            y+=2*textheight("I");
        else if(ch==72 || ch=='w' || ch=='W')
            y-=2*textheight("I");
        else if(ch==27)
        {
            closegraph();
            exit(0);
        }
        else if(ch=='\r' || ch==' ')
        {
            if(y>maxy/2-textheight("P") && y<maxy/2)
            {
                music(1);
                game();
            }
            else if(y>maxy/2 &&
y<maxy/2+textheight("E")*3)
            {
                music(1);
                instruction();
            }
            else if(y>maxy/2+textheight("E")*3)
            {
                music(1);
                closegraph();
                exit(0);
            }
        }
        music(5);
        if(y<maxy/2-textheight("P"))
            y+=2*textheight("I");
        else if(y>maxy/2+textheight("E")*4)
            y-=2*textheight("I");
        putimage(maxx/2-160,y,buffch,XOR_PUT);
        delay(0);
    }
}
```



```
        }  
    }  
}  
  
//Shows the instruction screen  
void instruction(void)  
{  
    cleardevice();  
  
    setfillpattern(pattern,1);  
    bar(0,0,maxx,maxy);  
  
    setcolor(15);  
    settextstyle(7,0,8);  
    outtextxy(maxx/2-(textwidth("Instructions")/2),maxy/2-  
(textheight("Instructions")/2)-150,"Instructions");  
    boundary();  
    settextstyle(3,HORIZ_DIR,3);  
    outtextxy(50,maxy/2-50,"This 2 player game is a  
computerised version");  
    outtextxy(50,maxy/2,"of the TIC TAC TOE game you play  
using pen");  
    outtextxy(50,maxy/2+50,"and paper. Use arrow keys or 'w',  
'a', 's' &");  
    outtextxy(50,maxy/2+100,"'d' keys to place the cross or  
zero to desired");  
    outtextxy(50,maxy/2+150,"place. Use 'Enter' or 'Space'  
key to select place.");  
  
    getch();  
    music(1);  
    mainscreen();  
}  
  
//Actually begins the game and controls it  
void game(void)  
{  
    //Reinitialisation of variables  
    n=4;  
    for(i=0;i<3;i++)  
        for(j=0;j<3;j++)  
            mat[i][j]=' '  
  
    cleardevice();  
    boundary();  
}
```

```

    settextstyle(3,0,4);
    setcolor(15);
    outtextxy(maxx/2-textwidth("Player 1! Choose your
option:)/2,maxy/2-150,"Player 1! Choose your option:");
    putimage(maxx/2-200,maxy/2,buffer,XOR_PUT);
    putimage(maxx/2+100,maxy/2-10,buffer,XOR_PUT);
    x=maxx/2-170;
    putimage(x,maxy/2+130,buffer,XOR_PUT);

    //To move the red tick mark which allows to choose
between X & O
    while(1)
    {
        if(kbhit())
        {
            putimage(x,maxy/2+130,buffer,XOR_PUT);
            delay(0);
            ch=getch();
            if(ch==75 || ch=='a' || ch=='A')
                x=maxx/2-170;
            else if(ch==77 || ch=='d' || ch=='D')
                x=maxx/2+140;
            else if(ch==27)
            {
                music(1);
                mainscreen();
            }
            else if(ch=='\r' || ch==' ')
            {
                music(1);
                break;
            }
            music(5);
            putimage(x,maxy/2+130,buffer,XOR_PUT);
            delay(0);
        }
    }
    putimage(x,maxy/2+130,buffer,XOR_PUT);

    if(x<maxx/2)
        xoro='x';
    else xoro='o';

    cleardevice();

    //Makes the grid that makes up the board used to play the
game

```

```
line(maxx/2-75,20,maxx/2-75,maxy-10);
line(maxx/2+75,20,maxx/2+75,maxy-10);
line(maxx/2-225,maxy/2-75,maxx/2+225,maxy/2-75);
line(maxx/2-225,maxy/2+75,maxx/2+225,maxy/2+75);

boundary();
delay(500);
music(2);
delay(200);
if(xoro=='x')
{
    putx();
    do
    {
        puto();
        putx();
    }while(--n);
}
else if(xoro=='o')
{
    puto();
    do
    {
        putx();
        puto();
    }while(--n);
}
matchdraw();
}

//Draws and allows movement of 0
void puto(void)
{
    x=10;
    y=350;
    music(1);
    putimage(x,y,buffo,XOR_PUT);
    while(1)
    {
        if(kbhit())
        {
            putimage(x,y,buffo,XOR_PUT);
            delay(0);
            ch=getch();
            if(ch==77 || ch=='d' || ch=='D')
                x+=5;
            else if(ch==75 || ch=='a' || ch=='A')
```

```

        x-=5;
    else if(ch==72 || ch=='w' || ch=='W')
        y-=5;
    else if(ch==80 || ch=='s' || ch=='S')
        y+=5;
    else if(ch==27)
    {
        music(1);
        mainscreen();
    }
    else if(ch=='\r' || ch==' ')
    {
        if(x+60<=maxx/2-75 && y+60<=maxy/2-75 &&
mat[0][0]==' ')
        {
            putimage(maxx/2-208,maxy/2-
210,buffer,XOR_PUT);
            music(4);
            delay(500);
            mat[0][0]='o';
            if(((mat[0][0]==mat[1][1]) &&
(mat[0][0]==mat[2][2])) || ((mat[0][0]==mat[0][1]) &&
(mat[0][0]==mat[0][2])) || ((mat[0][0]==mat[1][0]) &&
(mat[0][0]==mat[2][0])))
                win('o');
        }
        else if(x+60>=maxx/2+75 && y+60<=maxy/2-
75 && mat[0][2]==' ')
        {
            putimage(maxx/2+93,maxy/2-
210,buffer,XOR_PUT);
            music(4);
            delay(500);
            mat[0][2]='o';
            if(((mat[0][2]==mat[0][0]) &&
(mat[0][2]==mat[0][1])) || ((mat[0][2]==mat[1][2]) &&
(mat[0][2]==mat[2][2])) || ((mat[0][2]==mat[1][1]) &&
(mat[0][2]==mat[2][0])))
                win('o');
        }
        else if(x+60>=maxx/2-75 &&
x+60<=maxx/2+75 && y+60<=maxy/2-75 && mat[0][1]==' ')
        {
            putimage(maxx/2-60,maxy/2-
210,buffer,XOR_PUT);
            music(4);
            delay(500);
            mat[0][1]='o';

```

```

                                if(((mat[0][1]==mat[0][0]) &&
(mat[0][1]==mat[0][2])) || ((mat[0][1]==mat[1][1]) &&
(mat[0][1]==mat[2][1])))
                                win('o');
                                }
                                else if(x+60<=maxx/2-75 &&
y+60>=maxy/2+75 && mat[2][0]==' ')
                                {
                                putimage(maxx/2-
208,maxy/2+90,buffer,XOR_PUT);
                                music(4);
                                delay(500);
                                mat[2][0]='o';
                                if(((mat[2][0]==mat[0][0]) &&
(mat[2][0]==mat[1][0])) || ((mat[2][0]==mat[1][1]) &&
(mat[2][0]==mat[0][2])) || ((mat[2][0]==mat[2][1]) &&
(mat[2][0]==mat[2][2])))
                                win('o');
                                }
                                else if(x+60>=maxx/2+75 &&
y+60>=maxy/2+75 && mat[2][2]==' ')
                                {
                                putimage(maxx/2+93,maxy/2+90,buffer,XOR_PUT);
                                music(4);
                                delay(500);
                                mat[2][2]='o';
                                if(((mat[2][2]==mat[0][0]) &&
(mat[2][2]==mat[1][1])) || ((mat[2][2]==mat[2][0]) &&
(mat[2][2]==mat[2][1])) || ((mat[2][2]==mat[0][2]) &&
(mat[2][2]==mat[1][2])))
                                win('o');
                                }
                                else if(x+60>=maxx/2-75 &&
x+60<=maxx/2+75 && y+60>=maxy/2+75 && mat[2][1]==' ')
                                {
                                putimage(maxx/2-
60,maxy/2+90,buffer,XOR_PUT);
                                music(4);
                                delay(500);
                                mat[2][1]='o';
                                if(((mat[2][1]==mat[0][1]) &&
(mat[1][1]==mat[2][1])) || ((mat[2][1]==mat[2][0]) &&
(mat[2][1]==mat[2][2])))
                                win('o');
                                }

```

```

else if(x+60<=maxx/2-75 && y+60>=maxy/2-
75 && y+60<=maxy/2+75 && mat[1][0]==' ')
{
    putimage(maxx/2-208,maxy/2-
60,buffer,XOR_PUT);
    music(4);
    delay(500);
    mat[1][0]='o';
    if(((mat[1][0]==mat[1][1]) &&
(mat[1][0]==mat[1][2])) || ((mat[1][0]==mat[0][0]) &&
(mat[1][0]==mat[2][0])))
        win('o');
}
else if(x+60>=maxx/2+75 && y+60>=maxy/2-
75 && y+60<=maxy/2+75 && mat[1][2]==' ')
{
    putimage(maxx/2+93,maxy/2-
60,buffer,XOR_PUT);
    music(4);
    delay(500);
    mat[1][2]='o';
    if(((mat[1][2]==mat[0][2]) &&
(mat[1][2]==mat[2][2])) || ((mat[1][2]==mat[1][0]) &&
(mat[1][2]==mat[1][1])))
        win('o');
}
else if(x+60>=maxx/2-75 &&
x+60<=maxx/2+75 && y+60>=maxy/2-75 && y+60<=maxy/2+75 &&
mat[1][1]==' ')
{
    putimage(maxx/2-60,maxy/2-
60,buffer,XOR_PUT);
    music(4);
    delay(500);
    mat[1][1]='o';
    if(((mat[1][1]==mat[0][0]) &&
(mat[1][1]==mat[2][2])) || ((mat[1][1]==mat[0][1]) &&
(mat[1][1]==mat[2][1])) || ((mat[1][1]==mat[1][0]) &&
(mat[1][1]==mat[1][2])) || ((mat[1][1]==mat[2][0]) &&
(mat[1][1]==mat[0][2])))
        win('o');
}
else
{
    putimage(x,y,buffer,XOR_PUT);
    continue;
}
break;

```

```

    }

    //Checking the boundaries of the board for 0
    if(x<=maxx/2-225)
        x+=5;
    else if(x>=maxx/2+110)
        x-=5;
    else if(y<=maxy/2-220)
        y+=5;
    else if(y>=maxy/2+115)
        y-=5;
    putimage(x,y,buffo,XOR_PUT);
    delay(0);
}
}

//Draws and allows movement of X
void putx(void)
{
    x=10;
    y=350;
    music(1);
    putimage(x,y,buffx,XOR_PUT);
    while(1)
    {
        if(kbhit())
        {
            putimage(x,y,buffx,XOR_PUT);
            delay(0);
            ch=getch();
            if(ch==77 || ch=='d' || ch=='D')
                x+=5;
            else if(ch==75 || ch=='a' || ch=='A')
                x-=5;
            else if(ch==72 || ch=='w' || ch=='W')
                y-=5;
            else if(ch==80 || ch=='s' || ch=='S')
                y+=5;
            else if(ch==27)
            {
                music(1);
                mainscreen();
            }
            else if(ch=='\r' || ch==' ')
            {

```

```

mat[0][0]==' ')
    if(x+50<=maxx/2-75 && y+50<=maxy/2-75 &&
    {
        putimage(maxx/2-200,maxy/2-
200,buffer,XOR_PUT);
        music(4);
        delay(500);
        mat[0][0]='x';
        if(((mat[0][0]==mat[1][1]) &&
(mat[0][0]==mat[2][2])) || ((mat[0][0]==mat[0][1]) &&
(mat[0][0]==mat[0][2])) || ((mat[0][0]==mat[1][0]) &&
(mat[0][0]==mat[2][0])))
            win('x');
    }
    else if(x+50>=maxx/2+75 && y+50<=maxy/2-
75 && mat[0][2]==' ')
    {
        putimage(maxx/2+102,maxy/2-
200,buffer,XOR_PUT);
        music(4);
        delay(500);
        mat[0][2]='x';
        if(((mat[0][2]==mat[0][0]) &&
(mat[0][2]==mat[0][1])) || ((mat[0][2]==mat[1][2]) &&
(mat[0][2]==mat[2][2])) || ((mat[0][2]==mat[1][1]) &&
(mat[0][2]==mat[2][0])))
            win('x');
    }
    else if(x+50>=maxx/2-75 &&
x+50<=maxx/2+75 && y+50<=maxy/2-75 && mat[0][1]==' ')
    {
        putimage(maxx/2-50,maxy/2-
200,buffer,XOR_PUT);
        music(4);
        delay(500);
        mat[0][1]='x';
        if(((mat[0][1]==mat[0][0]) &&
(mat[0][1]==mat[0][2])) || ((mat[0][1]==mat[1][1]) &&
(mat[0][1]==mat[2][1])))
            win('x');
    }
    else if(x+50<=maxx/2-75 &&
y+50>=maxy/2+75 && mat[2][0]==' ')
    {
        putimage(maxx/2-
200,maxy/2+100,buffer,XOR_PUT);
        music(4);
        delay(500);

```



```

        mat[2][0]='x';
        if(((mat[2][0]==mat[0][0]) &&
(mat[2][0]==mat[1][0])) || ((mat[2][0]==mat[1][1]) &&
(mat[2][0]==mat[0][2])) || ((mat[2][0]==mat[2][1]) &&
(mat[2][0]==mat[2][2])))
            win('x');
    }
    else if(x+50>=maxx/2+75 &&
y+50>=maxy/2+75 && mat[2][2]==' ')
    {
        putimage(maxx/2+102,maxy/2+100,buffx,XOR_PUT);
        music(4);
        delay(500);
        mat[2][2]='x';
        if(((mat[2][2]==mat[0][0]) &&
(mat[2][2]==mat[1][1])) || ((mat[2][2]==mat[2][0]) &&
(mat[2][2]==mat[2][1])) || ((mat[2][2]==mat[0][2]) &&
(mat[2][2]==mat[1][2])))
            win('x');
    }
    else if(x+50>=maxx/2-75 &&
x+50<=maxx/2+75 && y+50>=maxy/2+75 && mat[2][1]==' ')
    {
        putimage(maxx/2-
50,maxy/2+100,buffx,XOR_PUT);
        music(4);
        delay(500);
        mat[2][1]='x';
        if(((mat[2][1]==mat[0][1]) &&
(mat[1][1]==mat[2][1])) || ((mat[2][1]==mat[2][0]) &&
(mat[2][1]==mat[2][2])))
            win('x');
    }
    else if(x+50<=maxx/2-75 && y+50>=maxy/2-
75 && y+50<=maxy/2+75 && mat[1][0]==' ')
    {
        putimage(maxx/2-200,maxy/2-
50,buffx,XOR_PUT);
        music(4);
        delay(500);
        mat[1][0]='x';
        if(((mat[1][0]==mat[1][1]) &&
(mat[1][0]==mat[1][2])) || ((mat[1][0]==mat[0][0]) &&
(mat[1][0]==mat[2][0])))
            win('x');
    }
}

```

```

else if(x+50>=maxx/2+75 && y+50>=maxy/2-
75 && y+50<=maxy/2+75 && mat[1][2]==' ')
{
    putimage(maxx/2+102,maxy/2-
50,buffx,XOR_PUT);
    music(4);
    delay(500);
    mat[1][2]='x';
    if(((mat[1][2]==mat[0][2]) &&
(mat[1][2]==mat[2][2])) || ((mat[1][2]==mat[1][0]) &&
(mat[1][2]==mat[1][1])))
        win('x');
}
else if(x+50>=maxx/2-75 &&
x+50<=maxx/2+75 && y+50>=maxy/2-75 && y+50<=maxy/2+75 &&
mat[1][1]==' ')
{
    putimage(maxx/2-50,maxy/2-
50,buffx,XOR_PUT);
    music(4);
    delay(500);
    mat[1][1]='x';
    if(((mat[1][1]==mat[0][0]) &&
(mat[1][1]==mat[2][2])) || ((mat[1][1]==mat[0][1]) &&
(mat[1][1]==mat[2][1])) || ((mat[1][1]==mat[1][0]) &&
(mat[1][1]==mat[1][2])) || ((mat[1][1]==mat[2][0]) &&
(mat[1][1]==mat[0][2])))
        win('x');
}
else
{
    putimage(x,y,buffx,XOR_PUT);
    continue;
}
break;
}

//Checking the boundaries of the board for X
if(x<=maxx/2-225)
    x+=5;
else if(x>=maxx/2+125)
    x-=5;
else if(y<=maxy/2-220)
    y+=5;
else if(y>=maxy/2+125)
    y-=5;
putimage(x,y,buffx,XOR_PUT);
delay(0);

```

```

    }
}

//Shows 'matchdraw' message in case no player wins
void matchdraw(void)
{
    setfillstyle(8,8);
    bar(maxx/2-150,maxy/2-100,maxx/2+150,maxy/2+100);
    setcolor(4);
    rectangle(maxx/2-150,maxy/2-100,maxx/2+150,maxy/2+100);
    outtextxy(maxx/2-textwidth("Game Drawn!!")/2,maxy/2-
textheight("Game Drawn!!")+10,"Game Drawn!!");
    music(3);
    setfillstyle(1,15);
    delay(500);
    music(1);
    mainscreen();
}

//Checks which player wins the game
void win(char won)
{
    wong=won;
    if(won=='o' && xoro=='o')
        winp[0]='1';
    else if(won=='o' && xoro=='x')
        winp[0]='2';
    else if(won=='x' && xoro=='o')
        winp[0]='2';
    else if(won=='x' && xoro=='x')
        winp[0]='1';
    winp[1]='\0';
    result();
}

//Shows the result of the match
void result(void)
{
    setfillstyle(8,8);
    bar(maxx/2-150,maxy/2-100,maxx/2+150,maxy/2+100);
    if(wong=='o')
        setcolor(2);
    else setcolor(9);
    rectangle(maxx/2-150,maxy/2-100,maxx/2+150,maxy/2+100);
    outtextxy(maxx/2-textwidth("Player ")/2,maxy/2-50,"Player
");
    outtextxy(maxx/2+textwidth("Player ")/2,maxy/2-50,winp);
}

```

```
    outtextxy(maxx/2-textwidth("won the
game!!")/2,maxy/2,"won the game!!");
    music(3);
    setfillstyle(1,15);
    delay(500);
    music(1);
    mainscreen();
}

//Plays various music used in the game
void music ( int type )
{
    /* natural frequencies of 7 notes */
    float octave[7] = { 130.81, 146.83, 164.81, 174.61, 196,
220, 246.94 } ;
    int mn, mi ;

    switch ( type )
    {
        case 1 :
            for ( mi = 0 ; mi < 7 ; mi++ )
            {
                sound ( octave[mi] * 8 ) ;
                delay ( 30 ) ;
            }
            nosound() ;
            break ;

        case 2 :
            for ( mi = 0 ; mi < 15 ; mi++ )
            {
                mn = random ( 7 ) ;
                sound ( octave[mn] * 4 ) ;
                delay ( 100 ) ;
            }
            nosound() ;
            break ;

        case 3 :
            while ( !kbhit() )
            {
                mn = random ( 7 ) ;
                sound ( octave[mn] * 4 ) ;
                delay ( 100 ) ;
            }
            nosound() ;

            /* flush the keyboard buffer */
    }
}
```


Bibliography

- *Graphics Under C*, by Yashavant Kanetkar
- *Computer Science A Textbook for Class XI*, by Sumita Arora