Rubric web technology assignments

This is an overview of the rubric the teaching assistants use to assess the web technology assignments.

A single assessment session is held per group for the three web assignments. Every web assignment is assessed as either passing or non-passing. Although groups of two students are assessed, the passing/non-passing grades are handed out per student. It is thus possible for one team member to pass all three assignments and for the other to fail all three assignments.

The rubric below contains example questions/tasks for each assignment **as well as an overview of the requirements** to achieve a pass. To answer the questions, students can always consult their notes and code.

Note: any required Brightspace uploads that did not happen before the assessment session can still be executed during the assessment session.

Table of Contents

Assignment HTTP+Design

Possible questions / tasks

- Show off a `HEAD` and `GET` request via telnet. What is the difference between the two?
- Execute the `PUT` request.
- What is the difference between authentication using telnet vs. a browser?
- What does the site's Cache-Control directive mean?
- Show off the board game design.
- Show off the four board game examples and discuss a positive and negative feature.
- Show off the HTML. How did you decide to use 'those tags' for your board game?

Passing requirements

- The student is able to answer most of the TA's questions correctly.
- The board game design is available on the Brightspace forum. The deliverables are uploaded to Brightspace.
- The splash / board game design fulfils the board game requirements (listed in the first web technology assignment).

Assignment JS+Node

Possible questions / tasks

- Show off the application's `package.json` and explain its purpose.
- Start the server and open the browser to play the game (splash screen has a play button, N players are randomly paired to play the game, the players can execute moves).
- Explain randomly chosen functions in your code.
- Explain the data structure(s) used to keep track of the game state on the server.
- How was it made possible for several players to play games at the same time?
- If `jQuery` was used on the client-side ... what does `$` refer to?
- Discuss the design pattern(s) (we discussed three in class) chosen for the game.
- Show off the messages client and server exchange to communicate the game state and discuss advantages/disadvantages.

Passing requirements

- The student is able to answer most of the TA's questions correctly.
- The required client-side and server-side components are implemented.
- The communication is based on WebSockets.
- One of the three design patterns discussed in class is used at least once.
- The deliverables are uploaded to Brightspace.
- The game works as intended: players are randomly paired, valid moves can be executed.

Assignment CSS+Node

Possible questions / tasks

- Start the server and open the browser to play the game (splash screen has a play button, N players are randomly paired to play the game, the players can execute moves, **invalid moves are rejected**).
- Show off the implemented CSS animation and explain the code.
- Show off the use of templating and explain the code.
- Show off the use of two different position attributes.
- What is the difference between a CSS animation and transition?
- How are the board game tiles styled with CSS? How are the movements of game elements implemented?
- Show off the employed media query and explain its use.
- Show off the fullscreen game functionality and explain its use.
- Explain the cookie code.

Passing requirements

- The student is able to answer most of the TA's questions correctly.
- The deliverables are uploaded to Brightspace.
- The implementation contains the necessary CSS elements (A3, 1.1).
- The splash and game screen have a consistent design.
- Templating (EJS) has been used.

- At least one client-side cookie has been implemented.
- The game forbids players to make invalid moves.