The Card Game

AOE Card Game

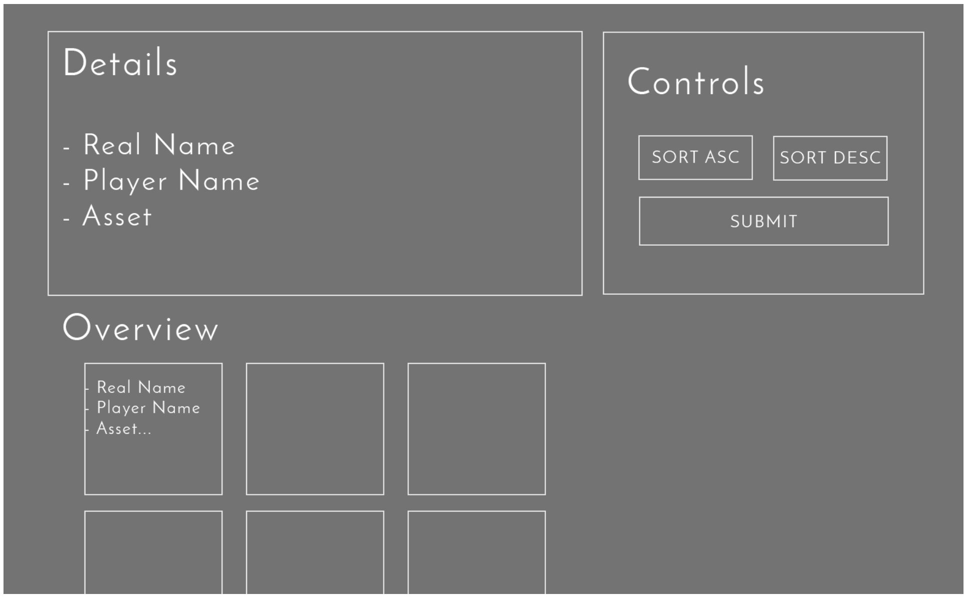
The Story

In an agency far away a PO had a product vision.

Because her dev skills are limited, she immediately turns over to you and asks you to develop a MVP, which shall feature her vision.

Her vision is a card presentation. She provides you with some text and a very hasty first draft (screenshot), which needs changes:

Small cards display information about players that participate in a card game.



Card data

Real Name	Player Name	Asset
Brianna Forbes	Dreamlurk The Unstoppable	Foghammer Lead
Darcy Candice Ball	Crystaldash	Secret Glowquake Gold
Hillary Gibbs	Speedsoul	Shifting Rainshadow Iron
Elva Becky Hammond	Seekvenom The Mystic	Valkyries' Opal Adamant
Enid Rose	Coincurse The Ghoul	Jewelevil Bronze Of Goddesses
Esmeralda Carrillo	Skulldart	Yellow Orichalcum Of Paladins

Requirements

The game is a single page application (SPA) where all actions and visualizations are shown on one page and mainly implemented in plain JavaScript/Typescript, the newer, the better. The MVP actually targets only desktop environments and here the latest browser generation.

Views

Overview

The PO thinks that there shall be only three cards per row. Also she thinks there should be no line breaks for the text within a card in the overview. Instead there shall be three dots (...) to indicate, that the information is larger than horizontal space available. Only the large card (details view) shall show the full information even with line breaks if not enough space is available. Once a card from overview was selected, it's entirely shown in the details view and also highlighted to indicate what's currently selected. It's also necessary to have some indicator for the user that shows interactions are possible.

Details

Initially when noting is selected the details view is hidden. It's important for the PO to avoid side effects like a "bumping" overview, once a card item was selected.

Controls and filter

There is one more section which holds all the UI elements for user interaction explained here (see functionality)

Functionality

Sorting

There should be functionality to sort the cards by name in ascending and descending direction. Initially when the app is bootstrapped the cards should be shown as they come from JSON response (same order as in the table).

Submitting

There is also functionality required to submit the currently selected data from details view using a mocked AJAX request. Here it's important to submit the data as JSON to the mocked backend.

Implementation

The last information you heard from the PO was, that you shall only implement the MVP and that your colleagues will take over this piece of code and extend it in another iteration. Equipped with this information, you start developing right away.

After a short meeting you agreed with your colleagues on the following action points:

Must Haves

- Build out a basic modular folder structure
- asynchronously load the player data in json format
- Build an object model for the player data
- Provide the required logical views as in the explained image

Furthermore you follow best practices like

- Separate concerns
- Avoid globals
- Use generic naming for both JS and CSS
- Use semantic HTML5

Things we usually do in our projects

- unit tests
- static code analysis
- create a production build while maintaining the development code
- and/or generally feel free to provide a setup which is positive (from your understanding) for the upcoming iterations when others take over the project and continue working on your implementation.

Development hints

Beside all the requirements above one useful hint is to develop the game in a manner where you would like to continue working on — *without the sentence "we need a complete rewrite ;-)"*.

Feel free to choose the frameworks and libraries you think are suitable for this project.

Since there is **absolutely nothing to do on the backend**, simply focus on the frontend.

Focus

Depending on your personal skills, you can focus more on the one or other aspect.

More Frontend less Javascript

If you're more focused on perfect markup, great styling and amazing usability etc. feel free to differ from the above mock in terms of layout, animation, responsivity etc. Keep the logical views such as overview, details and controlls/filter and provide us with an unique UI/UX. We're exited what you'll provide.

More JavaScript less Frontend

If you're more focused on JavaScript (software) development, feel free to use the mocked "design" above and provide a great peace of software. We're super exited and looking forward to get impressed by your solution.

Conclusion

Feel always free to ask if things aren't clear enough — *we'll always have an open ear and will improve this test by our applicants feedback*.

Now it's time to start coding and we (the AOE-frontend devs) hope you will enjoy this challenge :)

Enjoy :)