Frontend Programmer Assignment

Welcome, and thank you for applying for the Frontend programmer position at Exordium Games.

Please read the entire document carefully before you start the assignments. Good luck!

Instructions

Please adhere to the following syntax instructions:

- Variables → use variableName notation
- Functions → always CamelCase notation
- Constants → use CONSTANT_NOTATION

This assignment comes with a couple of optional ("Bonus") features. They are marked with the starting “Bonus” text after the feature number. If a feature does not have the “Bonus” text after the number, even between “Bonus” features, it is an obligatory feature. Each additional Bonus feature will be appreciated and valued.

External resource usage

- You can use any graphics you find useful from online sources
- You can use any jQuery Core (preferably 3.6.1) and jQuery UI (preferably 1.12.1) version
- Advice: check the accessibility, functionality, and testability of all the data you send from another computer (files, means to access repository)

Assignment specifics

- Use prototype inheritance in assignments 1 and 2 for the code regarding interaction buttons logic
- In assignments 1 and 2 you will need 2 colors to indicate the correct and wrong answers. Please use the following:
    - correct color - #77dd77
    - wrong color - #FD6861
- Other colors that you will need to use:
    - “Theme 1”:
        - primary color: rgb(253, 178, 89);
        - secondary color: rgb(230, 172, 96);
        - highlight background color: rgb(248, 243, 199);
        - background color #fff;
        - text color: #333;
    - “Theme 2”:

- primary color: rgb(63, 207, 152);
- secondary color: rgb(81, 246, 152);
- highlight background color: rgb(245, 255, 250);
- background color #fff;
- text color: #333;

Assignments

1. Multiple answer question

1.1. Minimum 3 answers in total

1.2. Minimum 2 correct answers

1.3. Three buttons below the interaction:

1.3.1. “Check” button -> to check if the selected answers are correct

1.3.2. “Reset” button -> to reset selected answers

1.3.3. A third button “Show Answers” is hidden at first, but if the user clicks 2 times on the “Check” button while some answers are incorrect, the “Show Answers” button appears between the “Check” and “Reset” buttons. The text on that button is “Show Answers” which highlights all the correct answers when clicked.

1.4. Selected answers’ style/design change when the “Check” button is pressed

1.4.1. If the answer is correct - the background is changed to the correct color and the text to white

1.4.2. If the answer is wrong - the background is changed to the wrong color and the text to white

1.4.3. Bonus: add a green checkmark or a red cross next to the selected answer

2. Drag and drop element sorting

2.1. Create the first container that initially contains all the elements that need to be sorted into other containers

2.2. Create two additional containers where elements from the first container need to be dragged and placed (Can be named “Summer” and “Winter”)

2.2.1. Bonus: Four containers (additional ones - “Spring” and “Autumn”)

2.2.2. Bonus: add an image of your choice as a container background

2.3. Text for the elements that need to be dragged and their corresponding containers:

- Sea (“Summer”)

- Ice cream (“Summer”)
- Snowman (“Winter”)
- Skiing (“Winter”)

2.3.1. Bonus: text for elements for bonus 2 containers:

- Flowers (“Spring”)
- Blossoming (“Spring”)
- Leafs falling (“Autumn”)
- Chestnut (“Autumn”)

2.4. Three buttons below the interaction

2.4.1. “Check” button -> to check if the dragged elements are in the correct container

2.4.2. “Reset” button -> to move all of the elements to the initial container

2.4.3. A third button “Show Answers” is hidden at first, but if the user clicks 2 times on the “Check” button while some elements are not in the correct container, the “Show Answers” button appears between the “Check” and “Reset” buttons. The text on that button is “Show Answers” which, when clicked, moves all elements to their correct container.

2.5. Placed elements style/design change when check button is pressed

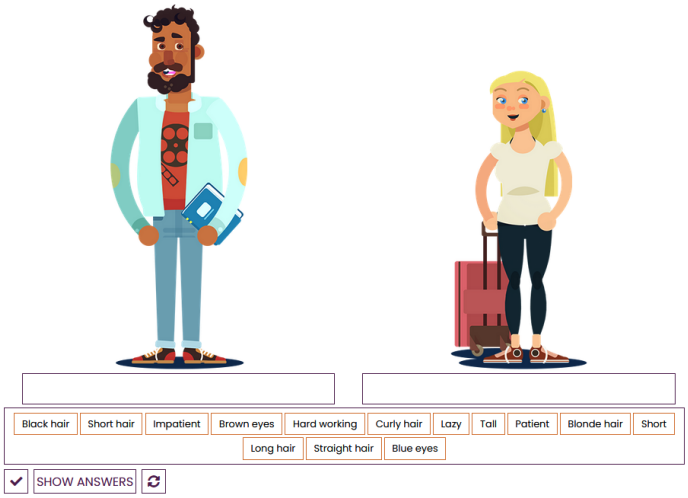
2.5.1. If an element is in the correct container - its background is changed to the correct color and the text to white

2.5.2. If an element is in the wrong container - its background is changed to the wrong color and text to white

2.5.3. Bonus: add a green checkmark or a red cross next to the element

2.6. Make sure that the interaction is responsive, as it will be tested on multiple aspect ratios (eg. Desktop 16:9, Mobile portrait 9:16).

Link to the example interaction



Example interaction image

3. 3D model display and interaction

3.1. Using the third-party plugin of your choice (examples: A-Frame, ModelViewer) show any 3D model on the page.

3.2. Ability to rotate the 3D model by dragging a pressed left mouse button over its display canvas

3.3. Bonus: center 3D view on the page with a width of 75% of the total screen width

3.3.1. Add description text below the 3D view matching its width (use “Lorem ipsum” template text paragraph)

4. Bonus: Theme change

4.1. Create a simple drop-down menu with 2 options: “Visual Theme 1” and “Visual Theme 2”. Colors for themes are defined in the Assignment specifics section.

4.2. When another “Visual Theme” option is selected, the colors on the page need to change accordingly

5. Data display and filtering

Create a single HTML web page, separate from the previous assignments, that will display data about a predefined set of games, along with filtering the games with the user-input criteria defined below.

Create the page with the following elements and their functionality (along with the required .js files, and CSS at your leisure)

5.1. When the HTML page is opened, all elements should be in their default state - specified below per element, and data about all the games should be visible.

5.2. Dropdown to choose from the following text and data

5.2.1. “Dataset 1” - Data from https://exordiumgames.com/jobs/frontend/dataset_1.json

5.2.2. “Dataset 2” - Data from https://exordiumgames.com/jobs/frontend/dataset_2.json

5.2.3. On dropdown value change, all game data should be loaded from the selected file, and all filters reset

5.2.4. When the HTML page is opened, Dataset1 should be loaded

5.2.5. All image URLs in JSON files are relative to the path https://exordiumgames.com/jobs/frontend/, for example, game01.png is located at https://exordiumgames.com/jobs/frontend/game01.png

5.3. “Reset all filters” button - on click all filters should reset to their default value, and all games from the current dataset should be displayed

5.4.1. A row containing: “Filter games by name” text, input text field, and “Reset input” button

5.4.2. On entering the text, only the games containing the entered text in the game name (ignore case) should be displayed.

5.4.3. On button click, the text input field is set to empty text, and game filtering is updated.

5.5.1. Vertical scroll rectangle of clickable toggles displaying all unique game genres from the currently selected Dataset, in the format of (“Genre name” [checkbox]) per element

5.5.2. The toggles are dynamically created depending on the required count.

5.6.1. Vertical scroll rectangle of clickable toggles displaying all unique game styles from the currently selected Dataset, in the format of (“Style name” [checkbox]) per element

5.6.2. The toggles are dynamically created depending on the required count.

5.7.1. Vertical scroll grid of rows containing at most 3 game display elements per row, where 2 rows can fit vertically to be visible at once inside the scroll rectangle

5.7.2. A single game display element consists of:

- text containing the Game name
- Game image (from URL in the Dataset)
- text “Genre:” + Game genre value
- text “Style:” + Game style value

Solution submission

During the assignment execution, you should use subversioning development:

- Create a new **private** online repository (for example: on Github)
- Develop the project so all the minimum relevant files are committed and pushed to the repository
- Use branches for separate interactions
- Maintain a branch named “development” where all features will be merged when completed
- Have a “production” branch where the final project version will be merged for our testing

In your response mail to jobs@exordiumgames.com you should provide the following:

1. **Attachment Docx document named “Exordium_FrontendProgrammerAssignment_WebInteractions_dd.mm.yyyy.docx”** - which is a copy of this document, with a green highlight of the text for each point you consider solved completely, yellow for partially, and red for unsolved.
2. **WebInteractions.zip** containing a .zip of all needed project folders (same as the last state pushed to the subversioning repository - detailed below)

- Each interaction must be in a separate folder that will contain:
    - folder with all media files (images, videos, etc.)
    - index.html in the root of the folder
- shared CSS and javascript scripts can be located in the parent folder or a separate sub-folder

3. **WebInteractions_README.txt** which explains the testing and grading guidelines for your solution and which script is relevant for each subtask point.
4. **Access to the repository**
    a. Either a message that you invited andrija.stepic@exordiumgames.com to the online repository
    b. or the link and credentials required to access the online repository
5. **Running instructions** - write the instruction text to Instructions_N.txt file inside the folder containing all the data required for the assignment with number N on how to run its HTML file if there are some special conditions and requirements, and the HTML won’t work when locally started offline.

For all questions and unclarities about the assignment, contact andrija.stepic@exordiumgames.com

Good luck and happy coding!