
**Faculty of Engineering and Technology Electrical and Computer Engineering Department**

# ARTIFICIAL INTELLIGENCE

## ENCS3340

## Second Semester 2023

## Project # 1 Magnetic Cave.


| Prepared by: | ID: | Section: |
|---|---|---|
| Leena Affouri | 1200335 | 2 |
| Mariam Hamad | 1200837 | 4 |


**Instructor:** Dr. Yazan Abu-Farha

**Date:** 20/6/2023

## Description of the program :

This is a Magnetic Cave game that need two player at most to play it .First the player most to determined the player mode then if the player pick to play first he must to put the first position on the board then wait to even the second player to play or to the automatic to do the second move and so on.

If the player pick a position where there is no magnetic a message will appear that Invalid move, Please choose another position also if he pick an already occupied position a message will appear that Invalid move. The position is already occupied. Please choose another position otherwise they will complete playing when a player is can make a five correct position in row ,column or diagonal he will win if the board is full the game will end and a message will appear It's a tie

## The main function on the program is :

**check_winner** : this function check if there is five row column or diagonal sequentially full for the same player it will return it and end the program and a message appear that this player is win the game.

**Is_move_can**: this function will check if the move that the player select is tracking the condition about the magnetic if yes it return true if no it return false

**make_move** : this function if the position that the player is select valed put the stare in the correct position on the board

**score_in_board**: this function give a score for each player if the player is max it add 100 to the score else it sub 100.

**minimax** : this is the main function that do the minmax algorithm It is used to determine the best move for the AI player in a game The minimax algorithm explores the game tree by recursively evaluating all possible moves and their resulting positions, considering the best moves for the maximizing player and the worst moves for the minimizing player. Alpha-beta pruning is used to optimize the search by pruning branches of the game tree that are guaranteed to not affect the final decision.

**get_best_move**: function determines whether the player can win in the following move and returns the appropriate move if so. Otherwise, it uses the minimax algorithm with a time restriction to choose the AI player's best move while taking rising depths into account until the time limit is achieved. If no advantageous move is detected, it chooses a legitimate move at random.

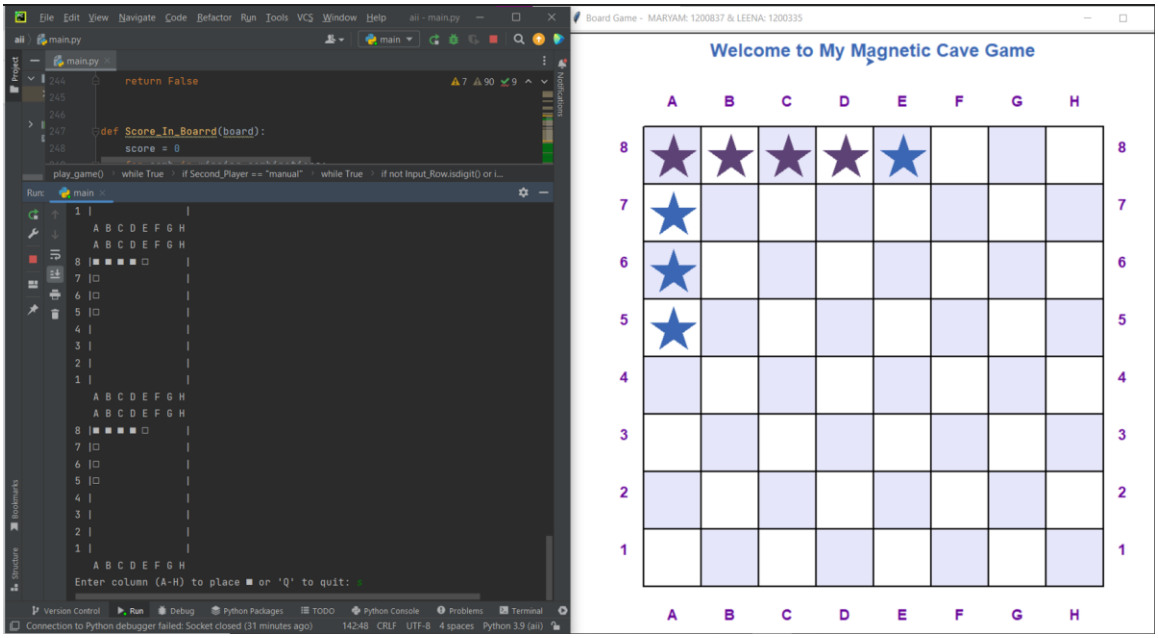
**play_game:** function simplifies the gameplay loop by managing the turns of both the human and AI players, validating movements, updating the board, displaying the board, determining whether a winner has been determined, and ending the game when a win or tie condition is satisfied.

## description of the heuristic :

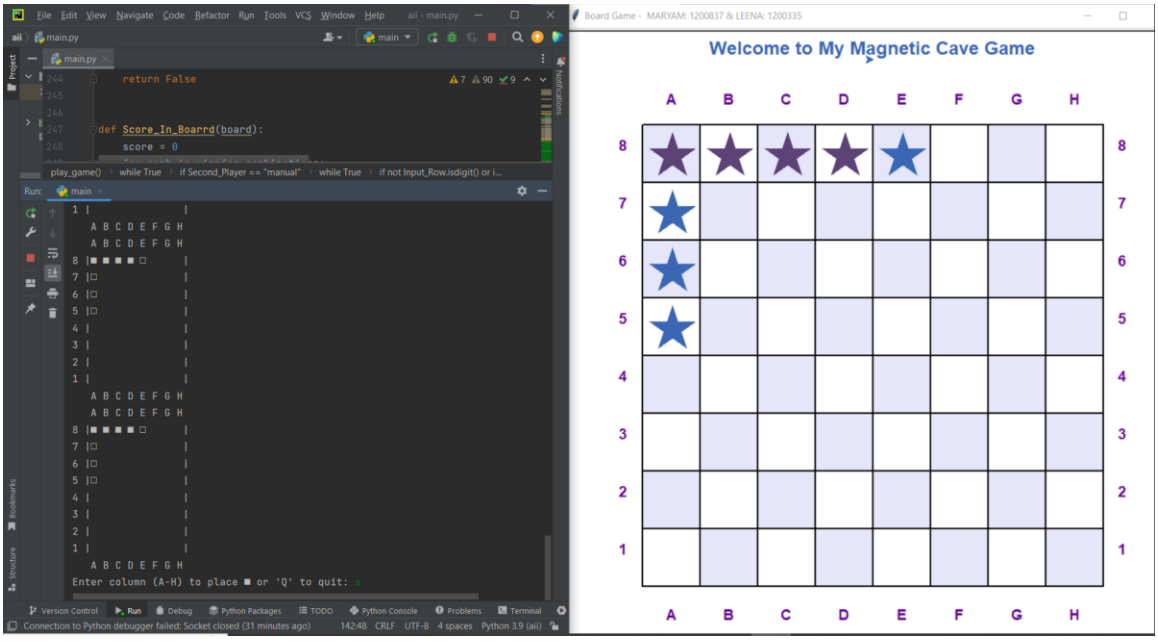
The evaluation function calculates a heuristic score for the current state of the game board ,Initialize the score variable to 0. Iterate through each combination of winning patterns in the winning_combinations list. If the AI player has 5 occurrences in the cells, it means it has a winning combination. In this case, add 100 to the score. If the opponent has 5 occurrences in the cells, it means the opponent has a winning combination. In this case, subtract 100 from the score. Return the final score. A higher positive score indicates a favorable position, while a lower negative score indicates a disadvantage. The AI player aims to maximize this score to make more advantageous moves.

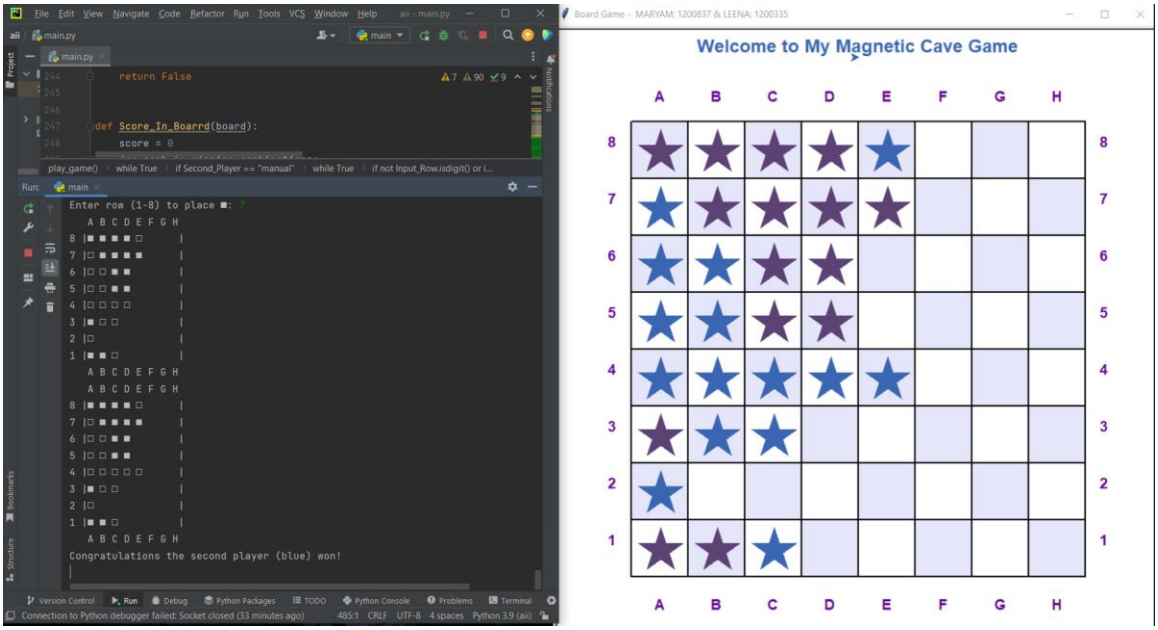
## describes of the results :

first and second player are manual :

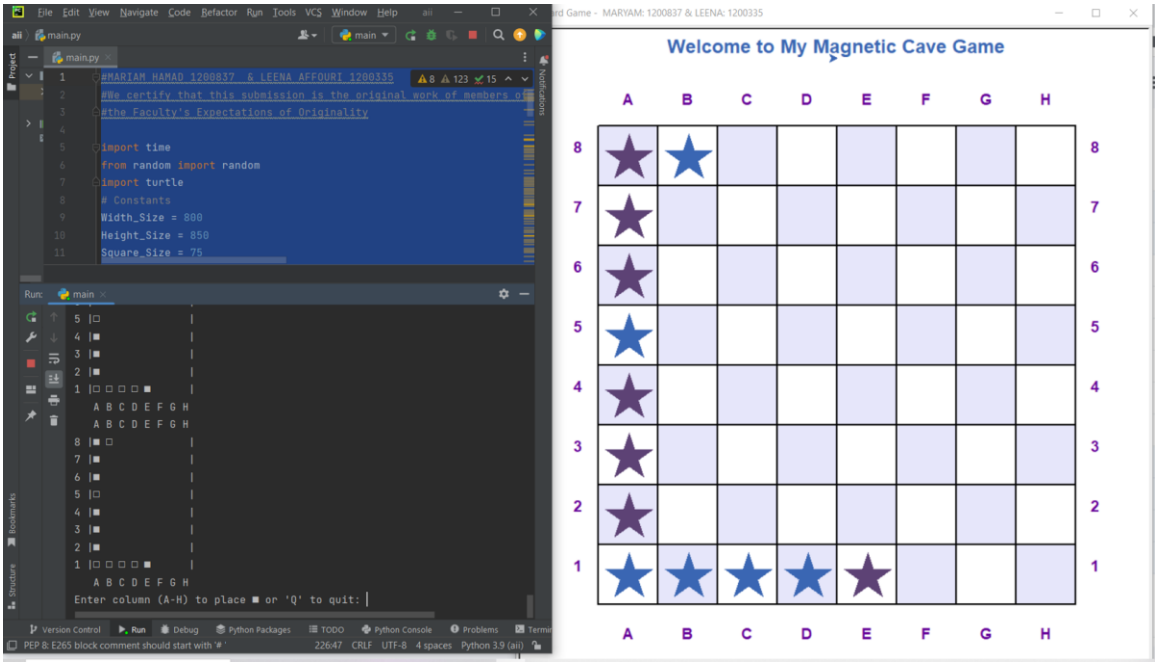


First player manual and second one is auto :



First player is auto and second one is manual :




From the result above we can see that the AI player always try to win himself and prevent me from winning, I think i lost against my opponent because the algorithm is working on prevent him from loss it take him out with minimal losses.