Experiment :  Tik Tac Toe AI Game.

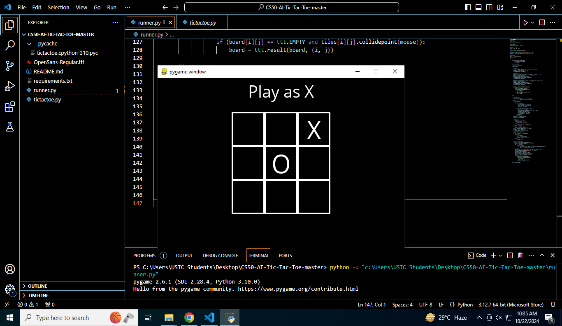
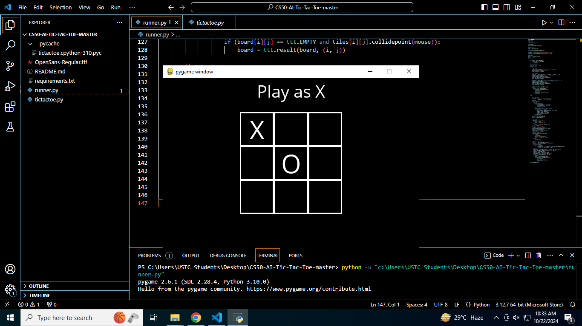
Explanation :

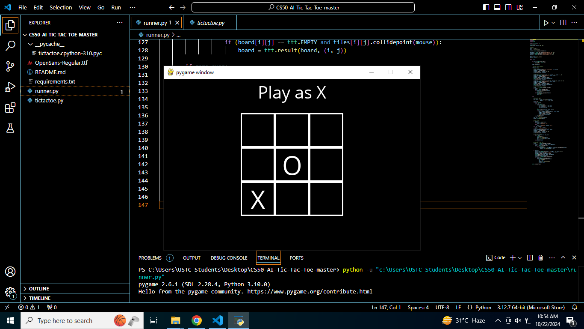
**How the Ai work in this game:**

In this game the game use the Minimax algorithm to determine the best possible move by simulating all future game states. AI (Maximizing player) aims to maximize its score, while the human player (Minimizing player) tries to minimize the AI's score. Ai tries to blocked the the the next move of the human.

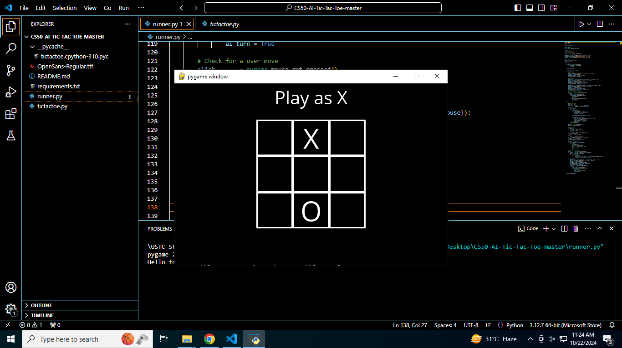
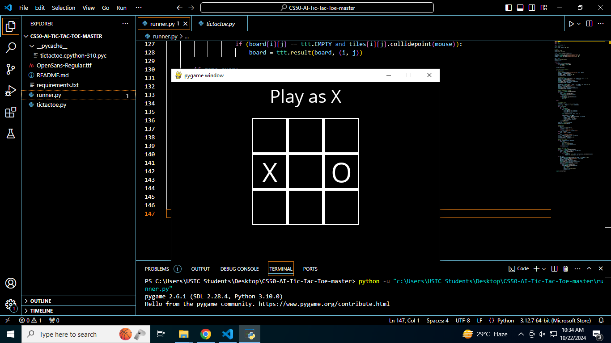
If human choses the X and give the 1 st move on 1st row 1 block ,1st row last block or 3rd row 1st block then  Ai give the move on 2nd row middle block. Because Ai determine the best possible move by simulating the future game states. Ai have to blocked the next possible  move of the human for every time.

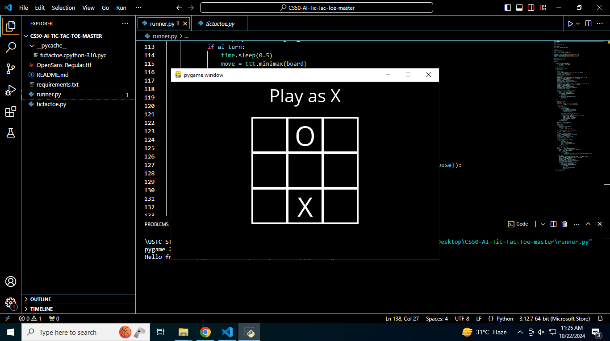
The AI recursively evaluates all possible moves, generating a game tree where each node represents a game state and each branch represents a potential move. For each terminal state (win, loss or tie), a utility value is assigned: +1 for an AI win, -1 for a human win and 0 for a draw.



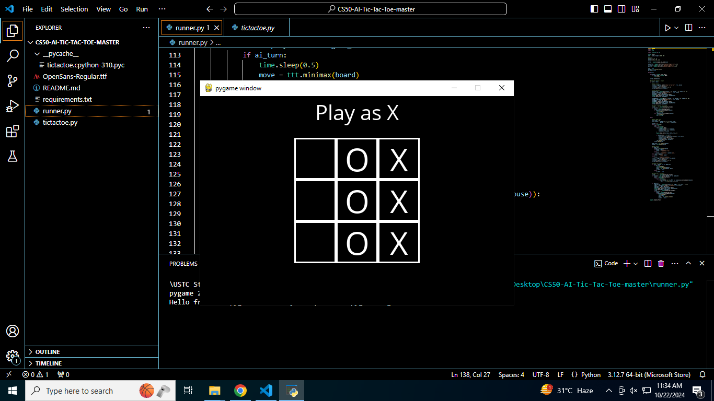


If human want to give the move on 2nd row 1st block Ai gives the move on 2nd row last block and for 1st  row 2nd block Ai give O on 3rd row 2nd block and for 3rd row 2de block Ai give O on 1st row 2nd block every time  .





But there is a bug in this game which is if human give the X on  3rd column any block Ai give the O on 2nd column and after choosing X time time in same column game doesn’t give X is winner. It tells to gives the X turn again. And if human gives the X on



**Which Concept flowing the Ai game:**

The AI traverses this tree, using a depth-first approach, to choose the move that maximizes its chances of winning while considering the opponent's optimal responses. Although the code doesn't implement it, the algorithm could be optimized with Alpha-Beta pruning to reduce the number of explored nodes, improving efficiency. Ultimately, the AI chooses the move with the highest utility, ensuring it plays optimally in every scenario.