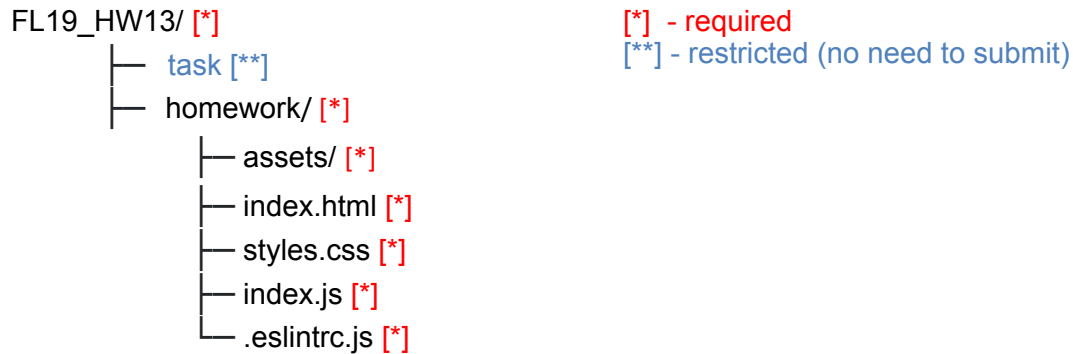


JS DOM Events

DEADLINE: 18/05/2022

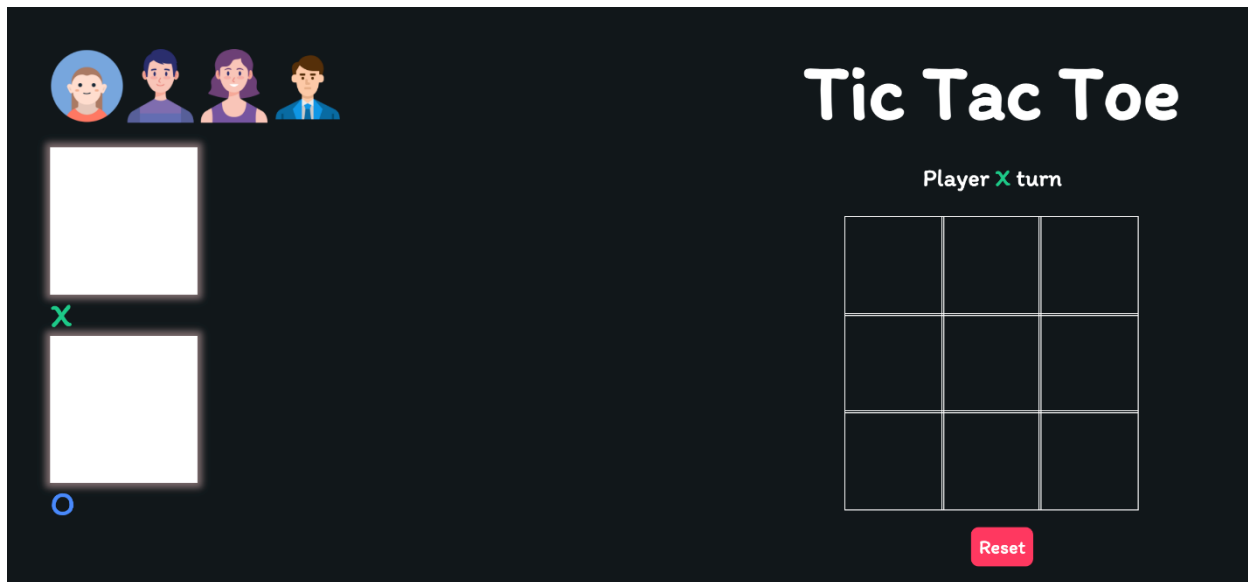
FOLDER STRUCTURE



TASK

Write all of the tasks inside the *index.js* file. *index.html* and *styles.css* already filled with code and you should not touch it. Main goal is to implement a **TIC TAC TOE (also known as crosses-zeroes)** game using vanilla JS.

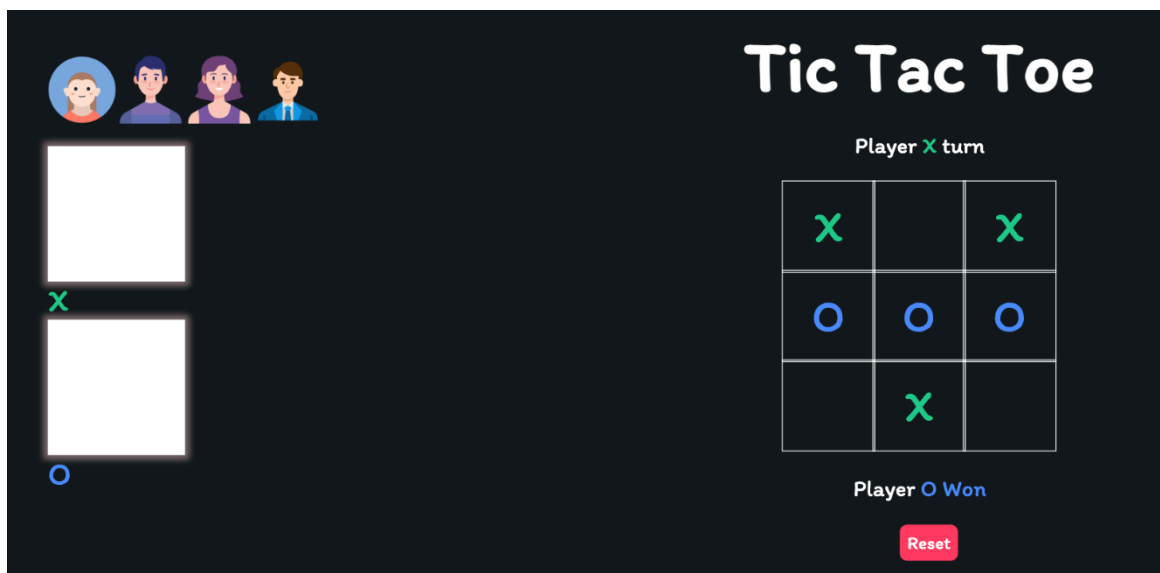
1. As you can see *script tag* is injected in the *head tag* so that means we are having some restrictions and specifics in our JS code. Game cannot be started until DOM built and ready to use. You should listen proper event and call initial game only in that case we are sure DOM is ready to use.
2. Initially game do not have any tiles but the container is in place. So your task is to generate each through the cycle. Create 9 divs and give each tile (div) css class *tile*. Below you can see how does game look like when initialized and tiles are has been generated.
3. *Reset* button should be used to reset all tiles to initial state.
4. Query all things that you need and add proper listeners to tiles, button and etc.
5. Please note, it is **required** from you to not listen each tile but the whole container (Use event delegation).
6. User can interact with tiles in two ways:
 - Using mouse to click on particular tile
 - Navigate through tiles using keyboard (We won't use here a full implementation with 4 arrows but instead a simple way with only two which are right arrow and left arrow).



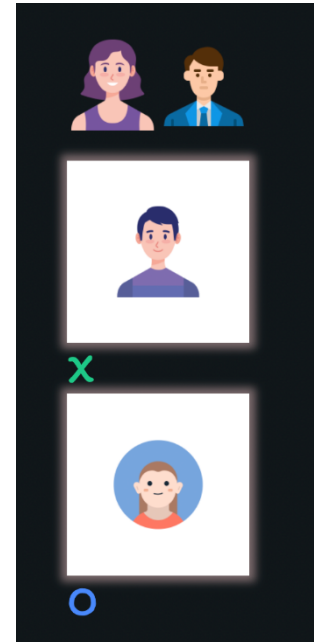
7. When user click on particular tile you are supposed to get on which exactly tile user clicked and then handle this click. To be able to see crosses or zeroes on the page you should change *innerText* property for the current tile and set X or O. To see correct styles you will need to add css class for the particular tile - *playerX* or *playerO*. After that you should change the player to O and so on.

```
▼<div class="container"> grid
  <div class="tile playerX">X</div> flex =
  <div class="tile"></div> flex
```

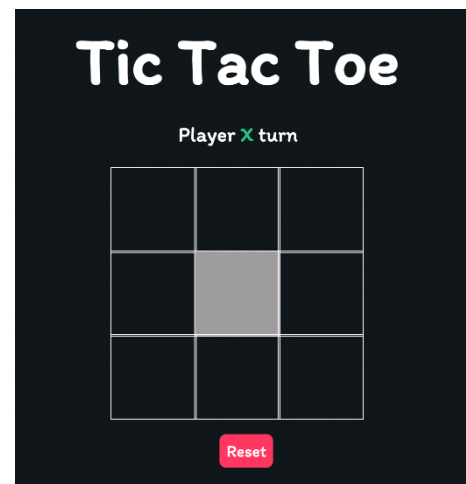
8. In case Player O or Player X won the round you need to announce the winners name and stop the game.



1. The second part of the task is implement drag and drop behavior on the left side of the screen. The expected result should be like this.
2. You would need to use existing support of drag and drop mechanism in JS and listen proper events.
3. Once the avatar was chosen and dropped into container you should block further tries to drop another one into same container. Only one avatar could be chosen.



1. The last part of the task is to implement navigation using keyboard as was mentioned before. Allowed arrows to navigate are *right and left*. *Enter* is for submitting your choice. Behavior should be all the same as for mouse clicking. When you are moving to the right you need to highlight the next element in game container, and if you moving left - highlight the previous element.
2. Create costume event to handle *Enter* key. Then let each tile to listen this event and dispatch when *Enter* key is pressed.



BEFORE SUBMIT

- Code should be clean, readable, and tested
- Verify that all functionality is implemented according to the requirements.
- Run the linter and fix all the warnings and errors.

- Make sure your GitLab folder structure meets folder structure from this document (without **task** folder)

SUBMIT

- The **FL19_HW13** folder without **task** folder should be uploaded to GitLab repository “**FL-19**” into **main** branch.

USEFUL LINKS

- <https://developer.mozilla.org/en-US/docs/Web/API/CustomEvent/CustomEvent>
- <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>
- <https://javascript.info/event-delegation>
- <https://javascript.info/dispatch-events>
- <https://javascript.info/keyboard-events>