TEAM POWERPUFF

TANYA GUPTA
PRANJALI BAJPAI

AI POWERED TIC TAC TOE

25th JULY 2020

OVERVIEW

In this project we have tried to stimulate a simple pen-paper traditional 2-player game of tic-tac-toe virtually on a web application. It has been provided with different modes for playing as such - against computer and against an actual crew member. There are many other options like choosing symbols, choosing who goes first, and choosing levels in order to improve the overall user experience by giving a comfortable environment. Also, flow of this application from the beginning to game start is kept sequential to avoid confusion and keep it simple for the user. Graphics are also included to improve the visuals.

WorkFlow of Options

WorkFlow for our game for customising options is given below

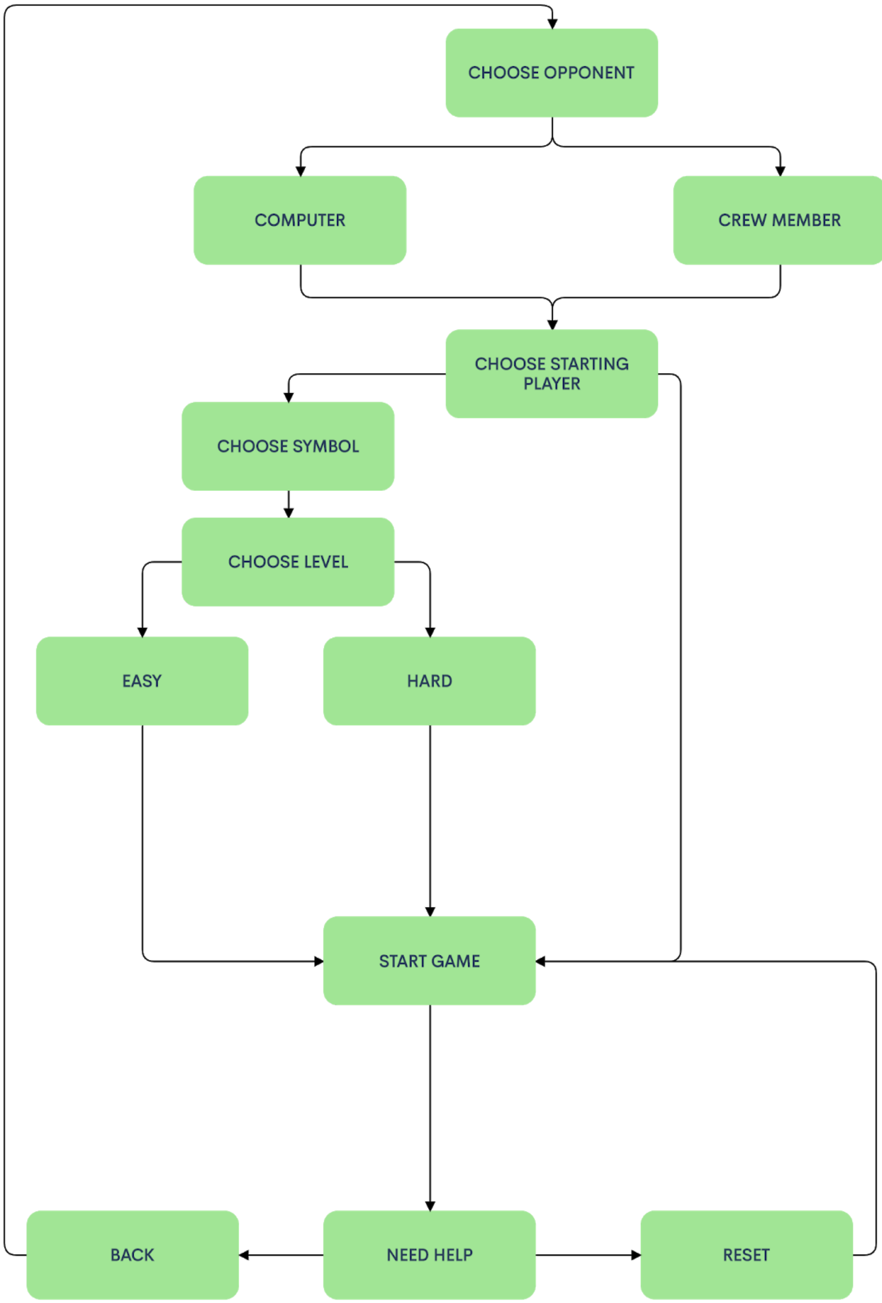
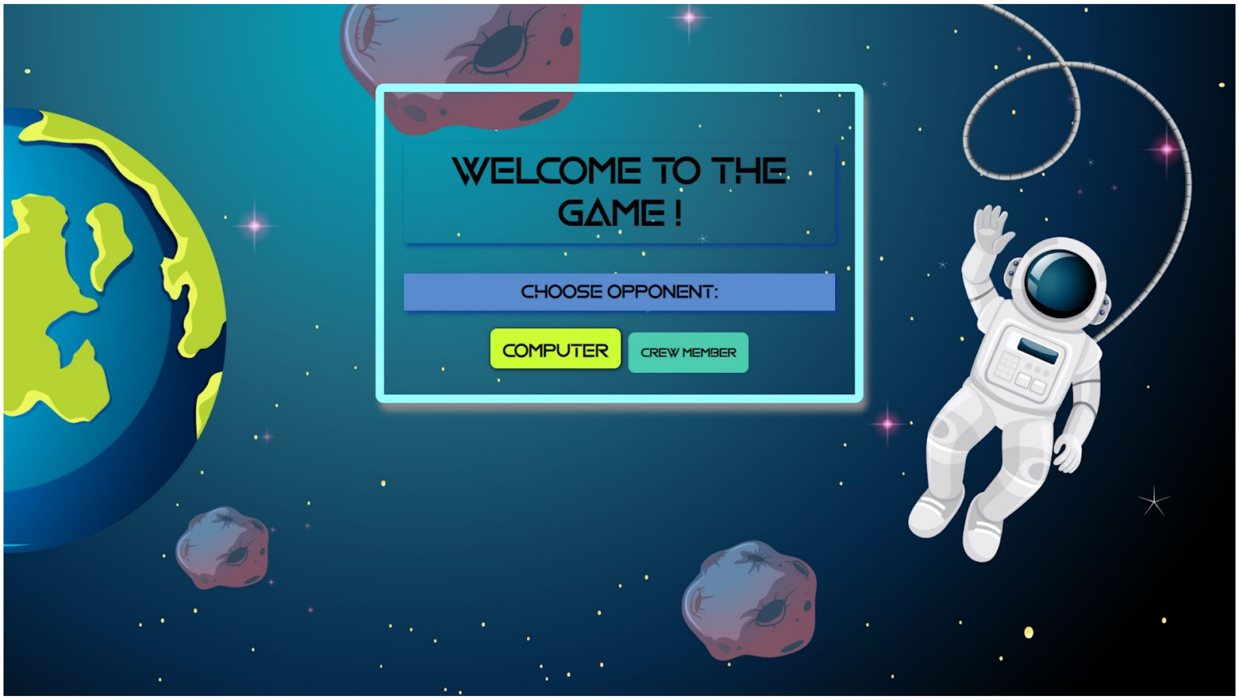


Chart-1 Workflow for user options
Right-side: shows workflow if 2nd-player is crew member; Left-side: shows workflow if 2nd-player is computer; while Need Help, Reset, and Back are common for both

Visual Representation

To illustrate our UI interface below are few visuals that user might see while selecting for options.

- **CHOOSE OPPONENT**

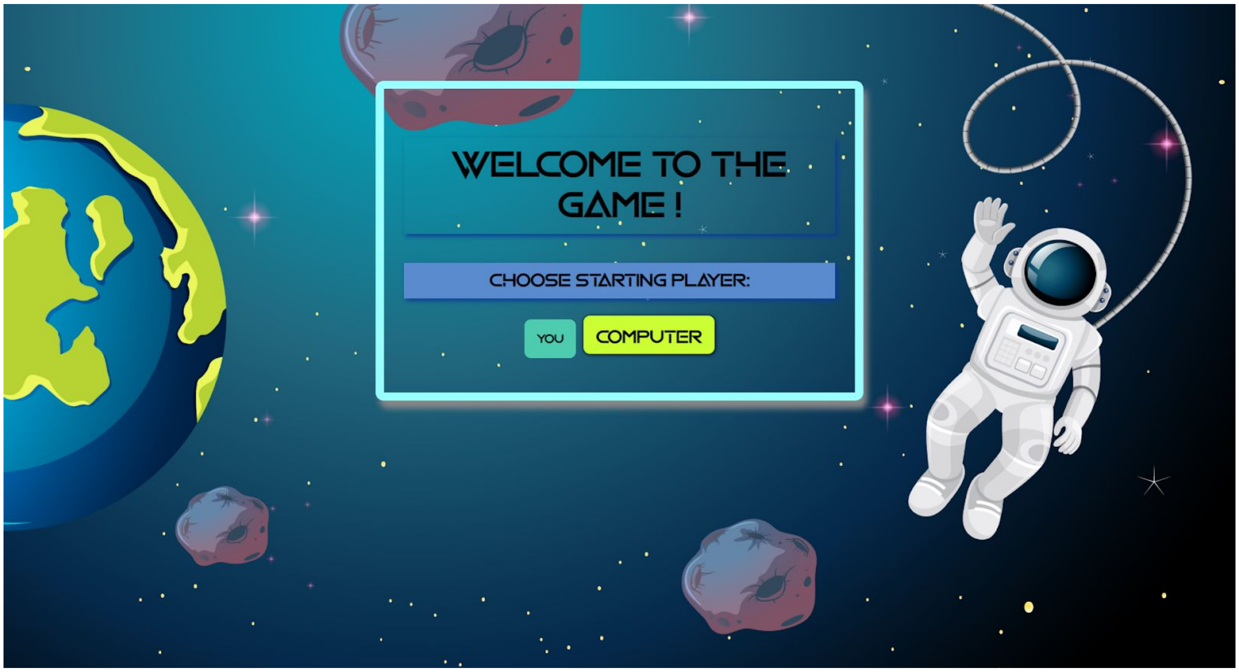


WELCOME TO THE GAME !

CHOOSE OPPONENT:

Fig-1: Choose opponent

- **CHOOSE STARTING PLAYER**



WELCOME TO THE GAME !

CHOOSE STARTING PLAYER:

Fig-2: Choose starting Player

CHOOSE SYMBOL

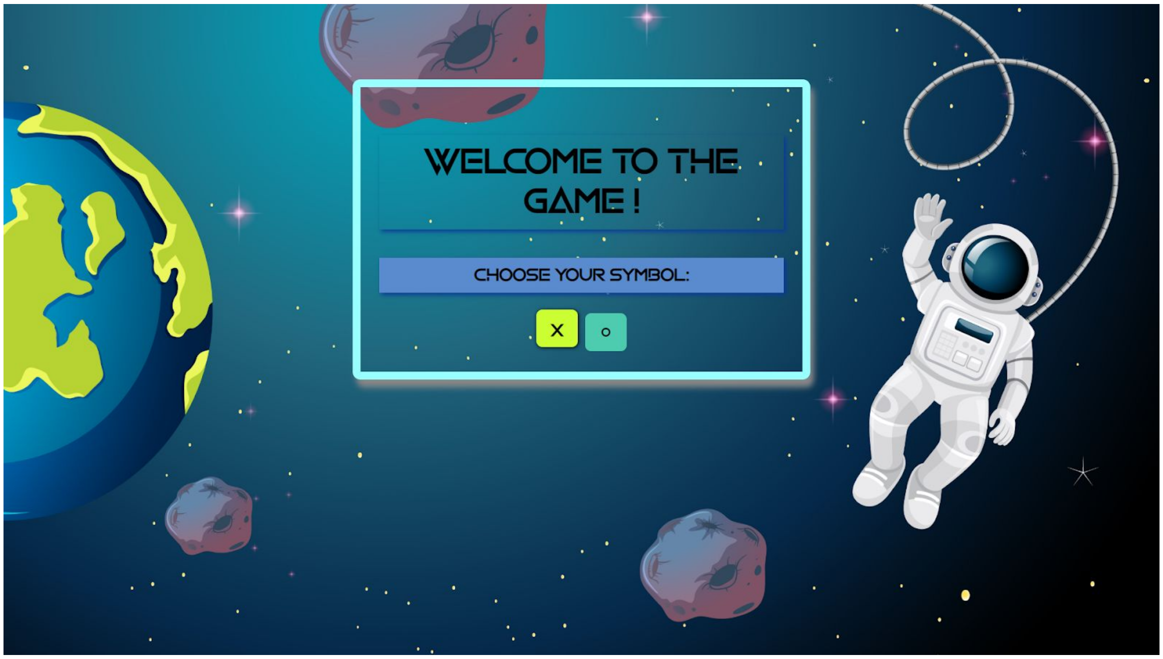

WELCOME TO THE GAME !

Fig-3: Choose your symbol

CHOOSE LEVEL (AGAINST COMPUTER)

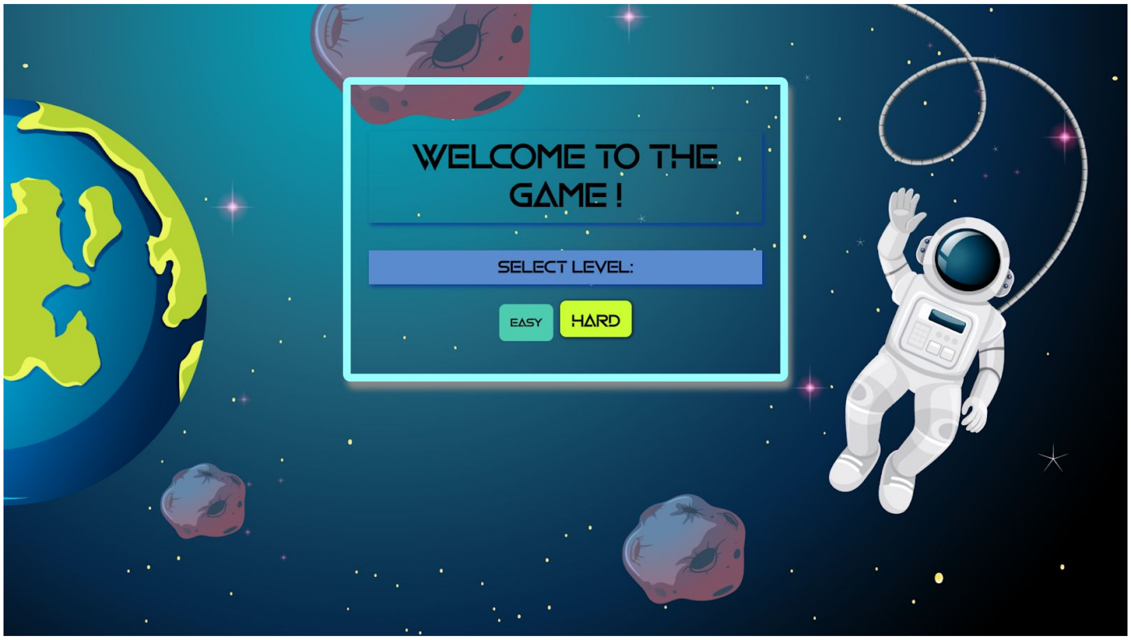
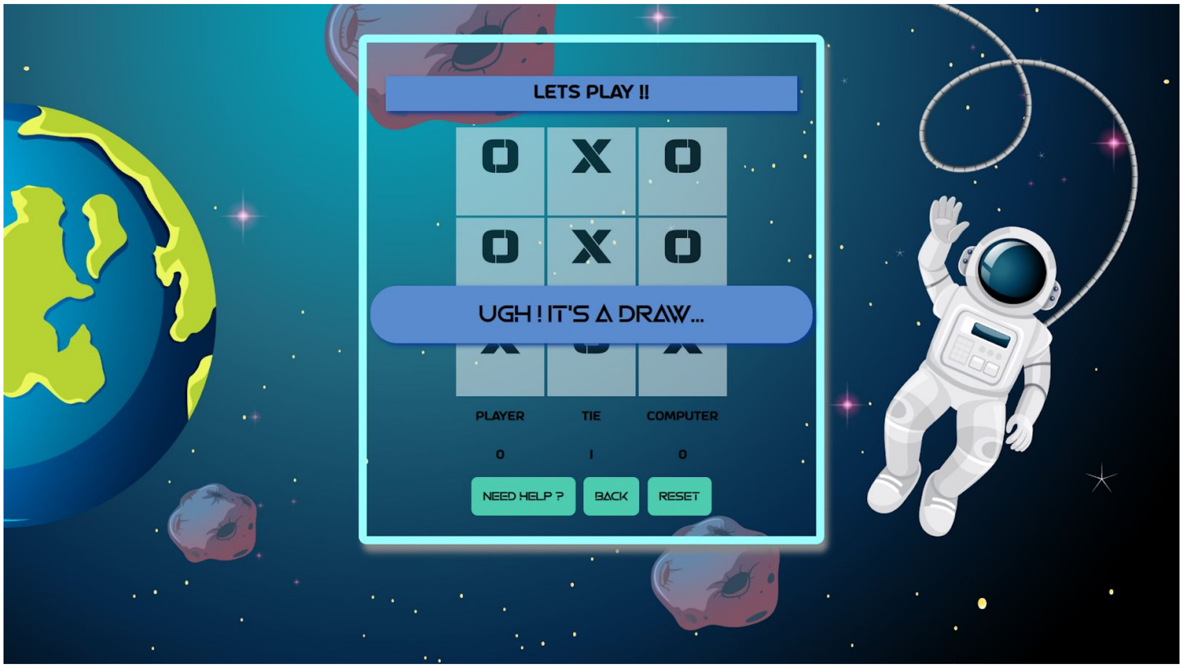

WELCOME TO THE GAME !

Fig-4: Select Level (This option only appears if 2nd player is computer)

- **GAME AGAINST COMPUTER**



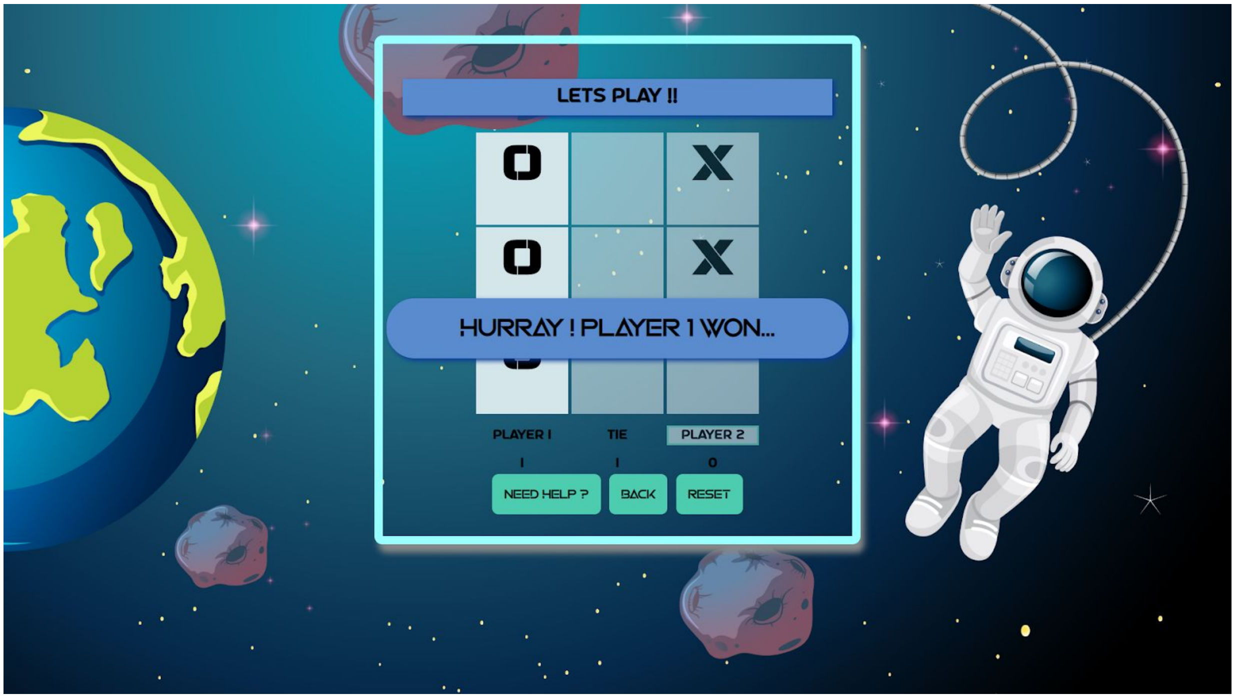
LETS PLAY !!

UGH ! IT'S A DRAW...

PLAYER	TIE	COMPUTER
0	1	0

Fig-5: Glimpse of a game played against Computer

- **GAME AGAINST A CREW MEMBER**



LETS PLAY !!

HURRAY ! PLAYER 1 WON...

PLAYER 1	TIE	PLAYER 2
1	1	0

Fig-6: Glimpse of a game played against a crew member

SPECIFICATIONS

- ### LEVELS

  Our game will allow players to choose two different difficulty levels when playing against the computer as shown in Fig-4:

  - **EASY**

    In easy level, the computer will choose it’s move randomly from the available set of moves. The player has a possibility to win at this level.

  - **HARD**

    In this case, the computer chooses it’s next move from the MINIMAX ALGORITHM. Due to which the opponent human player will not be able to win at this level. The result of the game for human players will be either draw or defeat. As clearly seen from Fig-5 which is a draw.

Brief about our Algorithm

Most widely used algorithm to make an AI enabled tic-tac-toe game is MINIMAX Algorithm. This algorithm works best when you need to defeat the opponent as well as maximize the current player. But this algorithm works only when there is a single player to be optimised rather than shifting players. For this part of the game we implemented a simple minimax algorithm to optimize the computer's score while minimizing the player’s score. This provided with sufficient results.

Dry run for this algorithm is depicted below:

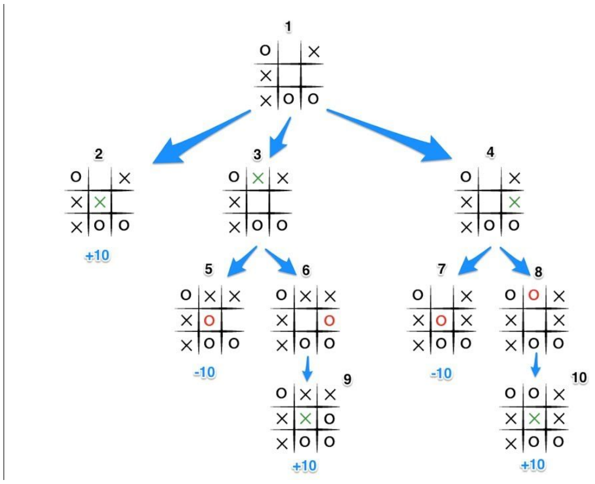


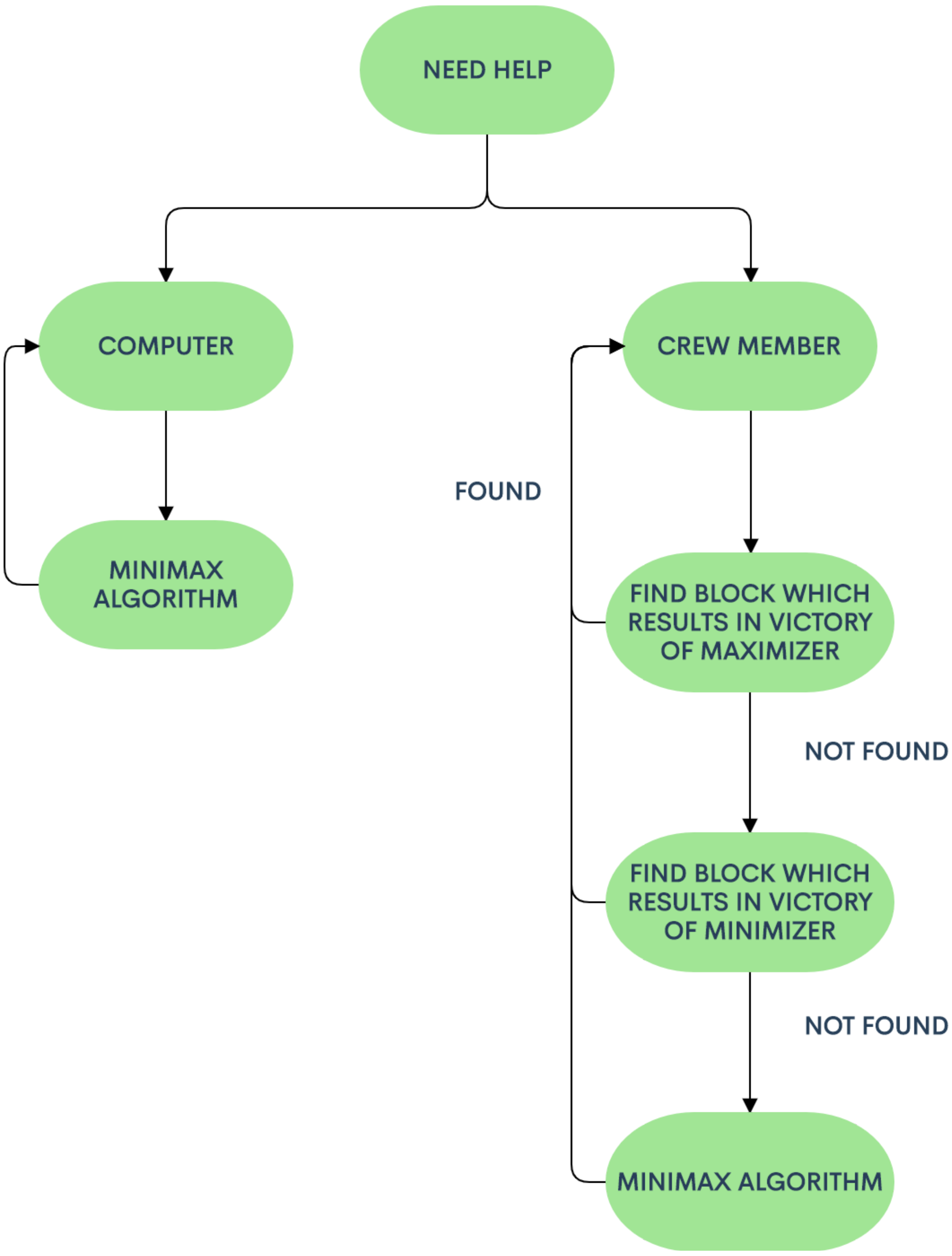
Fig-7:Minimax Algorithm (‘X’ is optimised while ‘O’ is minimised)

- **NEED HELP BUTTON**

Upon pressing the Need Help button, Suggested Move for the current user will be highlighted on the board. The Suggested Move function uses minimax function with current player as maximizer and opponent as minimizer to determine the best possible move for the current player, thereby helping the player to make a way to either win or draw the game.

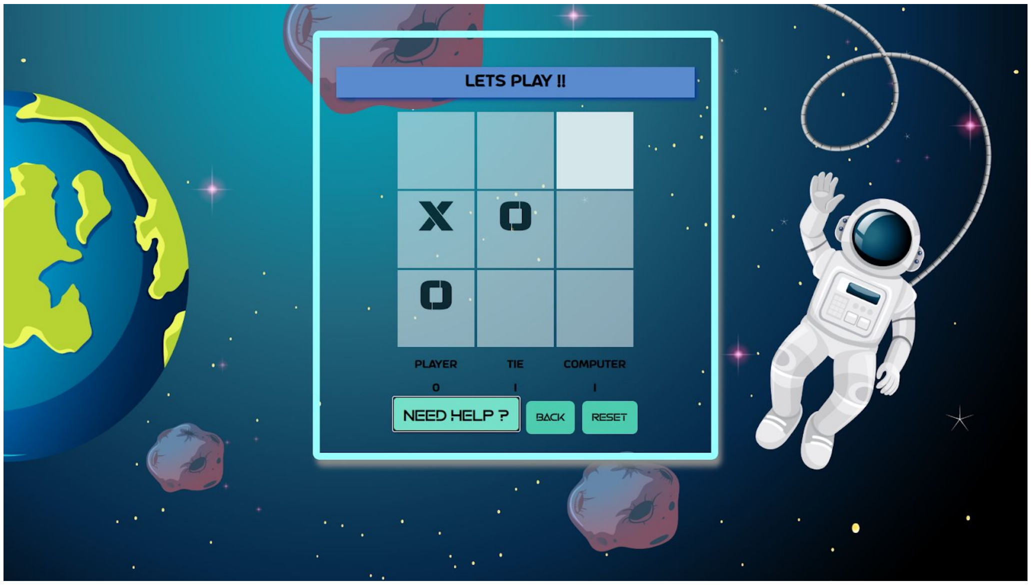
Brief about our Algorithm

For this part, simple minimax was not giving promising results as pretty obvious boxes were not suggested solely


NEED HELP
COMPUTER
CREW MEMBER
MINIMAX ALGORITHM
FOUND
FIND BLOCK WHICH RESULTS IN VICTORY OF MAXIMIZER
NOT FOUND
FIND BLOCK WHICH RESULTS IN VICTORY OF MINIMIZER
NOT FOUND
MINIMAX ALGORITHM

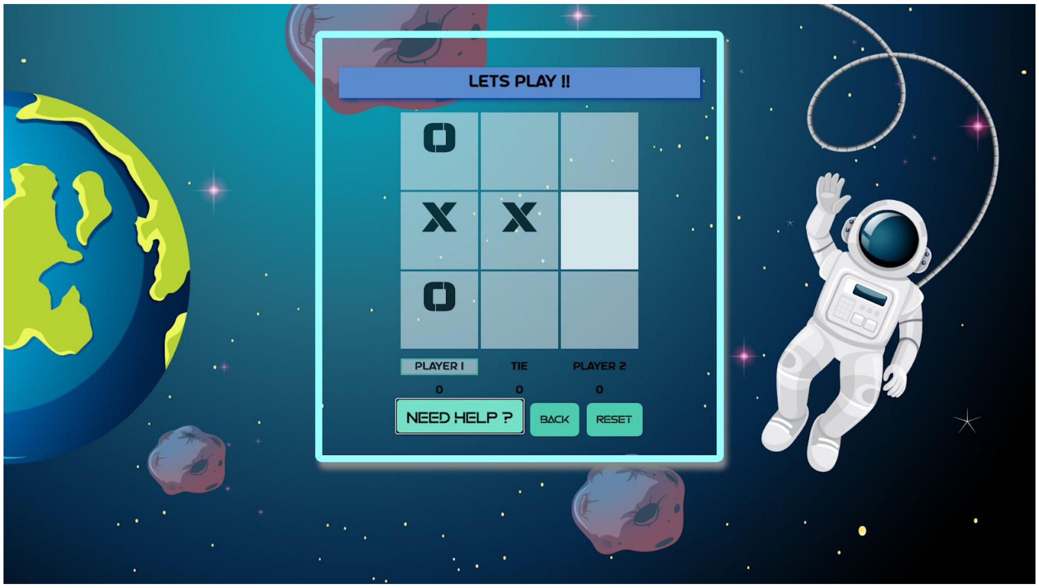
- **AGAINST COMPUTER**

  In this case, the computer is ‘O’ and the player is ‘X’. When the player presses the **NEED HELP** button then the 3rd block on the board will be highlighted, hence blocking the computer.

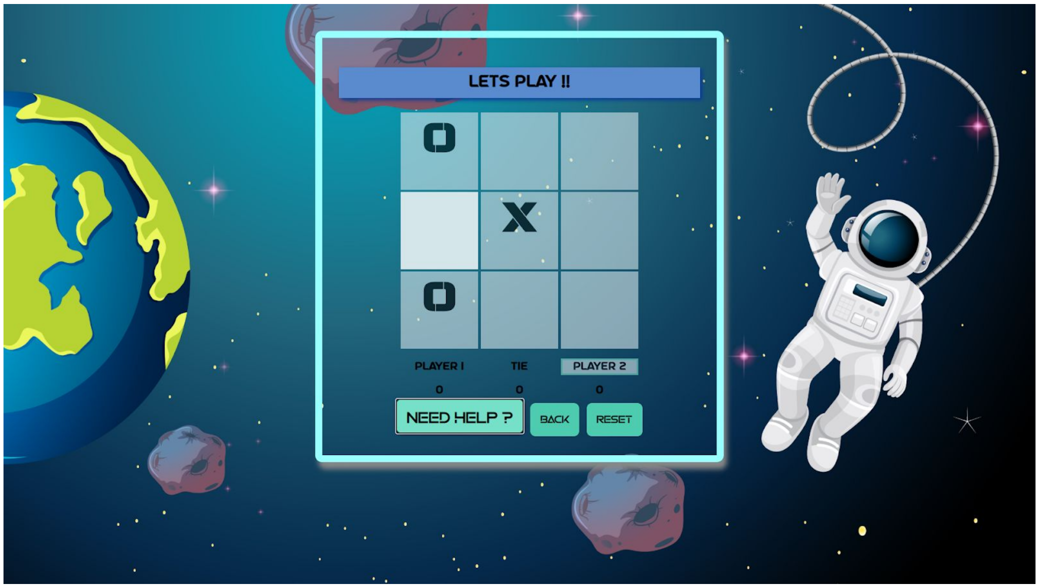


- **AGAINST A CREW MEMBER**

  In this case in order to determine the best possible move for the current player, slight modification in the minimax algorithm was introduced.The current player was set to maximizer and the opponent to minimizer.Firstly the block was determined which could result in victory of the current player.If such block was found this block was suggested else if no such block was found then the block was checked which could result in the victory of opponent.If this kind of block was found then maximizer is suggested to make next move to this block, else if no such block was found then next suggested move was found using **MINIMAX ALGORITHM.**



SUGGESTED MOVE FOR PLAYER 'O'



SUGGESTED MOVE FOR PLAYER 'X'

- **BACK**

  This button will redirect the player to the homepage of the web-app.

- **RESET**

  This button will reset the game, and a new game will be started.

URL : https://pranjalibajpai.github.io/Tic-Tac-Toe/

GITHUB REPOSITORY: https://github.com/pranjalibajpai/Tic-Tac-Toe