

Introduction

This document is part of the second mandatory assignment in the course Front-end Development held in Spring 2016 at BAAA - Business Academy Aarhus. * Link to source code on GitHub * Link to Heroku app

Case description

- Is this okay to include?

User Persona

Julie 23, single, studying Media Science, lives in Aarhus, wants to live in Copenhagen, Northside-fan, listens to P6 Beat (indie cool), changes for P3 (mainstream), when alone. Loves Instagram, just began on Snapchat, reads Danish fashion blogs, reads eb.dk for news, Zulu & DR3 as flow tv, has iPhone 5s, watches House of Cards, because people say it's cool, watches Game of Thrones, because she loves it & read all of the Harry Potter books.

Designer Persona

Peter 28, single, graduate from London School of Arts as Illustrator, lives in New York, illustrated his first children's book, been at Hummel as tshirt print designer, pursuing career as professional street artist. Got prints chosen on Threadless.com, has his own webshop with t-shirts he sell through Spreadshirt.

Design requirements

Create a Style Guide for the public website to Squares. The Style Guide must include:

- Target group description based on the personas acquired by Kong Orange. Comment on the personas and the target group they hint at. What values do they possess etc.?
- A colour palette with examples of colours and an explanation of your choices.
- Your choice of Font and an explanation of your choices.
- A Gestalt sketch and description of composition. Explain why you chose to organise your site in this particular way, knowing what you do about users and functionality.
- Explain how your design choices supports the sites content.

Technical requirements

You have to utilize technologies from the front end course: HTML5, CSS3, CSS preprocessor (i.e. Sass) and JavaScript. Using jQuery and jQuery plugins is voluntary and you are allowed

to use a CSS framework like Bootstrap. It is also voluntary to use automation tools like ex Gulp. You are allowed to use JavaScript frameworks like AngularJS.

In the report, you must document and explain how the following elements are used in your solution: * Semantic HTML * CSS3 and use of CSS preprocessor * Central JavaScript algorithms * Possible use of CSS framework * Possible use of jQuery and arguments for use of possible plugins * Possible use of JavaScript framework, i.e. AngularJS * Possible use of automation tools, i.e. Gulp * Browser compatibility * Validation of HTML and CSS

SQUARES

This section will shortly describe how the SQUARES business case is interpreted and how it will be supported by the website.

Everyone will be able to upload square sets to SQUARES - the only requirement is that you need to be logged in, as square sets are uploaded through the profile page and every square set will be linked to a user account. The poorly designed square sets will be filtered out using a rating system. For this prototype the rating system will be rather simple, but will ensure that the highest rated square sets and artworks will be shown first.

The users of SQUARES will be incentivized to upload quality looking square sets, as this is the only way people can make a profit on SQUARES. Whenever someone buys a product in the shop (a poster, t-shirt, physical cardboard pieces, etc.) with one of the artworks on SQUARES, the money left after covering expenses will be distributed between SQUARES and the user that created the underlying the square set. It doesn't make sense to give any contributions to the creator of the artwork, as it's easy on SQUARES to create artworks based on already completed artworks, so the artwork gallery will likely be floated with copies of high quality artworks.

The shop will not be implemented in this prototype, but the idea is to have a shop where its transparent what the square set creator will earn on the purchase. Furthermore, donations to the square set creator should also be possible through the shop by letting the buyer choose the price (a minimum price, a maximum price and a suggested price should be predefined for each product). This could also incentivize new designers to upload square sets as their earning potential is visible. This type of shop can for instance be found on <https://leanpub.com>.

Style Guide

The style guide presented here will cover the main design choices for the website. The overall theme will be to use squares in the design thereby supporting both the title of the website but also the business concept - combining squares to produce squared artworks.

Target group

Taking the user persona into account the user target group consists of people that want to signal to their surroundings that they are cool and up-to-date with the newest trends, but inside are struggling to find their own true individuality. They know how to handle a smart-phone but are still considered to be non-technical people.

Looking at the designer persona this part of the overall user group is quite different from the part described by the user persona and discussed above. They tend to be a bit more mature and business-minded, as they are not on a pursuit for finding their own individuality (they have found it), but are looking for something that can help them succeed with the career ambitions. They are discerning, as they easily can distinguish a well thought-out product from a poorly designed product. Additionally, they are considered professionals within their field of work.

To support these personas the design should be simple and intuitive so non-technical people can navigate the website and easily find square sets and produce artworks that they envision. Furthermore, the website should accommodate the picky designer that has a sense of what good design is.

Color Scheme

- A colour palette with examples of colours and an explanation of your choices.

Typeface

- Your choice of Font and an explanation of your choices.

Gestalt principles

- A Gestalt sketch and description of composition. Explain why you chose to organise your site in this particular way, knowing what you do about users and functionality.

Wrap up

- Explain how your design choices supports the sites content.

Technical Considerations

- Semantic HTML
- Possible use of JavaScript framework, i.e. AngularJS
- Browser compatibility
- Validation of HTML and CSS

The front-end part of the website is build using the JavaScript framework AngularJS together with the CSS framework Bootstrap, both helping speeding up the development process on the SQUARES prototype.

NodeJS is used as a simple back-end web server, where user information, square sets and artworks are stored. There is no real database behind the SQUARES prototype, as data is stored in back-end services, which are reset whenever the web server is restarted. However, having these back-end services simplified many of the application workflows as no data needs to be stored in the front-end - user information, square sets and artworks are just retrieved from the back-end using HTTP GET request whenever needed.

Besides a standard CSS reset stylesheet the styling is developed using Sass. The Sass files are compiled to CSS files using a Gulp task with a watch on all the Sass files recompiling when any changes are made to the Sass files.

Frameworks

As mentioned above the SQUARES prototype uses both a CSS framework and a JavaScript framework to simplify some of the development tasks thereby greatly shortening the development time. Bootstrap and AngularJS has been taught during the course, so these frameworks has been used developing the SQUARES prototype.

Bootstrap

Bootstrap is used to define the layout of the pages on the website using the Bootstrap 12-column grid system. Furthermore, Bootstrap is used in styling the navigation bar and the form elements and comes with a set of glyph-icons also used on the website.

The CSS framework is good for fast prototyping, as it makes some of the styling tasks faster to implement as the framework comes with a lot of pre-made CSS classes. However, going forward with the SQUARES prototype it should be considered to stop using Bootstrap, as it's a heavy framework (+400KB for the non-minified version used in this project) and it would make sense to style the website with custom-made CSS creating a unique look and feel.

AngularJS

Along with Bootstrap the website uses the JavaScript framework AngularJS. The SQUARES prototype uses AngularJS version 1.5.X introducing components and these are heavily used in the prototype. These components are special types of directives, which easily binds templates and controllers together forming encapsulated HTML element resulting in expressive mark-up. AngularJS components forms intuitive communication flows between components using clear component interfaces, one-way binding, callbacks to component events, lifecycle hooks, etc. Each page is then implemented using one or more components.

As sharing square sets and artworks through social media will likely be necessary for SQUARES to become succesful, the routing capabilities will be important having each part

of the website identified by a unique URL. This is done through the `ngComponentRouter` module, which makes it possible to route paths to relevant page components.

Libraries

The SQUARES prototype uses a few external libraries making some of the complex challenges more easy to solve and in far less time.

Canvas

The canvas page is developed using the library `InteractJS`, which comes with many needed features out-of-the-box including drag and drop, event callbacks and grid-snapping. Furthermore, the library has excellent documentation.

Alternatives to `InteractJS` could be using `AngularJS` directives simulating drag and drop functionality by hooking into mouse-down, mouse-up and mouse-over events, but still rather limited compared to `InteractJS`.

Another alternative is called `codeFormer`, which makes jQuery-like drag and drop easy in `AngularJS`. Still, the lack of key features such as grid-snapping is the reason `InteractJS` is used in the SQUARES prototype.

State compression

As previous stated, sharing through social media can become quite important for SQUARES going forward. To support this even further a prototype for a canvas state-generator has been included in the SQUARES website. This state-generator makes it possible to share a canvas without creating a user on SQUARES and save the progress as an artwork. A unique URL can instead be generated, which contains the current state of the canvas - much like the functionality available on <http://paletton.com/> - and can therefore be shared with everyone.

This is done by having the canvas state represented as a JSON object, which then can be serialized (using the `JSON.stringify` function) and then encoded to greatly reduce the length of the stringified JSON object. The encoding could be accomplished using the `btoa` base64 encoding function (`atob` is the decoding function), which is part of the HTML5 specification. However, a library called `lz-string` is used instead as it's both a lot faster than the `btoa/atob` functions and the length of the encoded string is also greatly reduced compared to the base64 encoding.

When the generated URL is entered in the browser the encoded state is part of the query string. This encoded state string is then decoded using `lz-string` and then parsed into a JSON object using `JSON.parse`, which is loaded into the canvas component.

The implementation in SQUARES is still sub-optimal as the generated URL is quite long, but there are plenty of opportunities to compress the canvas state even further thereby reducing the URL length.

This feature has a useful side effect as the canvas state can be stored along with each artwork, whenever an artwork is saved by a user on SQUARES. When browsing through the art

gallery users can then pick whatever artwork they like and load the stored state directly into the canvas starting where the creator of the art saved his work.

HTML to image

Square set upload

Future improvements

- improved mobile experience using square set slider on canvas page