Technical documentation for Tic-tac-toe Project

Introduction

HTML, CSS, and JavaScript are three district coding languages that together are used to build Website and Web Applications.

- **HTML: H**yper-**T**ext **M**arkup **L**anguage is used to put the structure of a website together. (Like a skeleton of a body)
- **CSS:** Cascading Style Sheets acts like makeup for the HTML. CSS improves the colors and layout of a website structure built with HTML.
- **JavaScript** is a full-on programming language that adds interactivity and functionality to a website.

I am doing this project as a sole contributor. My project is to implement the Tic-Tac-Toe game by using HTML,CSS and JavaScript which will features single player mode. In this, I have used game theory logics like minimax algorithms to determine the best move that the computer(which we can call it as AI agent) plays.

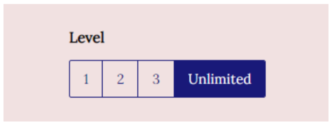
How to play

- Choose the level of difficulty and then choose the symbol of your choice.. either ‘X’ or ‘O’.
- Click the “New Game” button and click on the cell of your choice on the tic-tac-toe board.
- First player is you then AI player will mark its move.

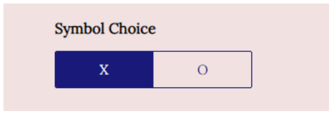
- Then each player taking turn draw their symbol on a space from that nine possible spaces.
- The one who met the winning condition first win.
- If after nine possible spaces are used, but no one wins, that game is a draw.

Special Features

- In this project, there is a button for choosing difficulty level. In that You can choose the difficulty of the game according to you choice. Level 1 is the easiest level you can win the game. But as the level will increase, the difficulty of the game increases. At the Level unlimited game becomes Unbeatable.

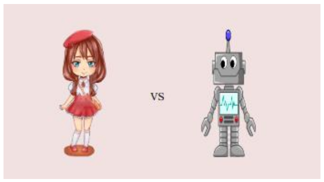


- Button for choosing symbol of your choice. In this, you can choose the symbol of your choice.. either 'X' or 'O'.



- A New Game button for starting the New Game.

- Just to give it a decent look or to make it more presentable I have used the picture of human and robot which I have taken from Internet.



- I have added a background image for the background of tic-tac-toe grid which also is taken from internet.

Implemetation

1. Structing the Web Application by using HTML5. Adding the basic features by using HTML .
   - Level of game and the symbol choice.
   - Adding a 3 x 3 grid with the help of table tag.

2. Designing the Graphical User Interface by using CSS.
   - Styling the interface as a one page web application.
   - Stying the buttons for symbol and Level.
   - Adding a background image for the tic-tac-toe board.
   - Styling the Endgame message etc.

3. Building the behavior of the page using JavaScript.

   - startGame(depth,Human_symbol) –

     1. It will assign value to cells index to the origboard.
     2. Assign the symbol to huPlayer and aiPlayer.

3. Removing the previous game moves and the background color.
4. Adding the event-Listener for the click on the tic-tac-toe board.
5. calling the turnClick function.

- turnClick(square) –

  1. checking the target key, if it is empty or full.
  2. If yes, then calling the turn function means it is human player turn.
  3. After returning checking win and tie for the human player, if not then calling the turn function for ai player move.

- turn(squareId, player) –

  1. Marking the symbol on the origBoard of the respected player.
  2. Checking win for the player.
  3. If yes then calling the gameOver function.

- checkWin(board, player) - checking the winner by looking the winning combo and also verifying that they are from the same player or not.

- gameOver(gameWon) – coloring the winning combos and removing the click event listener.

- declareWinner(who) – declaring the winner at the end of the program.

- emptySquares() – Checking the empty squares on the origboard and then returning it to the minimax function.
- bestSpot() - calling the minimax function for the best move and returning it to the turn function.

- checkTie() – if there is no available moves, and none of the player has been satisfied the winning condition then return true, else false.

- A Minimax algorithm can be best defined as a recursive function that does the following things:
    1. return a value if a terminal state is found (+10, 0, -10)
    2. go through available spots on the board
    3. call the minimax function on each available spot (recursion)
    4. evaluate returning values from function calls
    5. check the depth of the function and behaves accordingly.
    6. and return the best value by analyzing, and by considering the depth/Level of the function.
    7. For huplayer we will maximize the score as we are expecting the best move from huPlayer as well.
    8. For aiPlayer we will minimize the score as we are giving the best move so that human player cannot win.

Conclusion

In the conclusion of this project, I would like to say that HTML, CSS and JavaScript are easy programming languages and while creating a project like this, it has not just been a good experience but it also helped in the development of my creativity and logical thinking. I would be more than happy to work on other projects with Microsoft because it's just amazing to get Mentorship from Microsoft. The program is working and I hope, it's also bug-free.

References

- Wikipedia
- Youtube
- Geeks for Geeks
- Github Repository provided by the acehacker team.

Made by - Nidhi