Project Grading Guidelines

Here are the guidelines we apply when grading your project work. The purpose with these guidelines is two-folded:

- You as student get some idea about what you need to do in order achieve a certain grade.
- We as examiners know what to look for when we grade your work.

However, what is written in this document should only be seen as *guidelines*, and not as a checklist for a certain grade. We make an assessment of your entire work, which is not limited to these guidelines.

In the guidelines below, *including* does not imply *limited to*.

For grade 3, the student shall:

- Use semantic variables (one and the same variable should not be used to represent two different values).
- Use mnemonic names (both in JavaScript, HTML and CSS).
- Properly indent the code (stick to one convention).
- Write the code in a structured and maintainable way.
- Implement the program per the specification, including:
  - Having all specified functionality fully implemented and fully functional.
  - Preventing the user from doing strange things, e.g. uploading score to the server without being logged in.
- Handle and present descriptive error messages to the user, including:
  - Validation errors.
  - HTTP responses with a 4** status code.

  Note: status codes do not count as "descriptive error messages".
- Create at least one HTML file, at least one CSS file and at least one JavaScript file.
- Write everything (including code, comments and filenames) in English.
- To some degree follow the Web Content Accessibility Guidelines, including:
  - Proper usage of HTML (use tags to mark what type of data text represents).
  - Presenting none-text content as text as well (using the `alt` attribute for `<img>`).
  - Providing meta information about the web page (using the `language` attribute for `<html>`).
- Somehow make use of a JavaScript library (e.g. using jQuery to register listeners for events).
- Somehow make use of an HTML & CSS framework (e.g. using bootstrap).
- Upload the website to a folder with the same name as the student's JU id on a server provided by us.
- Submit the Ping Pong assignment *Project Work,* and:
  - Attach a ZIP file containing all the files the website consists of.
  - Attach a comment containing the full URL to the HTML file on the website on our server the user should arrive to first.
- Build the website so it passes the grandparent test (it should be obvious/described how to play the game, what happens when you click on a button, what error messages actually mean, etc.).

For grade 4, the student shall:

- Follow the guidelines for grade 3.
- Stick to one naming convention (both in HTML, CSS and JavaScript, although the convention used may differ between the languages).
- Avoid using hard coded values in the middle of the code (instead, use variables (constants) at the top of the code to store these values).
- Have code ready for delivery to customer, including:
    - Nothing is logged to the console while running the code.
    - No errors occur while running the code.
    - No code is commented out (if you do not need it, remove it).
- Have well written comments where it helps others to understand how the code works (Note: if one uses mnemonic variable names, the code tends to be quite self-documented and not in need of that many comments).
- Clearly separate the code used for presentation from the code used for logic computations, including:
    - Avoid placing JavaScript code inside HTML code.
    - Avoid using the `style` property in JavaScript (if possible, use CSS classes instead).
- Handle and present all error messages to the user in a fancy way (do not use the `alert` function; use the DOM to change the GUI).
- Validate and fix common HTML and CSS errors using a Markup Validation Service, such as:
    - W3C's validator: https://validator.w3.org
- Validate and fix common JavaScript errors using a JavaScript Code Quality Tool, such as:
    - JSLint: http://www.jslint.com
    - JSHint: http://jshint.com
    - ESLint: http://eslint.org/demo

For grade 5, the student shall:

- Follow the guidelines for grade 4.
- Avoid introducing new global variables, when possible and suitable.
- Make use of the prototype chain (classes do not exist in JavaScript in the same sense as in the most common object-oriented languages, but we can use constructors and the prototype chain to model things in a similar way), e.g. for:
    - Dice
    - Set of dice
    - Game
- Generalize solutions to make them more useful/easier to reuse in other projects.
- Structure the code in such way that it is very easy to reuse different parts of the code in other projects, including:
    - The game logic.
    - The validation logic.
    - The server communication.
- Solve encountered problems in a very good way and implement the solutions in an excellent way, including:

    - Avoid redundant/dead/unreachable code and code bloat.
    - Use functions instead of code duplication.
    - Write code that is easy to read and easy to understand.
- Dynamically hide/show/disable components in a way that simplifies navigation, e.g.:
    - Not showing the sign out component unless the user is signed in.
    - Not showing the sign in component when the user is signed in.
    - Disabling the "Play turn" button when the user's entered guess is invalid.