CSE104 Project Vanille Bourre

For this project, I did not have many ideas at first. I liked the idea of making a website, but wanted to learn more about the interaction part of javascript. That is why I decided to create games. However, I wanted to make sure I could completely finish the game I started, so I started with a bean game. I then decided to create the menu and ended with Tictactoe. I focused on easier games as Amanuel was working on a more complex game, but did multiple ones and the menu as I was more interested in css than him.

I started coding the bean game by the css and html part, which are obviously crucial to start the game, but are also easier in my opinion. That part was quite fast for me, which was good as the javascript took more time.

Making sure the buttons worked and the squares and circles appeared in the right place was quite easy. However, I could not find a function to count the number of plays done. I decided to create a list, and each action added an element to the list. This helped me to count the number of plays in an indirect way. The second problem I encountered was to create a reset button. I tried to inspire myself from the one seen in class. Mine had to also reset the players buttons, which made it more complex. My third problem was displaying a "Game Over" button that would finish the game and would disappear once the new game button was triggered. If I had more time, I would've liked to add a third button for each player displaying three dots or squares as it would have made the game more interesting. I also think displaying how many moves were left would have made the game more interesting, but I had trouble finding a function that would count the elements in "result".

Before starting a new game, since Amanuel was focusing on our table tennis game, I started creating a menu. This part was quite easy for me as I really enjoyed the css part of the course. I focused on the display and spent time on the screen change. As Amanuel's part was harder than mine for this part of the project, I decided to create a separate css file which could be added to the css of each game, and display a button to get back to the main menu. I also tried to code a css file that would enable us to reach each game we created, but as I didn't know how many games we would end up making, and it wouldn't be that useful so I only made a button to get back to the menu.

 I also decided to put pictures on the menu buttons, but the shapes weren't the same, and the menu wasn't as neat. I also did not manage to add a link behind the picture and we would have to click on the games name hidden in the image, so I decided to leave my menu with a neat display.

A small change I would make if I had more time would be to work more on the transitions once hovered.

As I had some time after finishing my part of the project, I decided to try to create a Tictactoe game. Unlike the bean game, I had trouble picturing how to sort the panel and refer to each element. I looked up the most common techniques and inspired myself from <https://dev.to/ayushmanbthakur/how-to-make-tic-tac-toe-in-browser-with-html-css-and-js-28ed>.

I used the same techniques as for my bean game, such as the reset button and the game over ideas. I spent more time trying to implement functions for multiple elements at a time as there was nine different elements each time instead of two or three in the bean game. The hardest part was to keep track of which player had played in each "boxe". I tried to add an id, but could not analyze the lines with names so I decided to put numbers which was more direct. Tictactoe was also trickier than the bean game as it could have different set ups to finish the game (with the columns, lines and diagonals). I had trouble analyzing each different possible outcome without repeating the same function over and over. That is where I used the link I referred to earlier the most. I separated the analysis in three different functions. The first one regrouped the boxes into lines, columns and diagonals. The second function analyzed if the outcome had three of the same symbol in a row. The third function kept track of the ties to make sure the game ended, even if there was a tie. After the extension, I decided to regroup the constants into variables to make the code more accessible, I worked in more details on keeping track of each boxe, and recenter the image. The last part was quite important as I could not find a cross without a backgroup, which you can observe if the image is not positioned right. I hope I managed to try every outcome to make sure the code is adapted to every situation, but it already covers most possibilities if not all.

I really enjoyed this project as it allowed me to be creative using what we learned. It is less frustrating to try and create an idea and adapt if we cannot find the exact command to do it than to have to recreate exactly something (although I understand its crucial importance in first understanding the language).

It allowed me to learn other commands that helped me bring complexity to my code. I am glad I could use what we learned in class to implement most of my ideas without encountering too many problems, and with codes that work for each game.