Origami Website Implementation Report

DECO1400

Iain Jensen 47485291

2022

# Implementation Summary

The created website consists of 10 pages, consisting of a:

* Home,
* Products, Tissue Foil Product, Elephant Hide Product and, Kraft Paper Product,
* Bases selection, Simple Folds base, Dog Folding Demonstration,
* Contact Us, and
* ‘Error 404 Page not found’ page

Within these pages, a number of CSS and JS features were implemented, such as

* Height changing, and sticky header
* Navigation dropdown arrows, and information and class toggling
* CSS @media and variable use for changing font, and other sizes
* CSS enabled re-sizing map
* Application of 3D CSS transitions to SVG elements
* JS enabled SVG origami demonstration
* Custom Carousel

3 complex uses of CSS and JavaScript will be highlighted, namely the ‘simple folds’ transitions, screen size adaptability, and an interactive JavaScript dog folding demonstration.

## Simple Folds Transitions

The simple folds page, bases-simple-folds.html, is primarily comprised of two interactive CSS based SVG elements demonstrating the process of two types of folds. The image is demonstration is modelled by two halves of a ‘piece of paper’, with two CSS transitions being performed on one half of the paper to simulate the piece of paper folding in half. These transitions are shown below:

A picture containing text

Description automatically generated

Figure 1 - Code to perform fold transition (transitions identified by their relevant fold, rotation and transition number)

A picture containing diagram

Description automatically generated

Figure 2 - Fold first transition

A picture containing text, businesscard, screenshot

Description automatically generated

Figure 3 - Fold second transition

The use of this CSS is complex because a 3D transition, applied to bespoke locations on the rectangle element is used to create a 3D effect of the paper moving. The implementation of this component was challenging, especially creating the 3D effect and balancing how realistic it was with what could practically be achieved without too much time spent debugging. This could be an element considered for improvement, and considerations of implementation using a polygon, configuring the sides using JavaScript instead of rotating the half page, or using the perspective options, may be alternate, and more visually appealing methods of implementation.

## Screen size adaptability

CSS Flexbox is used frequently and widely throughout the website to improve the user experience of users with varying screen sizes. One such example can be demonstrated in the website homepage, index.html. On a normal laptop screen size, all font on the page is sized to a viewable, and unobtrusive level.

Graphical user interface, website

Description automatically generated

Figure 4 - Home Page, index.html, at laptop width (1024px)

Items are spaced out evenly, but not excessively long, and no boxes overflow. Should the user view the same page on a smaller device, the webpage is responsive as shown below.

Graphical user interface, website

Description automatically generated

Figure 5 - Home Page, index.html, at tablet width (768px)

The nav menu and header shrink to accommodate more room in the main viewing area. The font size decreases, and items scale accordingly. For even smaller viewing areas, such as mobile phones, the Flexbox display allows for re-positioning of items as demonstrated.

Graphical user interface, application

Description automatically generated

Figure 6 - Home Page, index.html, at mobile width (425px)

Here, the image in the carousel enlarges relative to the screen size, and the text fits underneath, allowing for an easier viewing experience for the user.

In a similar regard, if a user was to use a larger device above a 1400px width, such as a monitor, the font size across the page increases to fill the space better.

This feature can be considered complex because of the overall responsiveness of the page and ability to cater to different users. I did not have too much difficulty implementing these features, as the main method that this was achieved was through @media tags, however the debugging and refinement on the boundaries for when to resize was slightly challenging.

## JavaScript and SVG Origami Dog folding demonstration

The page bases-dog.html is primarily comprised of an inlaid SVG JavaScript applet demonstrating the process of how to fold an origami dog. The SVG is comprised of different elements representing different parts of the paper in the model. Users are asked to click through the steps by clicking on the fold line. The fold line is made up of the visible line itself, and another rectangle that acts as the clickable area. When the area is clicked, the next elements are displayed, and previous elements hidden.

A picture containing shape

Description automatically generated

Figure 7 - Origami dog second last slide, showing back button and clickable fold line

Shape

Description automatically generated with medium confidence

Figure 8 - Origami dog final slide, notifying user of completion, and presenting option of restart or back

For ease of navigation, the applet also has back and restart buttons to cycle to and restart earlier steps as shown above. In addition to the above features, the SVG canvas, and all elements contained within it, including the font size, scale directly to the viewing width of the device. This feature greatly enhances a user’s ability to view the demonstration and ensures that the applet does not grow too big or too small.

The complexity of this section is present in this applet’s ability to respond to users’ interactions. It has multiple methods of interaction and is more interactive than a simple carousel. The placement of the items on the image were quite difficult, however meaningful identifiers contained within the JavaScript code, and commenting of individual parts made this process easier.

# Nielsen’s Usability Heuristics

## #1: Visibility of system status

Features such as changing colours in the carousel direction arrows and changing the cursor over certain features provides user feedback that the mouse location is being registered. On the product pages, the user is also alerted that their input is required on the left, providing the user with direct input of the requirements of the ‘in stock’ function. Consideration could be given to display a modal when the contact us form is submitted.

## # 2: Match between system and the real world

The system uses simple language and a clear font that is easy to understand. Where jargon is used in the folding and product names, a demonstration in the SVG transition, and product descriptions are given to show the action.

## #3: User control and freedom

The clearly marked navigation bar and sticky header, accessible over the whole page, allows for quick navigation to any page on the site map. The back and restart buttons in the dog demonstration provide flexibility for the user in that particular module as well.

## #4: Consistency and standards

The website uses a range of standard icons and symbols including drop down arrows for menus and carousel arrows to indicate ‘next’ and ‘back’ options. Further, the website follows the conventions of consistently providing a title header linking to the home page at the top of all web pages, and a footer at the bottom with the contact details of the website, providing quickly accessible information should the user require it.

## #5: Error prevention

The main location for mistakes would probably be a mis-click on a navigation bar hyperlink, however by having it consistent across all webpages, the requirement to go back to a previous webpage is not necessary. Otherwise, there is low user error possible across the website.

## #6: Recognition rather than recall

Users are presented with the full site map in the nav bar, directly providing them options with which pages they can access. This is better than a search function for example, which would require users to recall the site they wish to visit.

## #7: Flexibility and efficiency of use

The website is simple to operate with no lengthy scrolling areas or features that are inefficient to access and hence there was not a need to include efficiency gestures.

## #8: Aesthetic and minimalist design

The webpage has a relaxing, harmonious colour palette, and the design is highly minimalist, with clear border definitions and little clutter. The font and font size are all clear and readable.

## #9: Help users recognize, diagnose, and recover from errors

As discussed earlier, there is low risk of error in using this webpage. Webpage Not Found page has been included if the website was implemented properly, providing user feedback on the error, and showing the pages available to them.

## #10: Help and documentation

Prompts are present in the dog and simple fold modules however, help to use this website has been identified as not require. Brief commenting, meaningful identifiers and white space have been used to increase usability of the website source code by other developers.

# Conclusion

The website created was successful in providing the main functionality of the origami website as designed. Although lacking some aspects, such as additional origami bases, the functionality of these missing aspects are all demonstrated through other pages. I am happy with the website as a whole, and especially proud of the origami dog demonstration and simple fold transitions.

Through completing this assignment I particularly learnt about, and tried to practice the concept of clean CSS. In a lot of past projects I have had a bad practice of debugging by putting in random CSS code, and when something works, leaving it. In this project I especially tried to clean up unnecessary CSS and ensure that if this source code was to be used in the future, styling could be applied to elements with a confidence that unintended changes to external elements was minimal.

Another thing I learnt is the versatility of

# Future Work

This site could still be developed further, and a number of features ideated in the design were not included because of time constraints. Features like product filters, utilising Flexbox order and JavaScript sorting, or additional CSS transition base pages could be implemented.

# References

coolors.co. (n.d.). *Color Palette Generator*. Retrieved from coolors.co: https://coolors.co/6f4e37-e59f71-ffffff-6cae75-90bede

fontawesome. (n.d.). *caret-down*. Retrieved from fontawesome: https://fontawesome.com/icons/caret-down?s=solid

Kanillar, E. (2021, March 29). *How to Make an Image Carousel Using Basic HTML, CSS, and JavaScript*. Retrieved from Section.io: https://www.section.io/engineering-education/how-to-make-an-image-carousel-for-your-website/

*SVG element reference*. (n.d.). Retrieved from mdn web docs: https://developer.mozilla.org/en-US/docs/Web/SVG/Element

Terry, N. (n.d.). *Origami-shop*. Retrieved from Origami-shop: https://www.origami-shop.com/

w3schools. (n.d.). *MouseEvent offsetX Property*. Retrieved from w3schools: https://www.w3schools.com/jsref/event\_offsetx.asp

w3schools. (n.d.). *SVG*. Retrieved from w3schools: https://www.w3schools.com/html/html5\_svg.asp

Young, D. (n.d.). *Detecting Page Scroll Distance*. Retrieved from David Young: https://davidyoung.tech/detecting-page-scroll-distance