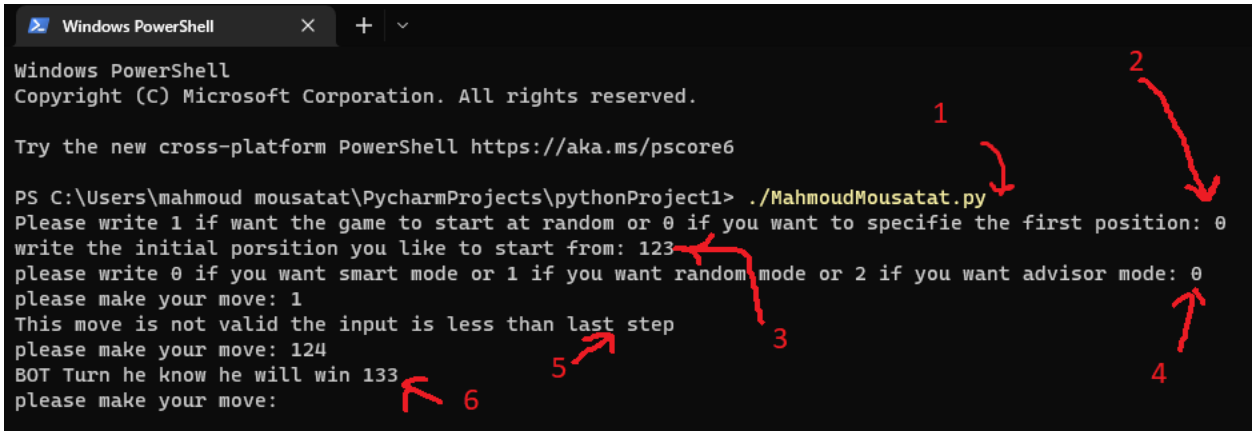
name:Mahmoud Mousatat
email:m.mousatat@innopolis.university
Date of birth: 5.11.2001
The first practice on Game theory
I divided this assignment into three parts:
In the first part I applied dynamic programming based on back tracking function, I used this algorithm to teach the bot which positions is winning for him and which is not:
The complexity of the algorithm is O(2*(final - initial)) = O(N)
Where final is the final position and the initial point is determined by the user or randmised between(0 and final)
Even the algorithm was not applied in the lecture neither labs, but I found it and interesting way to solve instead of backward induction by loops.
This algorithm use function back tracking to fill the winning position list, and by appling dynamic programic I just reduce the complexity too much.(instead of solving every position every single time I safe in list for future uses (the function of this algorithm in the code is called:
`def back_tracking(position , is1):`
In the second part I used simple random function to get the bot answer. However, it was easy to beat the bot but (believe or not, I lost once while I am trying it XD)
In the third part I used backward induction by loops to create the winning strategy list to advise the user, I spent a lot of time debugging it so unfortunately you might find the algorithm slow alittle bit, but it will give you the right answer because I fixed all the bugs.
This algorithm used techniques similar to the formula of lecture 2 slide 29
Not the same because I changed in the final position to fit the requirement.
——----------

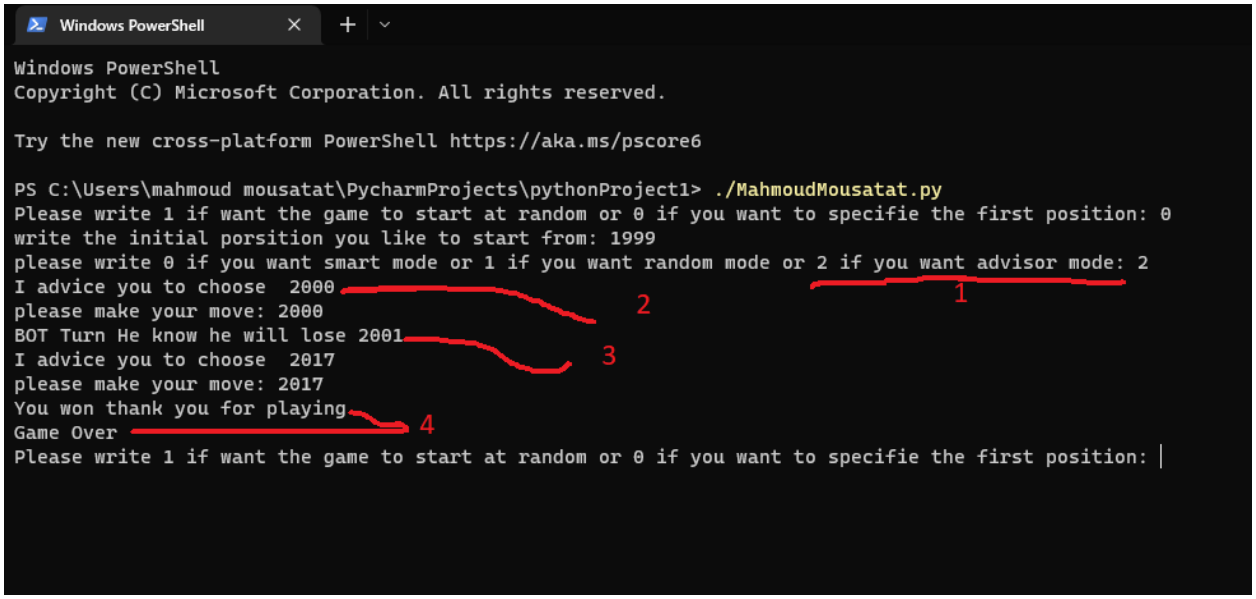
Now lets explain the interface:



1. you can run the code by running the “MahmoudMousatat.py” file from PowerShell
2. You can choose if you want the initial point to be by your choice or by random by writing 0 or 1
3. If you chooses 0 then you will have an option to write the initial point if you did not it will print an random initial point and you can start from it
4. You can choose between (smart, random, advisor) modes by typing (0 or 1 or 2)
5. After you chooses the initial position and you chosed the mode now its time to make you first move (step) by typing the next step that you would like to make

    a. If the step was in the interval (last_step, last_step + n) then the bot will take the turn
    b. Otherwise, it will show you invalid input message and will let you type the move again
6. The bot will make his move:
    a. However, if you choosed the smart mode he will show you if you still have a chance of wining or not at all because smart mode is very intelligence
    b. In random mode, there is no such option because the mode is very stupied
    c. One note, if you already choosed the adviser mode it will show you and advise message before you input your next move

An example about advisor mode:



1. here I choosed the advisor mode.
2. Because I put the first position as 1999 according to above, the advisor printed an advice message
3. After I make my decision as the advisor said the smart bot sent to me his decision with knowing that he gonna lose.
4. The I chooses the final position to win and teh program printed to me the winning message with restarting the game to play again.

Small notice, when the smart mode know that he will lose according to his algorithm what ever was his move, he will choose the smallest valid move which is “user_move + 1” because that will give hime the best chance to let the user have mistakes in the future moves and if the user made one mistake the smart bot will take it to his side and turn to winning positions for ever XD.
I thinking I covered ever thing in this assignment except the interface which has extra points, but I covered also the dynamic programing using backtracking function which is very useful tool in Game theory.
I hope I can get full grade because I am seeking for A in all my courses this semester, I will appreciate that. Kind Regards :)