**REPORT ON TIC TAC TOE GAME IN C**

PREPARED AND SUBMITTED BY

**Pravesh Bisaria(2017Btechcse107)**

**Hitika Ailani(2017Btechcse212)**

**Yashi Saini(2017Btechcse221)**

**Viral Natani(2017Btechcse307)**

FACULTY GUIDE

**Devendra Bhavasar**

**Associate Professor**



**Department of Computer Science**

**Institute of Engineering and Technology (IET)**

**JK Lakshmipat University, Jaipur**

**Table of Content**

1. **Objective**
2. **Overview**
3. **Core Logic-Humans**
4. **Functions Used**
5. **Libraries Used**
6. **Limitation**
7. **Future Plans**
8. **Game Output Screenshots**
9. **Source Code**
10. **References**
11. **Objectives:**

Our project name is Tic-Tac-Toe game. This game is very popular and is fairly simple by itself. It is actually a two player game. In this game, there is a board with *n* x *n* squares. In our game, it is 3 x 3 squares.

The goal of Tic-Tac-Toe is to be one of the players to get three same

symbols in a row - horizontally, vertically or diagonally - on a 3 x 3 grid.

1. **Overview:**

This game can be played in a 3x3 grid (shown in the figure 2.1) .The game

can be played by two players. There are two options for players:

1. Human (b) Computer



Figure: 2.1

**2.1 Players:**

For the option human, both the players are human and for the option

computer, the first player is human and the second player is computer.

**2.2 Theory of Game:**

A player can choose between two symbols with his opponent, usual games

use “X”and “O”. If first player choose “X” then the second player have to

play with “O” and vice versa.

A player marks any of the 3x3 squares with his symbol (may be “X” or “O”)

and his aim is to create a straight line horizontally or vertically or diagonally

with two intensions:

a) Create a straight line before his opponent to win the game.

b) Restrict his opponent from creating a straight line first.

In case logically no one can create a straight line with his own symbol, the

game results a tie.

Hence there are only three possible results – a player wins, his opponent

(human or computer) wins or it’s a tie.

**1 2 3**

**4 5 6**

**7 8 9**

Figure: 2.2

If any player is able to draw three Xs or three Os in the following

combinations then that player wins. The combinations are:

a) 1, 2, 3 b) 4, 5, 6

c) 7, 8, 9 d) 1, 4, 7

e) 2, 5, 8 f) 3, 6, 9

h) 1, 5, 9 i) 3, 5, 7

In Tic Tac Toe game, there are 765 [states of space complexities](http://en.wikipedia.org/wiki/State_space_complexity) or over 25,000 possible games on those different positions.

**Rules of the Game**

* The game is to be played between two people (in this program between HUMAN and COMPUTER).
* One of the player chooses ‘O’ and the other ‘X’ to mark their respective cells.
* The game starts with one of the players and the game ends when one of the players has one whole row/ column/ diagonal filled with his/her respective character (‘O’ or ‘X’).
* If no one wins, then the game is said to be draw.

**3. Core Logic - Humans:**

For each move, check whether any 3 combination is occupied by any player

and display the winner accordingly.

**4. Functions Used:**

**void menu()** – In this mini project, this function displays the menu or welcome screen of this project. Scroll down to view the photo of the menu. With this function, you can select whether you wish to play the game with X or with O.

**void go(int n)-**

**assigns values to boxes as X=3 and O=5.**

**void start\_game()**

**void check\_draw() -It checks the condition for draw bu counting the num ber of turns performed and declares a draw if it exceed 9.**

**void draw\_board() –**

**It draws the cross lines.**

**void player\_first()**

**void gotoxy(int x,int y)**-

This is an inbuilt function of C language which takes the pointer to the coordinates X and Y.

**void put\_X\_O(char ch, int pos)** – This function prints one of the character you input into the respective position in Tic-Tac-Toe. For example: if you are playing with X and you input 2, the X will go to first row – second column. If you want to place X in third row – first column, you have to enter 7. And, it is similar for the other positions.

**5. Libraries Used:**

* **Stdio.h**: The stdio.h header defines three variable types, several macros, and various functions for performing input and output.
* **Conio.h:** This header declares several useful library functions for performing "console input and output" from a program. Most C compilers that target [DOS](https://en.wikipedia.org/wiki/DOS), [Windows 3.x](https://en.wikipedia.org/wiki/Windows_3.x), [Phar Lap](https://en.wikipedia.org/wiki/Phar_Lap_(company)), DOSX, [OS/2](https://en.wikipedia.org/wiki/OS/2), or [Win32](https://en.wikipedia.org/wiki/Win32)[[2]](https://en.wikipedia.org/wiki/Conio.h#cite_note-2) have this header and supply the associated library functions in the default C library. Most C compilers that target [UNIX](https://en.wikipedia.org/wiki/UNIX) and [Linux](https://en.wikipedia.org/wiki/Linux) do not have this header and do not supply the library functions. Some embedded systems or [cc65](https://en.wikipedia.org/wiki/Cc65) use a conio-compatible library.
* **Stdlib.h:** stdlib.h is the header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others. It is compatible with C++ and is known as cstdlib in C++. The name "stdlib" stands for "standard library".
* **Windows.h:** windows.h is a [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows)-specific header file for the [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) programming languages which contains declarations for all of the functions in the [Windows API](https://en.wikipedia.org/wiki/Windows_API), all the common macros used by Windows programmers, and all the data types used by the various functions and subsystems. It defines a very large number of Windows specific functions that can be used in C. The [Win32 API](https://en.wikipedia.org/wiki/Windows_API) can be added to a C programming project by including the <windows.h> header file and linking to the appropriate libraries

**6. Limitations:**

1. GUI is not so attractive.

2. Only mouse interface is implemented, keyboard is not activated in the

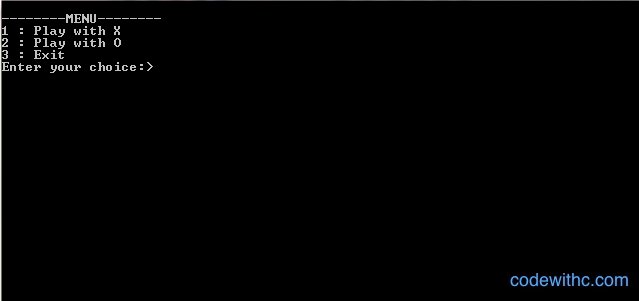
game.

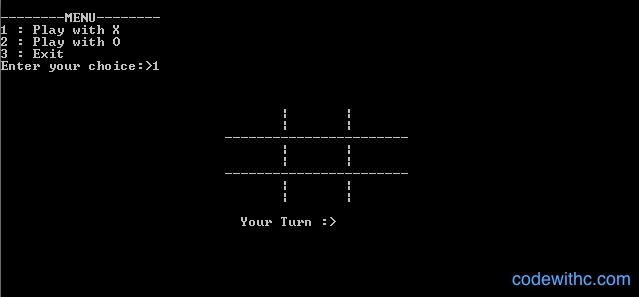
**7. Future plan:**

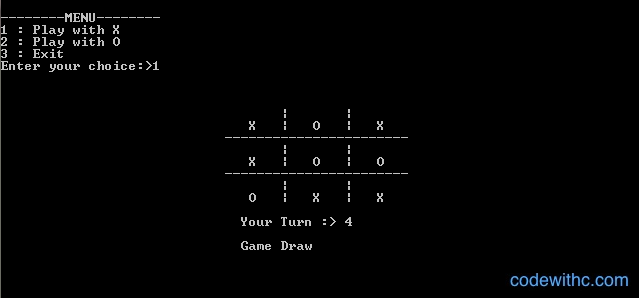
1. Keyboard functions will be added.

2. We want to design more complex boards for the game in future.

**8.Game Output Screenshots**







**9. Source Code**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include <windows.h>

int board[10] = {2,2,2,2,2,2,2,2,2,2};

int turn = 1,flag = 0;

int player,comp;

void menu();

void go(int n);

void start\_game();

void check\_draw();

void draw\_board();

void player\_first();

void put\_X\_O(char ch,int pos);

COORD coord= {0,0}; // this is global variable

//center of axis is set to the top left cornor of the screen

void gotoxy(int x,int y)

{

coord.X=x;

coord.Y=y;

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

}

void main()

{

system("cls");

menu();

getch();

}

void menu()

{

int choice;

system("cls");

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

printf("\n1 : Play with X");

printf("\n2 : Play with O");

printf("\n3 : Exit");

printf("\nEnter your choice:>");

scanf("%d",&choice);

turn = 1;

switch (choice)

{

case 1:

player = 1;

comp = 0;

player\_first();

break;

case 2:

player = 0;

comp = 1;

start\_game();

break;

case 3:

exit(1);

default:

menu();

}

}

int make2()

{

if(board[5] == 2)

return 5;

if(board[2] == 2)

return 2;

if(board[4] == 2)

return 4;

if(board[6] == 2)

return 6;

if(board[8] == 2)

return 8;

return 0;

}

int make4()

{

if(board[1] == 2)

return 1;

if(board[3] == 2)

return 3;

if(board[7] == 2)

return 7;

if(board[9] == 2)

return 9;

return 0;

}

int posswin(int p)

{

// p==1 then X p==0 then O

int i;

int check\_val,pos;

if(p == 1)

check\_val = 18;

else

check\_val = 50;

i = 1;

while(i<=9)//row check

{

if(board[i] \* board[i+1] \* board[i+2] == check\_val)

{

if(board[i] == 2)

return i;

if(board[i+1] == 2)

return i+1;

if(board[i+2] == 2)

return i+2;

}

i+=3;

}

i = 1;

while(i<=3)//column check

{

if(board[i] \* board[i+3] \* board[i+6] == check\_val)

{

if(board[i] == 2)

return i;

if(board[i+3] == 2)

return i+3;

if(board[i+6] == 2)

return i+6;

}

i++;

}

if(board[1] \* board[5] \* board[9] == check\_val)

{

if(board[1] == 2)

return 1;

if(board[5] == 2)

return 5;

if(board[9] == 2)

return 9;

}

if(board[3] \* board[5] \* board[7] == check\_val)

{

if(board[3] == 2)

return 3;

if(board[5] == 2)

return 5;

if(board[7] == 2)

return 7;

}

return 0;

}

void go(int n)

{

if(turn % 2)

board[n] = 3;

else

board[n] = 5;

turn++;

}

void player\_first()

{

int pos;

check\_draw();

draw\_board();

gotoxy(30,18);

printf("Your Turn :> ");

scanf("%d",&pos);

if(board[pos] != 2)

player\_first();

if(pos == posswin(player))

{

go(pos);

draw\_board();

gotoxy(30,20);

//textcolor(128+RED);

printf("Player Wins");

getch();

exit(0);

}

go(pos);

draw\_board();

start\_game();

}

void start\_game()

{

// p==1 then X p==0 then O

if(posswin(comp))

{

go(posswin(comp));

flag = 1;

}

else if(posswin(player))

go(posswin(player));

else if(make2())

go(make2());

else

go(make4());

draw\_board();

if(flag)

{

gotoxy(30,20);

//textcolor(128+RED);

printf("Computer wins");

getch();

}

else

player\_first();

}

void check\_draw()

{

if(turn > 9)

{

gotoxy(30,20);

//textcolor(128+RED);

printf("Game Draw");

getch();

exit(0);

}

}

void draw\_board()

{

int j;

for(j=9; j<17; j++)

{

gotoxy(35,j);

printf("| |");

}

gotoxy(28,11);

printf("-----------------------");

gotoxy(28,14);

printf("-----------------------");

for(j=1; j<10; j++)

{

if(board[j] == 3)

put\_X\_O('X',j);

else if(board[j] == 5)

put\_X\_O('O',j);

}

}

void put\_X\_O(char ch,int pos)

{

int m;

int x = 31, y = 10;

m = pos;

if(m > 3)

{

while(m > 3)

{

y += 3;

m -= 3;

}

}

if(pos % 3 == 0)

x += 16;

else

{

pos %= 3;

pos--;

while(pos)

{

x+=8;

pos--;

}

}

gotoxy(x,y);

printf("%c",ch);

}

**Reference:**

**Book:**

* Let Us C by Yashavant Kanetkar
* <https://www.codewithc.com/mini-project-in-c-tic-tac-toe-game/>
* <https://www.geeksforgeeks.org/implementation-of-tic-tac-toe-game/>
* <https://en.wikipedia.org/>