Tutorial 01 Tic Tac Toe

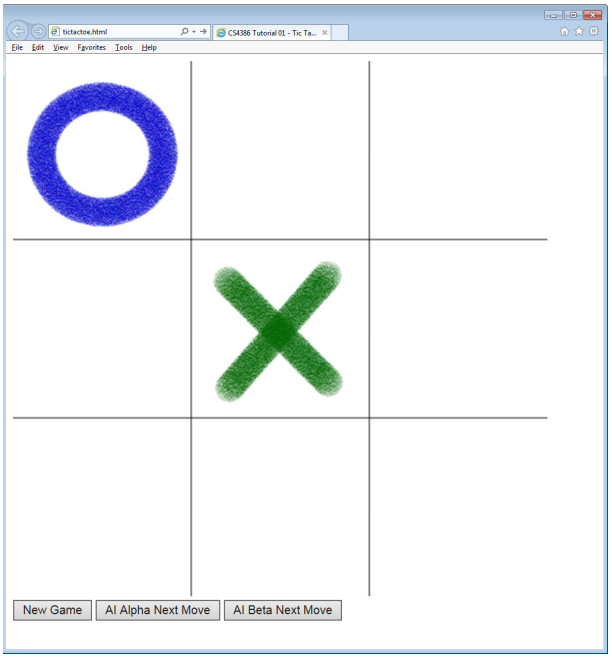
Introduction

In this tutorial, you are going to work on a Tic Tac Toe game that is written in HTML 5 and Javascript.

Task 1. Get Ready

Download the files `tictactoe.html` and `ttt.png` from the course website.

Run the html file on a web browser.



You can click on any of the 3x3 cells to provide the current player's move, or you can click the button "AI Alpha Next Move" or "AI Beta Next Move" to let the AI decide on the current's player's move.

Note that the rules for this Tic Tac Toe game are as follows:

1) Player 1 (indicated by blue circle) goes first and then Player 2 (indicated by green cross) goes next.
2) Each player takes turn to occupy a cell until a player wins or the game ends in a draw.
3) A player wins if that player has occupied 3 cells in the same row, in the same column, or diagonally.
4) The game ends in a draw if all the cells are occupied and there is no winner.

Try the game a few times to make sure that you understand how the gameplay and how the buttons work.

Task 2. Examine the Code

Examine the code in `tictactoe.html` in a code editor so that you understand the overall flow of the program. In particular, note that

1) The array `boardState` stores the current game state. The first index stores the game state of upper left corner of the game board while the last index stores the lower right corner of the game board. The complete mapping of index and (x,y) values are given below:
   [0][1][2] [(0,0)] [(1,0)] [(2,0)]
   [3][4][5] [(0,1)] [(1,1)] [(2,1)]
   [6][7][8] [(0,2)] [(1,2)] [(2,2)]

Each element of `boardState` is an integer that indicates the state of the corresponding cell:
0: unoccupied
1: occupied by Player 1
2: occupied by Player 2

2) The variable `Turn` indicates whose turn it is:
Endgame (0): the game ends so it is no one's turn
Player1 (1): currently it is Player 1's turn
Player2 (2): currently it is Player 2's turn
The variable is updated in the function `mark()`.

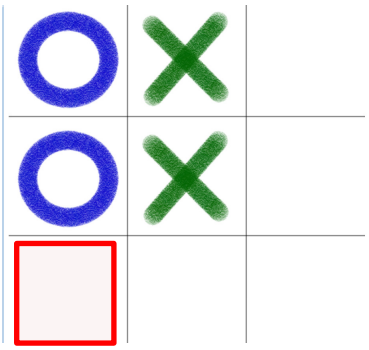
3) The functions `AI_a_NextMove()` or `AI_b_NextMove()` are called when the buttons "AI Alpha Next Move" or "AI Beta Next Move" is clicked. It will generate a move for the current player.
   - AI Alpha just scans the board sequentially and occupies the first unoccupied cell
   - AI Beta scans for all unoccupied cells in the board and randomly picks an unoccupied cell to occupy

---

Task 3. Add Your Code

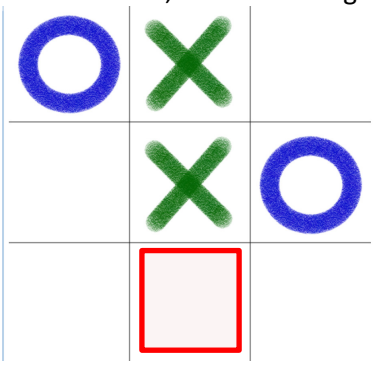
Add more buttons which will call new functions to make the next move using different strategies:

1) AI Gamma
It will first check the board and occupies the cell that will win the game immediately. For example:



It is Player 1's turn and AI Gamma should occupy cell with index=6 (x=0, y=2).
If there is no such cell that will let the current player win the game immediately, then it will just select an unoccupied cell randomly, i.e., the same as AI Beta.

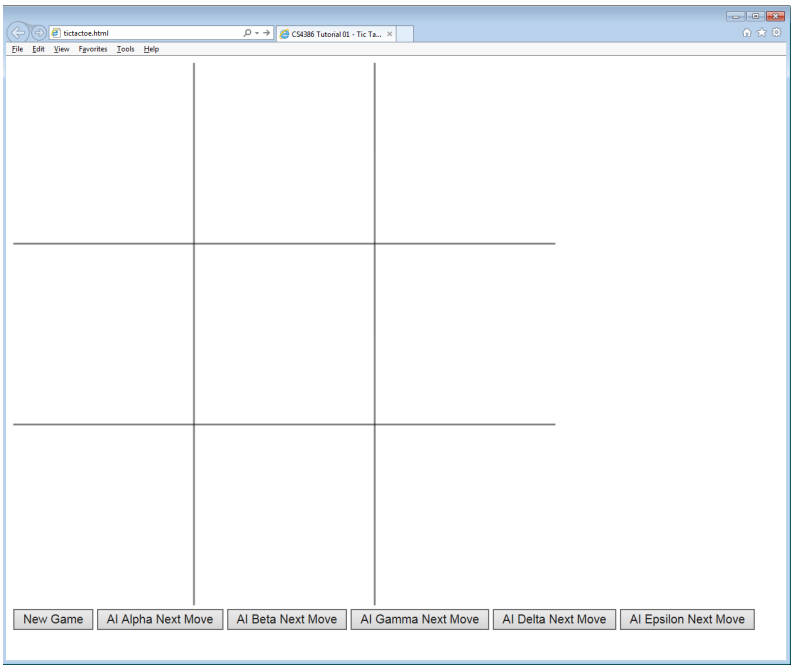
2) AI Delta
It will first check the board and occupies the cell that will win the game immediately (same as AI Gamma up to now). If there is no such cell that will let the current player win the game immediately, then it will check the board and occupies the cell where the other player has occupied two cells in the same row, column or diagonally. For example:



It is Player 1's turn and AI Delta should occupy cell with index=7 (x=1, y=2).

If there is no cell that will let the current player win the game immediately, and there is no cell with which the other player has occupied two cells in the same row, column or diagonally, then it will just select an unoccupied cell randomly, i.e., the same as AI Beta.

3) AI Epsilon
   Here you are free to implement your own AI strategy. See if you can come up with an AI that is better than AI Alpha, AI Beta, AI Gamma and AI Delta.



Note that in writing the AI function, you can use the variable `Turn` to know whether it is Player 1 or Player 2's turn. Your algorithm needs to determine an index (0…8) that indicates an unoccupied cell for the current player to occupy. Then it should calculate the corresponding x and y values from the index and call the function `mark(x,y,Turn)` to make the move.

You can test your algorithms by letting different AIs play against each other.

Task 4. Complete the Canvas Quiz

Complete the quiz "Tutorial 01" on the Canvas course page (Assignments > Tutorial 01)