Mohd. Aftab Alam
02013302717

TIC–TAC–TOE

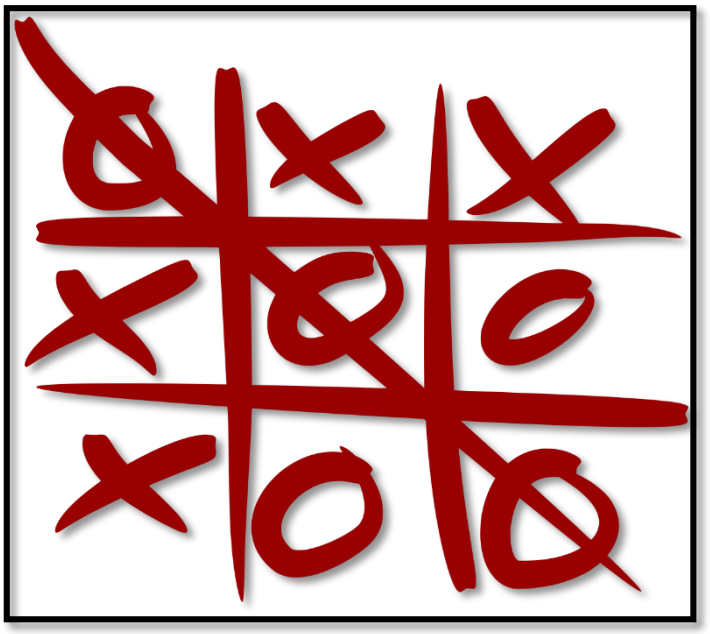
Aim :

Program To Implement Tic-Tac-Toe Game Using HTML, CSS And JavaScript.

Theory :

Tic-Tac-Toe (American English), **Noughts and Crosses** (British English), or **Xs and Os** is a paper and pencil game for two players, **X and O,** who take turns marking the spaces in a **3×3** grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

Players soon discover that the best play from both parties leads to a draw. Hence, tic-tac-toe is most often played by young children, who often have not yet discovered the optimal strategy.



Because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence that deals with the searching of game trees.

The game can be generalized to an **m, n, k-game** in which two players alternate placing stones of their own colour on an **m×n board,** with the goal of getting **k** of their own colour in a row. Tic-tac-toe is the **(3, 3, 3)-game. Harary's** generalized tic-tac-toe is an even broader generalization of tic-tac-toe. It can also be generalized as a $\mathbf{n^d}$ game. Tic-tac-toe is the game where **n equals 3** and **d equals 2.** If played optimally by both players, the game always ends in a draw, making tic-tac-toe a futile game.

Combinatorics

When considering only the state of the board, and after taking into account board symmetries, there are only 138 terminal board positions. A combinatorics study of the game shows that when **"X"** makes the first move every time, the game is won as follows :

- **91** distinct positions are won by **(X)**
- **44** distinct positions are won by **(O)**
- **3** distinct positions are **drawn**

Mohd. Aftab Alam
02013302717

Strategy

A player can play a perfect game of tic-tac-toe (to win or at least, draw) if each time it is their turn to play, they choose the first available move from the following list, as used in **Newell and Simon's 1972** tic-tac-toe program.

1. Win : If the player has two in a row, they can place a third to get three in a row.

2. Block : If the opponent has two in a row, the player must play the third themselves to block the opponent.

3. Fork : Create an opportunity where the player has two ways to win (two non-blocked lines of 2).

4. Blocking An Opponent's Fork : If there is only one possible fork for the opponent, the player should block it. Otherwise, the player should block all forks in any way that simultaneously allows them to create two in a row. Otherwise, the player should create a two in a row to force the opponent into defending, as long as it doesn't result in them creating a fork.

5. Centre : A player marks the centre. (If it is the first move of the game, playing on a corner gives the second player more opportunities to make a mistake and may therefore be the better choice; however, it makes no difference between perfect players.)

6. Opposite Corner : If the opponent is in the corner, the player plays the opposite corner.

7. Empty Corner : The player plays in a corner square.

8. Empty Side : The player plays in a middle square on any of the 4 sides.

The **first player,** who shall be designated **"X",** has 3 possible positions to mark during the first turn. Superficially, it might seem that there are 9 possible positions, corresponding to the 9 squares in the grid. However, by rotating the board, we will find that in the first turn, every corner mark is strategically equivalent to every other corner mark. The same is true of every edge (side middle) mark. For strategy purposes, there are therefore only three possible first marks: corner, edge, or centre. Player X can win or force a draw from any of these starting marks; however, playing the corner gives the opponent the smallest choice of squares which must be played to avoid losing. This might suggest that the corner is the best opening move for X, however another study shows that if the players are not perfect, an opening move in the centre is best for X.

The **second player,** who shall be designated **"O",** must respond to X's opening mark in such a way as to avoid the forced win. Player O must always respond to a corner opening with a centre mark, and to a centre opening with a corner mark. An edge opening must be answered either with a centre mark, a corner mark next to the X, or an edge mark opposite the X. Any other responses will allow X to force the win. Once the opening is completed, O's task is to follow the above list of priorities in order to force the draw, or else to gain a win if X makes a weak play.

Mohd. Aftab Alam
02013302717

Code : **tictactoe.html file**

```
<div id = "tic-tac-toe">
<div class = "span3 new_span">
<div class = "row">
<h1 class = "span3">Tic Tac Toe</h1>
<div class = "span3">

<div class = "input-prepend input-append">
<span class = "add-on win_text">O won</span>
<strong id = "o_win" class = "win_times add-on">0</strong>
<span class = "add-on">time(s)</span>
</div>

<div class = "input-prepend input-append">
<span class = "add-on win_text">X won</span>
<strong id = "x_win" class = "win_times add-on">0</strong>
<span class = "add-on">time(s)</span>
</div>

</div>
</div>

<ul class = "row" id = "game">

<li id = "one" class = "btn span1" >+</li>
<li id = "two" class = "btn span1">+</li>
<li id = "three" class = "btn span1">+</li>
<li id = "four" class = "btn span1">+</li>
<li id = "five" class = "btn span1">+</li>
<li id = "six" class = "btn span1">+</li>
<li id = "seven" class = "btn span1">+</li>
<li id = "eight" class = "btn span1">+</li>
<li id = "nine" class = "btn span1">+</li>
</ul>

<div class = "clr"> </div>
<div class = "row"><a href = "#" id = "reset"
class = "btn-success btn span3">Restart</a></div>

</div>
</div>
```

Mohd. Aftab Alam
02013302717

Code : **style.css file**

```
#tic-tac-toe .disable {
    text-transform : uppercase;
    font-size : 30px;
    font-family : Georgia, "Times New Roman", Times, serif }
#tic-tac-toe #game li {
    float : left;
    padding : 0;
    list-style : none;
    text-align : center;
    margin-bottom : 20px;
    color : #fff;
    height : 60px;
    line-height : 60px;
    font-size : 40px;
    color : #ccc }
#tic-tac-toe #game li.disable { color : #fff }
#tic-tac-toe #game { float : left; padding : 0; clear : both }
.new_span { width : 226px }
#tic-tac-toe #reset {
    padding : 5px 10px;
    color : #fff;
    font-family : Arial, Helvetica, sans-serif;
    font-size : 20px;
    clear : both;
    cursor : pointer;
    float : left;
    text-align : center;
    text-transform : uppercase;
    outline : none;
    width : 204px }

.input-prepend span.pre_text { width : 55px }
.input-prepend .span1 { width : 93px }
.input-prepend { margin-bottom : 10px }
.clr { clear : both; height : 0 }

#tic-tac-toe h1 { text-align : center; font-size : 28px }
#tic-tac-toe li::-moz-selection { background : none; color : #000; }
#tic-tac-toe li::-webkit-selection { background : none; color : #000; }
#tic-tac-toe { width : 220px; margin : 0 auto }

.input-append .win_times { background : #fff; width : 101px }
.input-append .win_text { width : 52px }
```

Mohd. Aftab Alam
02013302717

Code : **new.js file**

```
$(document).ready(function() {
    var x = "x"
    var o = "o"
    var count = 0;
    var o_win = 0;
    var x_win = 0;

$('#game li').click(function() {
if($("#one").hasClass('o')&&$("#two").hasClass('o')&&$("#three").
hasClass('o')||$("#four").hasClass('o')&&$("#five").hasClass('o')&&
$("#six").hasClass('o')||$("#seven").hasClass('o')&&$("#eight").
hasClass('o')&&$("#nine").hasClass('o'))
{
    alert('O has won the game. Start a new game')
    $("#game li").text("+");
    $("#game li").removeClass('disable')
    $("#game li").removeClass('o')
    $("#game li").removeClass('x')
    $("#game li").removeClass('btn-primary')
    $("#game li").removeClass('btn-info') }

else if($("#one").hasClass('x')&&$("#two").hasClass('x')&&$("#three").
hasClass('x')||$("#four").hasClass('x')&&$("#five").hasClass('x')&&
$("#six").hasClass('x')||$("#seven").hasClass('x')&&$("#eight").
hasClass('x')&&$("#nine").hasClass('x'))
{
    alert('X wins has won the game. Start a new game')
    $("#game li").text("+");
    $("#game li").removeClass('disable')
    $("#game li").removeClass('o')
    $("#game li").removeClass('x')
    $("#game li").removeClass('btn-primary')
    $("#game li").removeClass('btn-info') }

    else if (count == 9) {
        alert('Its a tie. It will restart.')
        $("#game li").text("+");
        $("#game li").removeClass('disable')
        $("#game li").removeClass('o')
        $("#game li").removeClass('x')
        $("#game li").removeClass('btn-primary')
        $("#game li").removeClass('btn-info')
        count = 0 }
```

Mohd. Aftab Alam
02013302717

```
    else if ($(this).hasClass('disable')) {
        alert('Already selected') }
    else if (count % 2 == 0) {
        count++
        $(this).text(o)
        $(this).addClass('disable o btn-primary')

if($("#one").hasClass('o')&&$("#two").hasClass('o')&&$("#three").
hasClass('o')||$("#four").hasClass('o')&&$("#five").hasClass('o')&&
$("#six").hasClass('o')||$("#seven").hasClass('o')&&$("#eight").
hasClass('o')&&$("#nine").hasClass('o'))
{
    alert('O wins')
    count = 0
    o_win++
    $('#o_win').text(o_win) }
}

else {
    count++
    $(this).text(x)
    $(this).addClass('disable x btn-info')

if($("#one").hasClass('x')&&$("#two").hasClass('x')&&$("#three").
hasClass('x')||$("#four").hasClass('x')&&$("#five").hasClass('x')&&
$("#six").hasClass('x')||$("#seven").hasClass('x')&&$("#eight").
hasClass('x')&&$("#nine").hasClass('x'))
{
    alert('X wins')
    count = 0
    x_win++
    $('#x_win').text(x_win) }
}
} );

$("#reset").click(function () {
    $("#game li").text("+");
    $("#game li").removeClass('disable')
    $("#game li").removeClass('o')
    $("#game li").removeClass('x')
    $("#game li").removeClass('btn-primary')
    $("#game li").removeClass('btn-info')
    count = 0

} );
} );
```

Mohd. Aftab Alam
02013302717

Screen Shots :

Result :

Program For Tic-Tac-Toe Game Using HTML, CSS And JavaScript Was Implemented Successfully.