***TIC TAC TOE***

***A HOUSEHOLD GAME***

*PRESENTED BY:*

1. *Abhishek Sarkar*
2. *Debojyoti Dey*
3. *Sk Shahir Halim*

ALGORITHM

***Some Considerations:-***

1. The grid of tic tac toe game is basically a 3X3 matrix
2. Internally all the cells are initialized by an arbitrary number (say 5) which denotes to be vacant.
3. Cross ‘X’ is internally taken as 1 and Zero ‘0’ by itself.
4. Player’s (1st player in case of multiplayer) input is represented by ‘X’
5. Computer’s (2nd player in case of multiplayer) input is represented by ‘0’.
6. Rest of the rules are as per the basic game rule.

***Player VS Computer:***

1. The player will start the game, hence he can get maximum 5 chances (5 iterations needed).
2. Every time an iteration begins, current situation of the game is printed.
3. Then it is checked whether diagonally/horizontally/vertically there are three consecutive 0’s,

If there is, then computer wins else user is allowed to give an input.

1. Game won’t proceed until valid position of input is given by player.
2. After player input, it is checked whether diagonally/horizontally/vertically there are three consecutive 1’s.
3. If there is, then player wins, else if it was the last chance to the player then it is a Draw Match, & game ENDS!!!
4. If this was 1st chance to player

If the center position is not empty then top left is made 0 by Computer else center is made 0.

Else

If in any linear order there is two 0’s and the remaining one place is empty, it is made zero.

Else if any linear order there is two 1’s and the remaining one place is empty, it is made zero.

Else if it is 2nd iteration & extreme diagonal elements are 1 and center is zero, middle-left is made 0.

Else

If 2nd row/column is totally empty excluding center, left/top position is made 0 respectively.

Else if 2nd row/column is having 0, 1, and an empty space, it is made 0.

1. A complete iteration ends here and repeats same logic until 5 iterations are done or anyone wins.

***Computer VS Player:***

1. Computer will start the game, hence it will get maximum 5 chances (5 iterations needed).
2. If this is the 1st iteration, then either top left or center is made 0 (randomly chosen) by computer.

Else if it 2nd iteration, then either top-left/bottom-right whichever is empty is made 0, else top right is made 0.

Else

If in any linear order there is two 0’s and the remaining one place is empty, it is made zero.

Else if any linear order there is two 1’s and the remaining one place is empty, it is made zero.

Else if it is 3rd iteration, bottom left is made 0.

Else

If 2nd row/column is totally empty excluding center, left/top position is made 0 respectively.

Else if 2nd row/column/minor diagonal is having 0, 1, and an empty space, it is made 0.

1. Current situation of the game is printed.
2. Then it is checked whether diagonally/horizontally/vertically there are three consecutive 0’s,

If there is, then computer wins else if it was 5th iteration, it’s a draw match & GAME ENDS!!!

Else player is allowed to give input.

1. Game won’t proceed until valid position of input is given by player.
2. After player input, it is checked whether diagonally/horizontally/vertically there are three consecutive 1’s.
3. If there is, then player wins & Game ENDS!!!
4. A complete iteration ends here and repeats same logic until 5 iterations are done or anyone wins.

***Multiplayer:***

1. Names of both the players are entered.
2. 1st player starts the game and turns are alternate to both the players, hence 9 iterations needed.
3. Every time an iteration begins, current situation of the game is printed.
4. Corresponding player is allowed to give the input.
5. Game won’t proceed until valid position of input is given by player.
6. Then it is checked whether the corresponding player wins by present situation by any linear order, if it is then he is declared to have won, Else if it is the 9th iteration, it is a draw match, & game ENDS!!!
7. A complete iteration ends here and repeats same logic until 5 iterations are done or anyone wins.

IMPLEMENTATION

*WHY JAVASCRIPT??*

**JavaScript** (**JS**) is a [dynamic](http://en.wikipedia.org/wiki/Dynamic_programming_language) computer [programming language](http://en.wikipedia.org/wiki/Programming_language).[[5]](http://en.wikipedia.org/wiki/JavaScript#cite_note-FOOTNOTEFlanaganFerguson20061-5) It is most commonly used as part of [web browsers](http://en.wikipedia.org/wiki/Web_browser), whose implementations allow [client-side scripts](http://en.wikipedia.org/wiki/Client-side_scripting) to [interact with the user](http://en.wikipedia.org/wiki/User_interface), control the browser, communicate asynchronously, and alter the [document content](http://en.wikipedia.org/wiki/Document_Object_Model) that is displayed.[[5]](http://en.wikipedia.org/wiki/JavaScript#cite_note-FOOTNOTEFlanaganFerguson20061-5) It is also being used in server-side programming, game development and the creation of desktop and mobile applications.

*JAVASCRIPT AND JAVA*

A common misconception is that JavaScript is similar or closely related to [Java](http://en.wikipedia.org/wiki/Java_(programming_language)). It is true that both have a C-like syntax (the C language being their most immediate common ancestor language). They also are both typically [sandboxed](http://en.wikipedia.org/wiki/Sandbox_(computer_security)) (when used inside a browser), and JavaScript was designed with Java's syntax and standard library in mind. In particular, all Java keywords were reserved in original JavaScript, JavaScript's standard library follows Java's naming conventions, and JavaScript's Math and Date objects are based on classes from Java 1.0,[[25]](http://en.wikipedia.org/wiki/JavaScript" \l "cite_note-popularity-25) but the similarities end there.

The differences between the two languages are more prominent than their similarities. Java has [static typing](http://en.wikipedia.org/wiki/Static_typing), while JavaScript's typing is dynamic (meaning a variable can hold an object of any type and cannot be restricted). Java is loaded from compiled bytecode, while JavaScript is loaded as human-readable source code. Java's objects are [class-based](http://en.wikipedia.org/wiki/Class-based_programming), while JavaScript's are [prototype-based](http://en.wikipedia.org/wiki/Prototype-based_programming). Finally, Java did not support [functional programming](http://en.wikipedia.org/wiki/Functional_programming) until Java 8, while JavaScript does, as it contains many features based on the [Scheme language](http://en.wikipedia.org/wiki/Scheme_(programming_language)).

*Advantages*

* **Javascript is executed on the client side**  
  This means that the code is executed on the user's processor instead of the web server thus saving bandwidth and strain on the web server.
* **Javascript is a relatively easy language**  
  The Javascript language is relatively easy to learn and comprises of syntax that is close to English. It uses the DOM model that provides plenty of prewritten functionality to the various objects on pages making it a breeze to develop a script to solve a custom purpose.
* **Javascript is relatively fast to the end user**  
  As the code is executed on the user's computer, results and processing is completed almost instantly depending on the task (tasks in javascript on web pages are usually simple so as to prevent being a memory hog) as it does not need to be processed in the site's web server and sent back to the user consuming local as well as server bandwidth.
* **Extended functionality to web pages**  
  Third party add-ons like Greasemonkey enable Javascript developers to write snippets of Javascript which can execute on desired web pages to extend its functionality. If you use a website and require a certain feature to be included, you can write it yourself and use an add-on like Greasemonkey to implement it on the web page.