

Creative Extension for Connect 4 Game

For the creative extension for Connect 4 Game I have added quite a few functionalities to the game to make it more user friendly and fun.

Choice of grid size

I added in code so that the user had the ability to choose how large the grid size is for the connect four board. In the original version they can't choose it, and it is set the moment the program is run. The user is now asked how large they want the board to be at the start of the program before the game starts.

Instructions at game initiation

I added instructions that are printed at the start of the game when the program is first run before the game starts (and before the user is asked what size they want the board to be). As the original program did not explain the extended rules of the connect four game, adding instructions in makes the game a lot more user friendly. These instructions explain how the game is connect four, but only finishes when the gameboard is filled up and the player with the most points wins the game.

Replay option

I made it possible to play again without having to run the program again. In the original program the game only runs through once and ends once it is finished. However, for my extended version I made it so the program asks the user if they wish to play again, looping infinitely until they choose to exit the game.

Slowed Text (Typing Emulation)

I have changed all the text so that when they are printed, the letters are printed one by one. This makes the user read the text printed on the screen, and it gives off the impression that there is someone on the other side typing the text out.

AI taunts

To make the game more interesting I have added AI taunts. These taunts both appear at during the middle of a game, and after each game finishes. The taunts are based off the score, and it is a randomized 'reaction' of the AI to the score, either taunting the player if the AI is winning, or exclaiming disbelief if the AI is losing. The 'reactions' are chosen randomly from a list of different possible text 'reactions'.

Leaderboard

Implemented a leaderboard function. This function allows the player to access a leaderboard, which is updated after every game. Like the arcade leaderboards. The function writes to a text file which stores the top 5 scores.

This was a relatively novel and innovative idea which I had gotten inspiration from the 90s arcade games. I tried to be retro in the style, and it matches quite well with the typing emulation functionality that I added prior. It also adds a separate goal apart from beating the AI – to get on the leaderboard.

Technical Difficulties:

Slowed Text:

Changing the text from normal `print()` to a function that I added in `slow_print()` resulted in a lot of formatting issues. The biggest issue was the `slow_print()` function used `sys.stdout.write` and `sys.stdout.flush` to output the letters, which resulted in a lack of a new line at the end of the text. Especially for user input, however that was relatively easy to resolve as I simply had to add a new line `print()` in the function.

Leaderboard Function:

This was the most technically difficult functionality that I added to the game.

I implemented a file input and output to create this leaderboard function. This was actually quite difficult, as I had to add a lot of exception handling to handle different cases, like when the file was not already existing in the folder, and if there were any issues with the file content. It was also quite difficult to process the information in the existing function, I had to determine whether or not the score the player had made it into the top 5. However, that was resolved quite easily with a for loop and list insertion. Another issue with formatting was the differences between 1st, 2nd, 3rd, 4th, etc with the differing postfixes. This was resolved by using if else case statements.

The most difficult part of creating the function was debugging and testing. This was due to the exception handling being written out in a try except manner, which made it extremely difficult to test if it worked correctly. I had to code trace every chunk of code to check for errors and debug, there were quite a few simple errors but they were difficult to catch.

Note: Please do not tamper with the `leaderboard.txt` file. This file cannot be empty. I believe this is a reasonable request, as most video games cannot have their files tampered with, otherwise some sort of corruption or error happens.

