Tic Tac Toe Report         Yichun Zhou

**Requirements**:

The purpose of the project is to create a Tic Tac Toe game program which allows a user to play against the computer. The user can pick either PC or player to start first. Also, the user can restart a name game anytime by pressing a button. After the game is over or the game board is filled, there should be signs that show the game is finished, including something that states who the winner was, or a tie game. Also, no more moves are permitted until the client player click the start new game button. PC should never lose a game and try best to win the game, or force a tie if the game is not winnable.

**Project Analysis, Design, and Algorithm:**

I used JavaScript and HTML to create the Tic Tac Toe game; it has a very simple interface design, the player can choose either player, or computer start first by click “Player First” or “Computer First” buttons at the index page. For both PlayerFisrt and ComputerFisrt pages, there is a game board which has nine boxes. Also, there are “start new game” and “back” button at the top inside the game information box, which allows the user to restart the game and back the front page, the gaming status will also be shown in the table when the game is over, a win or a draw occurs.

In both the computer goes first and the player goes first cases, a function called checkIfPlayerWon indicates all the winning and drawing status. Firstly, the function will set integer won to 0, if there is a win occur, then won will become 1, if there is is a tie game, which means all the boxes are filled, but no winning status is found, then won will become -1. The method will pass to a different function called mark to show some indications. Also, a method called computerTurn are in both cases. However, there are several differences.

The Method computerTurn is called at the beginning, and it gets two parameters table and patterns (cross or circle), and I hardcoded the circle for computer and the cross for users. In the computer goes the first case, it uses nested if conditions to let the computer decide which is the best spot to use. First, it checks if a spot is empty and then it looks for the consecutive spots to check if there is a winning opportunity for the computer by checking if there are two circles in the same row, or column, or diagonal. Then check whether a user has any marked spot there and if any then the computer stops the player from winning by using that spot.

There is a special case for computer in order to win the game, which is generated by testing all other conditions, if the top middle and bottom right is taken by cross, then the computer moves on the bottom left, which seems hard to understand, but it will let to win a win for computer. And if the computer doesn’t find any winning opportunity or a case where the user is close to winning, then it finds a spot near its best choice and tries to win by choosing that spot. The computer would always go for the tricky choice, a corner for a start, I hardcoded the top left corner for this case. And then look for the center box is the user did not take it. In sum, the computer tries to win first, then defend the player, if either case appears, it will play on the top left corner then the center box.

In player vs. computer case computerTurn method is called after each user’s turn and look if there is a win state right after user’s turn and also after computerTurn method and same algorithm is applied in computer vs. player case, however computer plays defensively and almost all the time try to block the user’s winning state. Also, the computer will go for a defensive corner before any sides.

As a conclusion, in both cases, the computer shares a similar algorithm. However, some part such the special winning case is more relative to the computerFisrt case, and the defensive corner part is more important to the playerFirst case. If a win state is true then all the inputs are disabled, and a message will be shown to the user, and same goes to tie case.

**How to run the program:**

Internet connection is needed, and a browser to run the program. A library which is used in this program is jQuery which is fetched online from the URL. Simply open the index.html file in a browser, and you can start to play the game, or it can be excess from people.wku.edu/yichun.zhou474

**Known problems, Bugs, Limitations, Unimplemented Features:**

Internet connection is required to run the Jquery library. Otherwise, the game board sometime will not respond to cursor inputs unless you click the “start new game” button.

The cross is fixed to the player, and the circle is fixed to the computer in both PlayerFisrt and ComputerFirst cases.

**References:**

http://www.chessandpoker.com/tic\_tac\_toe\_strategy.html

https://api.jquery.com/

https://www.w3schools.com/