In this project your job is to implement [Tic-Tac-Toe](https://en.wikipedia.org/wiki/Tic-tac-toe) for two players. You also can try writing some AI to play the game. If you find it easy, try to make it unbeatable.

**What are you going to learn?**

* forEach
* Error handling in the browser's console
* Get in touch with the browser's User Interface

**Tasks**

**SHOW ALL CRITERIA**

****

**Restart the Game**

**JAVASCRIPT FUNCTION**

**0**

Pressing the restart button restarts the game.

1. When the restart button is pressed the game mode selection is displayed again.
2. When the restart button is pressed the button responsible for generating AI coordinates is hidden.
3. When the restart button is pressed the input responsible for entering the coordinates for a human player is hidden.

**CRITERIA**

**Enter coordinates when a human is playing**

**JAVASCRIPT FUNCTION**

**0**

**JAVASCRIPT BUILTIN METHODS**

**0**

During the turn of a human player an input responsible for entering coordinates is displayed

1. When it is player X's turn, entering the coordinate A1 will mark the coresponding position on the board with an X
2. After the user has entered a valid coordinate on the board the input for entering coordinates is hidden from page IF the next player is an AI
3. After the user has entered a valid coordinate the Generate AI coordinates is made visible on the page IF the next player is an AI

**CRITERIA**

**Generate coordinates when an AI is playing**

**JAVASCRIPT FUNCTION**

**0**

**JAVASCRIPT TYPES**

**30**

**JAVASCRIPT BUILTIN METHODS**

**0**

During the turn of an AI player a button responsible for generating coordinates is displayed

1. Pressing the Generate AI coordinates button will mark an empty position on the game board
2. After the user has pressed the Generate AI coordinates button, this button is hidden from the page IF the next player is a human
3. After pressing the Generate AI coordinates button an input for the human player to enter coordinates is displayed IF the next player is a human

**CRITERIA**

**Display game status**

**JAVASCRIPT BUILTIN METHODS**

**0**

**JAVASCRIPT TYPES**

**20**

**JAVASCRIPT FUNCTION**

**0**

There is a place on the page where the message of the game state is displayed

1. The message Player X's turn is displayed when player X is entering coordinates
2. The message Player O's turn is displayed when player O is entering coordinates
3. The message Invalid coordinate entered is displayed when the human player enters an invalid coordinate (ex D3, 1A etc)
4. The message Position is already taken on board is displayed when the user enters a coordinate for a position on the board that is already taken.
5. The message It's a tie is displayed when there are no more free spots on the board BUT neither player won
6. The message Player X has won ! is displayed when player X has won

**CRITERIA**

**Game end**

**JAVASCRIPT BUILTIN METHODS**

**0**

**JAVASCRIPT FUNCTION**

**0**

**JAVASCRIPT CALLBACK**

**0**

The game can be ended when there is a tie or one of the player's have won

1. The html elements for entering human coordinates or generating AI coordinates are hidden when the game has ended.
2. The Restart game button is displayed when the game has ended.

**CRITERIA**

**AI can play**

**JAVASCRIPT BUILTIN METHODS**

**0**

**JAVASCRIPT CALLBACK**

**0**

Implement the game mode for a human to play against an AI

1. When the game mode Human vs AI is chosen the human player plays as X and the AI player plays as O
2. When the game mode Human vs Human is chosen the human player plays as X and O

**CRITERIA**

**OPTIONAL TASK: AI goes for easy win**

**JAVASCRIPT BUILTIN METHODS**

**0**

**JAVASCRIPT ARRAY**

**10**

**JAVASCRIPT CALLBACK**

**0**

AI is capable of recognizing the opportunity to win the game with one move.

1. Function getUnbeatableAiCoordinates() picks the winning move if there is one on the board.

**CRITERIA**

**OPTIONAL TASK: AI prevents easy lose**

**JAVASCRIPT BUILTIN METHODS**

**0**

**JAVASCRIPT ARRAY**

**10**

**JAVASCRIPT CALLBACK**

**0**

AI is capable of recognizing if its enemy could win the game with the next move, and (supposing there is no direct winning move) moves against it.

1. Function getUnbeatableAiCoordinates() (when there is no winning move in one step) picks a move which prevents a certain winning move for its enemy.
2. When there is a direct winning move, function getUnbeatableAiCoordinates() still picks that.
3. When there are multiple one-step options for the enemy, getUnbeatableAiCoordinates() tries to prevent one of them.

**CRITERIA**

**OPTIONAL TASK: Unbeatable AI**

**JAVASCRIPT BUILTIN METHODS**

**0**

**JAVASCRIPT ARRAY**

**10**

**JAVASCRIPT CALLBACK**

**0**

AI is unbeatable in all cases.

1. There is no strategy or combination of steps that can win the game against the AI.

**CRITERIA**

**OPTIONAL TASK: AI vs AI**

**JAVASCRIPT BUILTIN METHODS**

**0**

**JAVASCRIPT CALLBACK**

**0**

You can choose the game mode AI vs AI

1. When the game mode AI vs AI is selected only the buttons for generating coordinates are visible

**CRITERIA**

**General requirements**

* Use forEach instead of for loops when applicable

**Hints**

* This time you don't have to run any file in the terminal, just open the index.html in the browser
* You don't have to come up with an AI strategy. You can search the internet for strategy descriptions. Do not use external code; implement written instructions instead.
* You don't have to implement a general playing strategy. Tic-Tac-Toe has a rather easy unbeatable strategy that can be expressed as a sequence of conditionals.
* Ideal team size is 3. Maximum team size is 3.
* for the tasks linked to hide / display certain html elements the code is already implementd in the file do-not-modify-this-file.js as functions that you can call in the code you implement.