Ain Shams University

Faculty of Engineering

Computer and Systems Department

AI Project Report

Team Members:

- Abdelrahman Mahmoud Mohamed [43809]
- Abdallah Reda Abdallah [43812]

Contents:

Important Links:

- **Github** Repo : https://github.com/AbdallahReda/Connect4
- **YouTube** Demo : https://youtu.be/ezdxt7kZKMQ

○ A brief description the game including bonus features:

- **The game is implemented in python 3.6**
- **The game is consists of 4 code files which are:**
  - " **utility.py** " : which contain the utility function that evaluate the board and help the minimax algorithm to take the right decision based on its value (a full description for this function will be provided later)
  - " **minimaxAlphaBeta.py** " : which contain the minimax alpha beta pruning algorithm which is the core of our project
  - " **board.py** " : which contain the function which deals mainly with the board and contains of important functions responsible for printing, modifying, and checking the board
  - " **main.py** " : which contain the save/load functions , check winner functions and the main function
- **The Bonus features supported:**
  - User can set the ***difficulty*** level from 1 to 5 , where 1 is very easy and 5 is very hard
  - User can ***save/load*** the board state and can create his own board in a text file to be loaded and played later.
- **To run the game you have two options:**
  - Run the game.exe file in the executable folder and enjoy the game
  - **OR** Run the main.py file and start to play "**`python main.py`**"

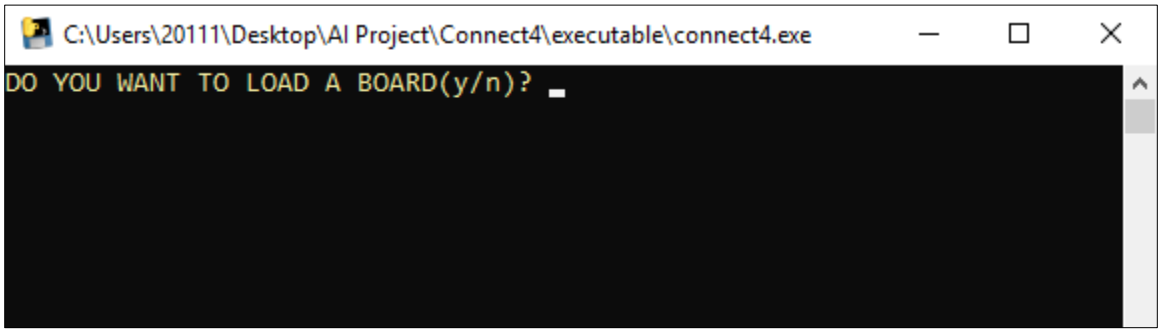
A detailed description of the utility function:

- **"countSequence" function:**
  - Counts the number of sequences of ('x' or 'o') and returns this number to the "utilityValue" function.
  - Example: we want to count the sequence of 3 x's in a row in every direction in the board so we send the board, player ('x' or 'o'), length of the sequence to the function and it returns how many sequence with this specific length is existed in the board.

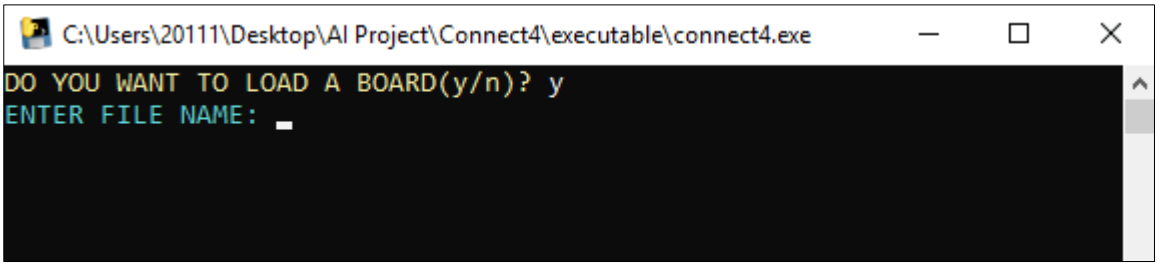
- **"utilityValue" function**
  - We evaluate the state of the board by computing the difference between the player score (Maximizer) and the opponent score (Minimizer).
  - The player score is computed by getting the number of sequences of twos, threes and fours of player's move then each sequence is multiplied by a factor (the factor of sequences of fours > threes > twos) then we accumulate these numbers of sequences.
  - The same is also done to compute the opponent score.

A user guide with snapshots:

- **First run the game from one of the following ways:**
  - Run the game.exe file in the executable folder and enjoy the game
  - **OR** Run the main.py file and start to play "**`python main.py`**"
- A console window will appear like the following image

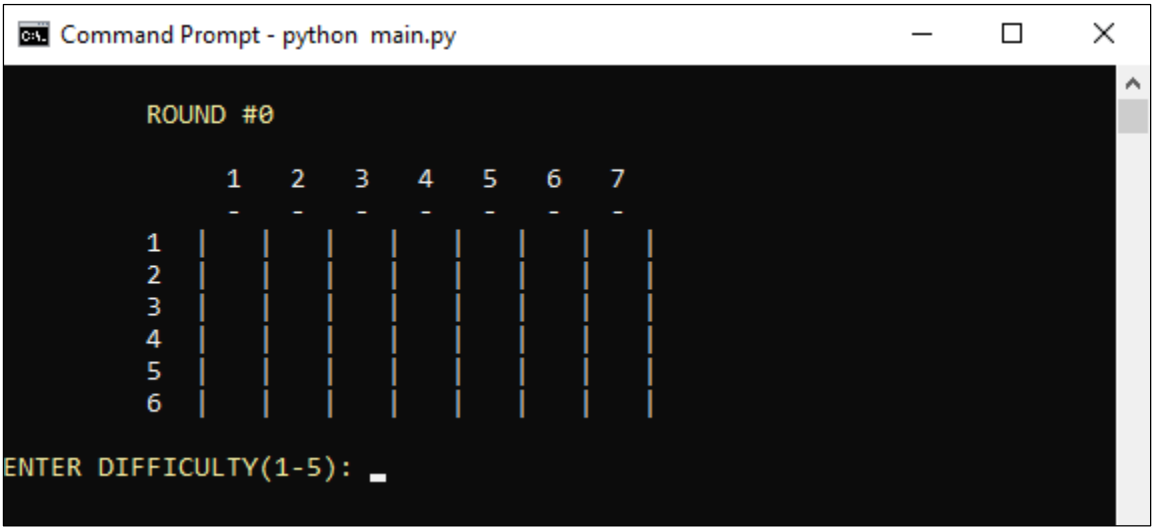


- The game will ask if you want to load a currently existing board
  - If you enter "Y" , it will ask for filename to be loaded

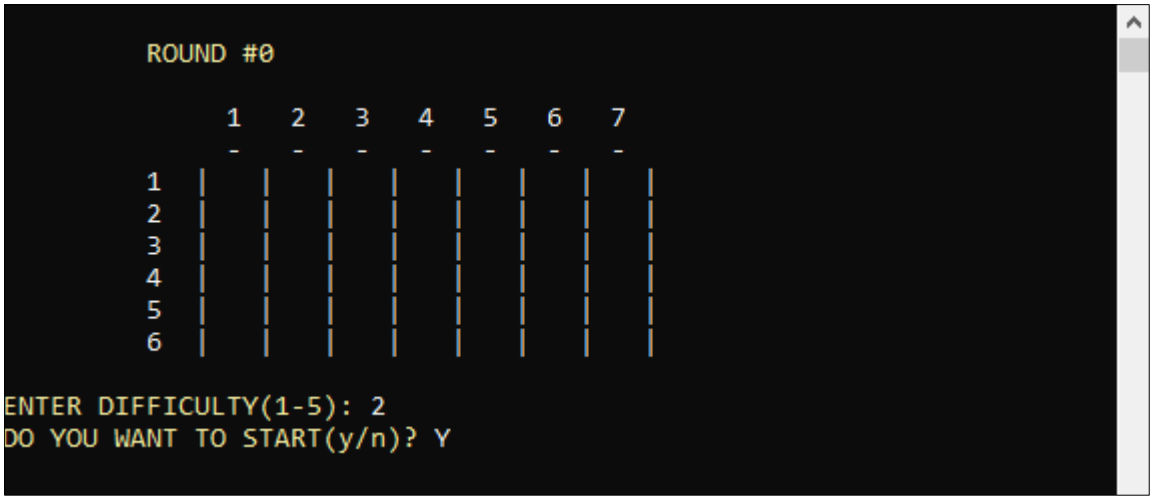


- **NOTES on LOAD/SAVE files:**
  1. The board file should exist in "**saved-games**" folder.
  2. You enter the filename **without any extensions**.
  3. You can make your own board by editing any existing file or create a text file and edit it with the following format:
     - 3.1. Blank fields in the board denoted by "**.**".
     - 3.2. Human move in the board denoted by "**x**".
     - 3.3. AI move in the board denoted by "**o**".
     - 3.4. Number of human moves **must equal** the AI moves.
     - 3.5. You should take care of pieces to be placed such that there is **no empty place** is under it.

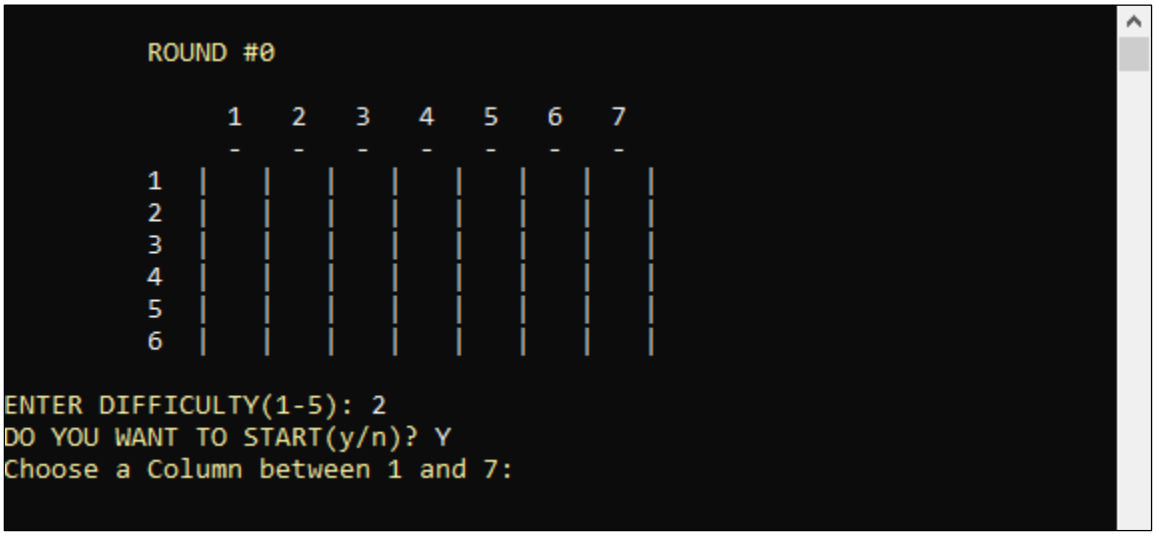
- If you enter “N” , the game will start a new game



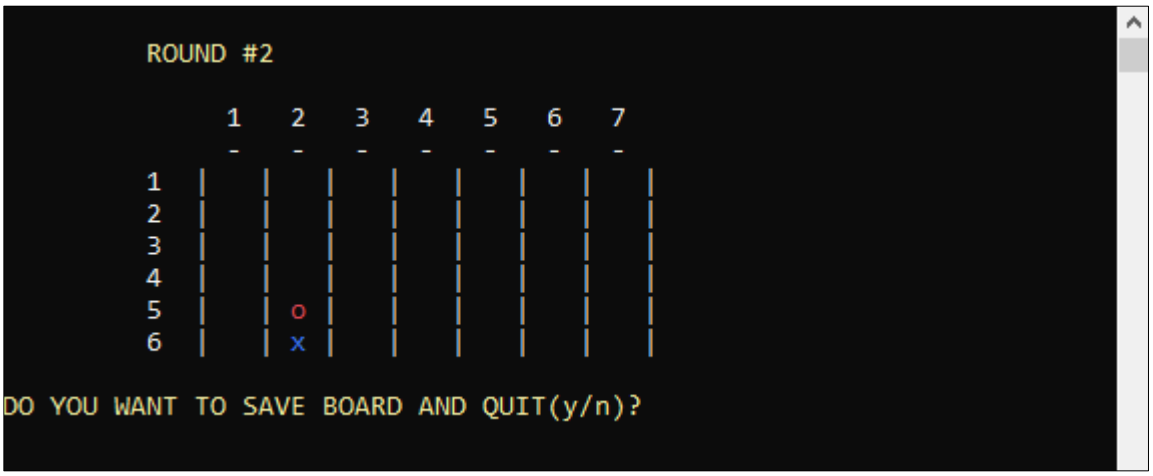
- In both cases the next question is for the difficulty level you want to play with AI, you should choose a value between 1-5 (1 for very easy and 5 for very hard).
- The Human move is registered as “x” with BLUE color
- The AI move is registered as “o” with RED color
- The game will ask you if you want to play the game first or let the AI starts first.



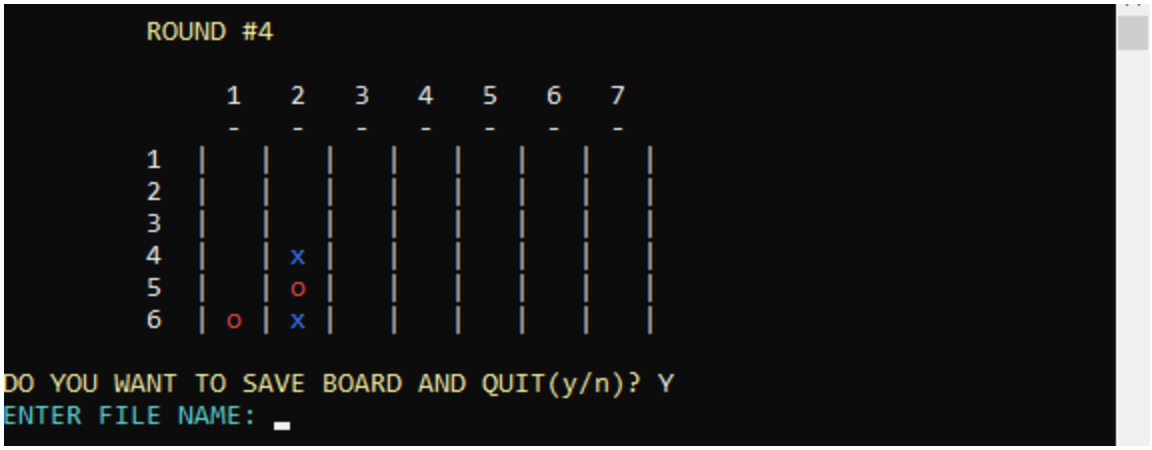
- After that you should select the column you want to play in



- You will notice that the AI plays directly after your play

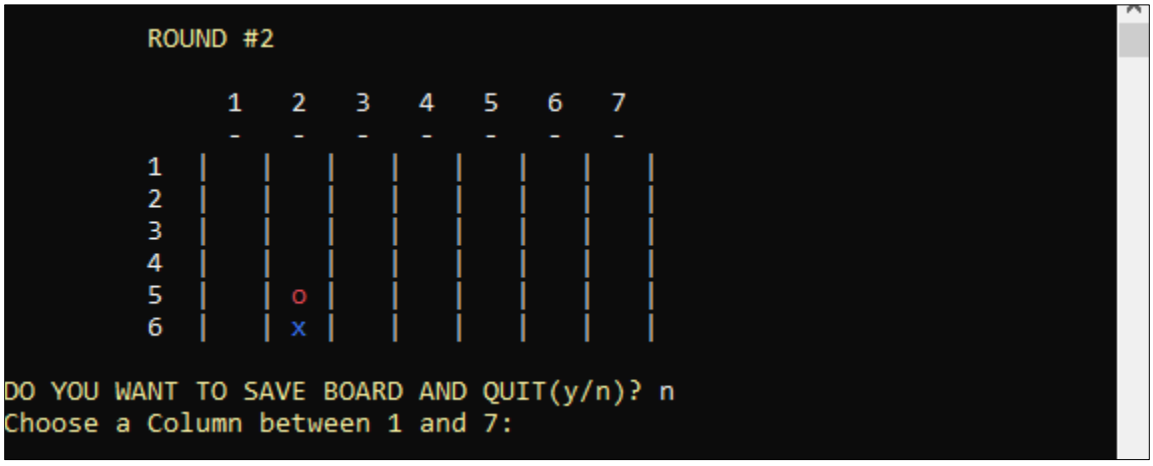


- Every time your turn comes , the game will ask you if you want to save the current board for later play or not:
  - If you enter " Y " , it will ask for filename to be saved

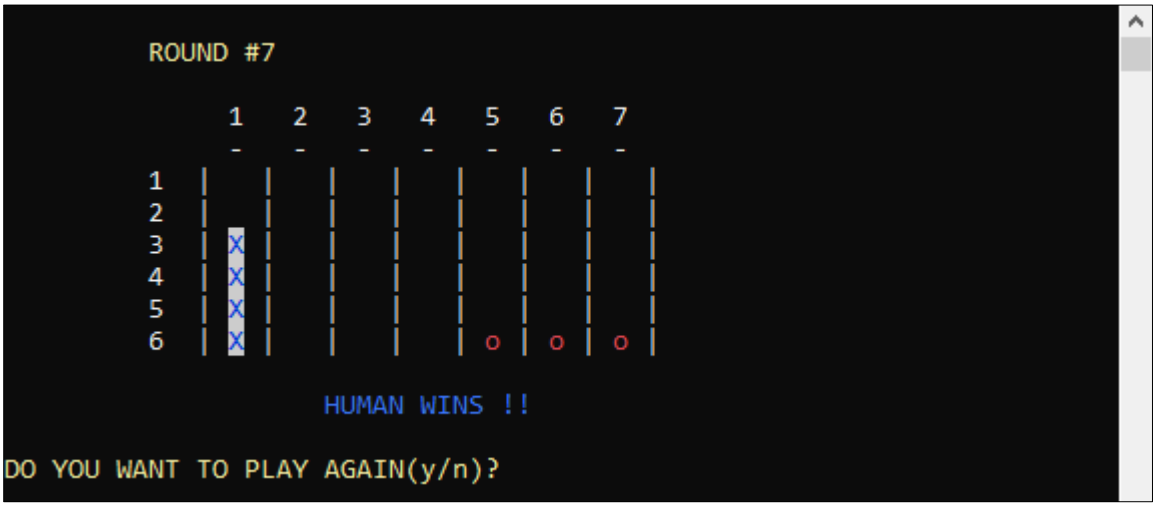


➲ **You must follow NOTES on LOAD/SAVE that explained previously.**

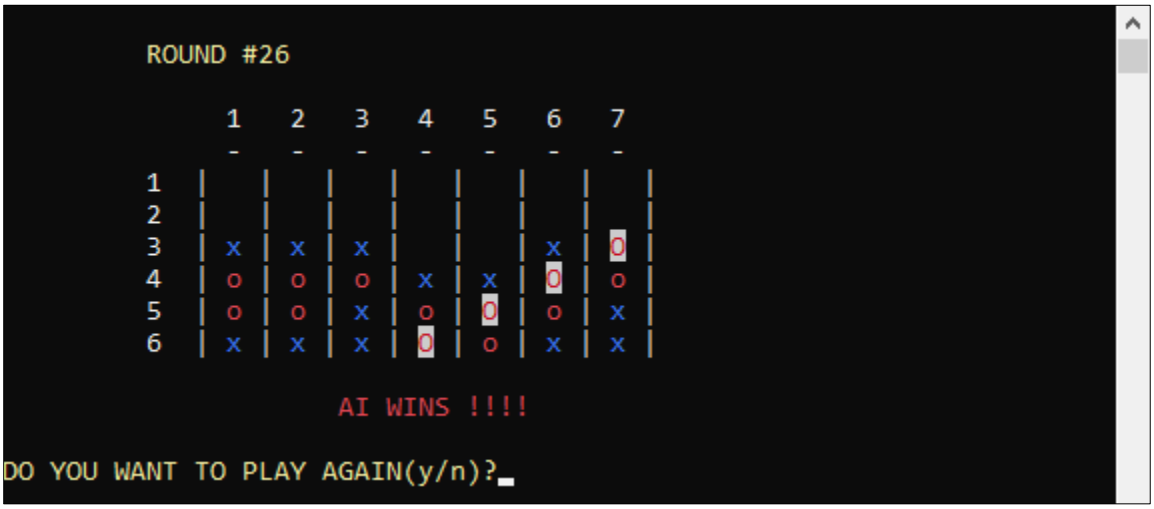
- If you enter " N " , then you should select your next move



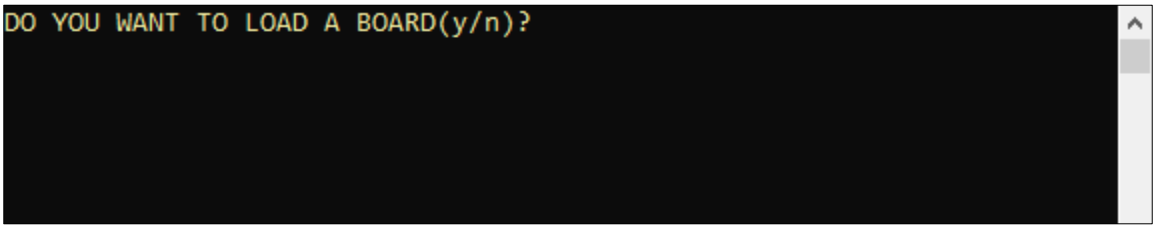
- If you want to play the game till end , then you **shouldn't** save the board and select your move until the game ends
  - If you win, the game will print "HUMAN WINS!" and it will highlight the wining pattern with white background



  - If the AI win, the game will print " AI WINS !!! " , and it will highlight the wining pattern with white background



- In both cases (either you win or the AI wins) the game is OVER and it will ask you if you want to play another game or not
  - If you enter "Y", the game will start from the begging and it will ask you if you want to load a board …etc.



  - If you enter "N", the game terminate and the console will be closed automatically.

○ A summary of how the work was split among team members

- Team Member "**Abdallah Reda**" responsible for implementing the **utility** function, **minimax alpha pruning** function and **depth bonus feature**.
- Team Member "**Abdelrahman Mahmoud**" responsible for implementing the **board** functions, the **main** function and **save/load bonus feature**.
- We both co-operate together and help each other to make a good code integration and to enhance quality of code.