OOP Intro

DEADLINE: 13/06/2022

FOLDER STRUCTURE

```
FL19_HW3/ *                                        *  - required
    ├─ task
    ├─ homework/ *
         ├─ index.html *
         ├─ .eslintrc.js *
         ├─ js/ *
             ├─ index.js *
         ├─ css/ *
             ├─ index.css *
```

TASK

Warcraft 3:

You'll need to implement 3 classes: a ***Game*** , a ***Unit*** and a ***Display*** classes. The first time You load the page you have to create a header with the name of the game and a start button that will start the game. When You press the start button, an alert should be displayed with the 'Choose your fighter' sentence. After the alert, a list of characters should be displayed as selectable buttons. The start button must be hidden. You need to choose a character. the second player will be a computer that will select a random character from the list after You have selected a character. Then the list of characters disappears and the 'Fight' button appears. Also there should be displayed two containers with the name and stats of the character. as well as his current health and what will be deducted. After You press the Fight button the characters will attack each other and take away health considering attack rate and armor. If one of the characters health falls below zero, he loses. After the match there should be displayed an alert with the victory of the player. After accepting the alert, the player must return to app to the initial state.

Game Class Description

Implementation Details:

This is the main class and this class is responsible for the user's actions.
For example choose player.

Unit Class Description

Implementation Details:

This is a constructor class for unit creation and unit actions.

Display Class Description

Implementation Details:

This class is responsible for visual manipulation.

HELLPFUL LINKS

- https://wowpedia.fandom.com/wiki/Footman_(Warcraft_III)
- https://warcraft3.info/articles/208/overview-of-armor-and-damage-reduction

RESTRICTIONS

- Editing index.html is forbidden
- You don’t need to spend much time on implementing design of the application
- You need to use only classes

BEFORE SUBMIT

- Remove all unnecessary files that you might have included by mistake
- Verify that all functionality is implemented according to requirements
- Make sure you code is well-formatted, and validated via validator (w3org Markup Validation Service)
- Add comments if the code is difficult to understand
- Fix warnings/errors in the browser console
- Verify that the name of the folders and files meet the requirements
- Make sure there are no errors/warnings in the browser console
- Run the linter and fix all warnings and errors.

SUBMIT

- The folder should be uploaded to GitLab repository 'FL-19' into main branch