Project 1

Tic-Tac-Toe

Description:

Tic-Tac-Toe is a multiplayer game. Whoever succeeds to place three of their marks (‘X’ or ‘O’) in diagonal, horizontal or vertical row will win the game. You can click on any of the boxes to place the sign. The first person to click the button will be an ‘X’ and the second person will be an ‘O’. First click will always be an ‘X’, second will always be an ‘O’, third will be an ‘X’, fourth will be an ‘O’ and will keep changing like this until someone wins or the boxes are full, and the game is a tie.

I had initially planned to have two options for this game, multiplayer and single player, but I ended up only implementing a multiplayer option.

I used python 3.9 and Python’s GUI (Graphical User Interface) package Tkinter for this game.

What went wrong?

I originally planned to have both multiplayer and single player. But I had a lot of trouble incorporating a single player game. First of all it was hard for me to decide whether to make the computer randomly choose from empty boxes when it's the computer’s turn, or make the computer strategize on its game play and choose positions that would increase its chance of winning. The second option would definitely make the game interesting but it would definitely be a very difficult task for me.

In the end I had decided to add weights to each box, and the computer would place it’s sign where the weight is the highest and the box is still empty. However, I don’t think the computer would have much of a defence strategy.

I did try to incorporate this to my code, but it became very complicated especially because I needed to find a way to switch between multiplayer and single player. So I would need to create two pages in tkinter. With limited time and a lot of new things to learn and incorporate I decided to not add a single player feature. So I just stuck with multiplayer. I would definitely want to expand my game and add a single player feature in the future.

Another thing that went wrong is that I wasn't able to work along with the timeline I had created. While creating the timeline I did not think about Lab 1. Because I had to spend so much time on Lab1 I was running behind for this project. However, the deadline extension really helped with this. Thanks Dave!

What went right?

First of all I was worried because I had never used tkinter before. But after I did some research I didn't have much trouble using it. With tkinter I was able to make an interactive and user friendly multiplayer tic-tac-toe game.

I also made sure to make my code consistent and neat. I documented every necessary piece of code that I wrote and I feel quite confident about it.

I also really enjoyed experimenting with different styles and colors and had a really good time building this game.

After building the game. I also did some testing. I made a couple of my friends test the game. I even played with a few of them. Then I asked for their feedback. I got mostly good feedback. One feedback that I got was that it would be nice if the player could undo their move. I thought about this a lot, about whether I should try to implement this. But I ended up not implementing this because, tic-tac-toe is a paper and pen game and if we were playing on paper, this is not something that would be allowed to do.

Things I implemented from the book:

I tried my best to be a Strategic coder, I put in extra effort to think about ways I can reduce messiness and complexity in the code. One example is that, while I was writing a function that checks who wins the game, the first thing that came in my mind was to check all the possible combinations one by one, and writing down if statements for each possible combination. But after a bit of thinking I came up with an idea to use a 2D list. I included all the possible combinations in the 2D list and looped through each possible combination to check if anyone has won. This option kept my code clean and was less work.

I also made sure to add comments for all the necessary parts of the code. Necessary parts meaning, all the parts that were not obvious. I made documentation of my code so that it reduces cognitive load and would help another coder understand what's going on in the code.

I also made sure to properly name all of my functions and variables. For example, I was Counting the number of times the board was clicked, and I just used counting as the variable name. But I remembered what the book said. It said not to use a vague variable name so instead of counting I named it num_clicks. This makes it easier for the person who is looking at the code to understand what I am counting. I also made sure to be consistent with the names I used. For example I have used _ for every function or variable name that consist of two words. Eg: all_buttons.