Hostyle UI: Report

IS53028A: Project in Creative Computing

Sahil Miah (Smiah008)

Table of contents

List of Figures

List of Abbreviations

Abbreviation	
UI	User Interface
HCI	Human computer interaction
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
SCSS	Sassy CSS
GSAP	GreenSock Animation Platform
DOM	Document Object Model

1.0 Project overview

(approx. 500 words)

A succinct description of your project at the start of your report. Make sure your project has a title. Try to answer the questions 'What is it?' 'Why is it?' and 'How is it?' before going on to describe the project in more detail.

I will be exploring the interaction between users and an interface that is made with intentionally bad (hostile and 'disfluent') design. This project is called Hostyle UI, and it involves making a web app that explains and highlights the important aspects of HCI and design principles in a hands-on method. The different pages will cover a certain aspect of HCI that incorporates both written descriptions and live examples of the downfalls of said aspects. Users will navigate through the page and experience the frustration of the design. I'd like users to be able to explore and compare these design principles by showing how the UI would look with both good and 'bad' design. I am hopeful that this would help them further understand and consolidate the information on the different principles that are explored on the site.

2.0 **Background research**

(approx. 1500 words)

Place your project in context by making reference to existing materials. Depending on your project, these could be software, products, research papers, artworks, artists, specific techniques, or blogs. Make sure to explain the significance of each to your chosen project. Make sure you reference these using Harvard style referencing.

In my research I found an interesting blog at Arstechnica by Samuel Axon about "the most (intentionally) poorly designed website ever created" (Axon, 2019). It had a fun take on poorly designed web pages, mainly terrible form inputs and slightly drastic design choices. The page is cleverly called User Inyerface (bagaar, 2017) and includes a very frustrating form to fill in that is meant to lead to a game. Painfully small buttons, confusing instructions and animations that throw you off completely; this playful site is a very big inspiration to what I want the experience from my work to be like.

I also found another fascinating blog post on Elegant Themes which pointed out that the "design of the site itself is funny because of how terrible it is" (Barron, 2015). The examples shown in the blog were indeed humorous and I could see some aspects in the sites that would even make the experience of my web page both fun and a good learning point.

3.0 **Context, users/audience, goals**

(approx. 1000 words)

Describe the context of your work (i.e. where does it take place, or where will it be used?)

Relate the context to its participants or audience by describing whom your project is aimed at:

- For projects that are primarily experienced by someone (for example, an artwork or performance), describe the intended effect on your audience. You should compare that to some of the key inspirations from your background research section.
- For projects that are primarily intended to be used by someone or something (for example a tool, game, product, app), describe the types of uses and the (testable) goals for its usage

4.0 Project in depth

A detailed description of the final output (approx. 1500 words)

4.1 Key features

- Start with a precise listing of all the key features of your project. This is a simple description of what your project does, and how it reacts to input. Features might provide a benefit to the users (for projects that are used) or contribute to the experience (for projects that are experienced). A description of features might include:
    - A particular set of sounds that you designed
    - A detailed description of how a game controller works, and what each element does
    - A description of the key visual components of a projection-mapped artwork
    - The rules and goals of a game you designed

4.2 Supporting materials

List the supporting materials that your project uses, including languages, libraries, frameworks, hardware and software. Explain what features of these that you used and why.

The main programming languages will be HTML, CSS and JS as it will be a web page. Given that the site might eventually get large, I decided to use SCSS for an easier implementation as you can use "loops, functions, imports, variables, and mathematical operations, thus making CSS writing more powerful" (Aulak, 2018). I will not require any physical materials as this will simply be running on web browsers; I will make it so this is compatible with all/most web browsers. Furthermore, in the prototyping stage of the project I will use Adobe illustrator and XD to design and build mock-ups.

4.3 Techniques

Highlight any interesting techniques that you used and describe how they work.

CSS Grids?

GSAP animations?

I have switched browsers from Chrome to Firefox as it is deemed better for development. The main feature I started using that isn't present on Chrome was the accessibility tools, namely the contrast checker. This could be used for making sure the colour choices are clear and readable, on the other hand, it can also be used to make intentionally poor colour choices for the live examples.

4.4 Structure of Code

- Describe the structure of your code and demonstrate how data flows through it. Supplement written descriptions with flow charts, process and class diagrams to make things clearer for the reader. For example:
    - a diagram to explain how a user interacts with a web quiz
    - a diagram showing how messages originate in the front-end UI and pass to a web server and database.
    - a diagram showing how data passes from the CPU to the GPU in your program

5.0 Creative process and user/audience evaluation

(approx. 1500 words)

Describe your creative process. Demonstrate how you used iterative design in your process:

- For projects that are experienced, discuss how they changed each iteration, and provide evidence from tests: by yourself, with others, or critiques and discussions with experts.
- For projects that are used, discussed how your user centred design process was carried out with key stages of the process (e.g. following the Double Diamond) and evidence of user testing, logical thinking and discussions with experts.

Notes-

Wireframes to Paper prototype

Write about getting feedback in between each step

Formally write up notes from discussions,

all suggestions, changes and things they liked

Feedback and changes gathered from testing Adobe XD prototype.

After a number of discussions with various people who have seen and used the XD prototype, I have gathered helpful insights that I believe would improve the project.

Firstly, this testing made me realise I wanted to create a guided navigation through the UI as it will be something the user experiences, taking this idea into account means I am willing to discard the navigation bar and have the user click through. Finding their way around the UI in the way that entices them to follow the trail, rather than a stale nav-bar that offers up everything with no thoughtful co-ordination.

Another thought was to have the toggle feature that shows both designs for specific parts of the elements, breaking them down into smaller digestible changes rather than changing the entire page. Users would then be able to focus on different aspects at their own speed and take in more information overall.

Another point that was brought up was to have the hostile elements match the theme/topic of the page instead of having randomly placed hostile elements. I believe having a more focused arrangement of these hostile elements would reduce the amount of unnecessary distraction while keeping the interactions more focused and beneficial in delivering the information intended by the specific page.

6.0 Debugging and problem solving

(approx. 1000 words)

Use this section to highlight technical challenges that you faced in your build. Describe the problem carefully and give a blow by blow account of the stages you went through to solve them. Describe the dead ends and rejected approaches as well as the final approach. You might reference tutorials that you followed or Stack Overflow posts which you tried. If a classmate or tutor helped you, you can reference them here too.

Slow load times and optimising Images

For a smoother experience and quicker load times, I have to remember to optimise all the images used in the site. I currently have a logo image that is far too large and high resolution that it affects the load times, I intend to reduce the size as well as put in measures to keep the image size small depending on the screen size so only the resolution that is required is loaded.

A long time was taken to rearrange the layout of the first page, so I looked into different CSS Grid techniques. I've come to learn about CSS grids in greater detail and found a very useful and easy method to set up appealing layouts as well as being able to adjust them with significant ease. The different pages may have different layouts depending on the content, switching layouts by simply altering `grid-template-areas`; I can explore an endless number of layouts to see which one is best.

```
.home{
    height: 100vh;
    display: grid;
    grid-template-columns: .3fr 1fr 1fr 1fr .3fr;
    grid-template-rows: 1fr 1fr 0.5fr 1fr;
    grid-template-areas:". . . . ." ". b2 b1 b1 ." ". . . . ." ". b3 b3 b3 .";
}
.b1{
    grid-area: b1;
    align-self: center !important;
    text-align: center;
    @include border-radius(10px);
}
```

FIGURE 6.1

I spent a considerable amount of time trying to figure out how to implement the toggle functionality, specifically the taking away the clickability of buttons. I searched for ways to efficiently change the DOM elements themselves to removing the element completely, but the was no success until I came across the CSS property pointer-events and with one line it solved the issue of removing clickability.

I realised that most changes would be done through CSS anyway, so I created a small and simple CSS class and used JavaScript to toggle the Classnames of the elements, adding and removing the class name that brings the clickability. This exact technique could be replicated for other changes in the future, for example having two CSS classes that style the elements as good or hostile and toggling between the two with JS.

```css
.inactive{
    pointer-events: none
}
```

FIGURE 6.2

```html
<a id="nextPage" href="colour.html" alt="Colour in design">Here </a>
<br><br><br>
<button class="btn-toggle" onclick="toggle('nextPage')">Toggle</button>
```

FIGURE 6.3

```javascript
function toggle(id){
    var np = document.getElementById(id);
    np.classList.toggle("inactive");
}
```

FIGURE 6.4

7.0 Self-evaluation and conclusions

(approx. 1000 words)

You should address:

- How successful was your project according to your goals.
- How did your goals change over the duration of the project. Was your project appropriately scoped in the first place?
- How have you tested your project? Justify why those tests are appropriate.
- Which aspects of your creative process worked well and which could be improved.
- How successful was your approach to problem solving. How could it be improved?
- How could you see your project developing in the future?

Evidence your commentary by making reference to your initial research, iterative stages, final outputs and final tests.

References

Aulak, A. (2018, 10 09). *Benefits of SCSS in Wordpress*. Retrieved 11 03, 2020, from Grazitti: https://grazitti.com/blog/benefits-of-scss-in-wordpress

Axon, S. (2019, July 4). *Behold, the most (intentionally) poorly designed website ever created*. Retrieved December 2020, from Ars Technica: https://arstechnica.com/gadgets/2019/07/behold-the-most-intentionally-poorly-designed-website-ever-created/

bagaar. (2017). *User Inyerface*. Retrieved October 2020, from https://userinyerface.com

Barron, B. (2015, April 25). *Bad Web Design: A Look At The Most Hilariously Terrible Websites From Around The Web*. Retrieved December 2020, from Elegant Themes: https://www.elegantthemes.com/blog/resources/bad-web-design-a-look-at-the-most-hilariously-terrible-websites-from-around-the-web

Nikolov, A. (2017, April 8). *Design Principle: Consistency*. (Medium, Producer) Retrieved January 2021, from UX Collective: uxdesign.cc/design-principle-consistency-6b0cf7e7339f

Shneiderman, B. (2016, May). *The Eight Golden Rules of Interface Design*. Retrieved 11 04, 2020, from UNIVERSITY OF MARYLAND: https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant , C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the User Interface: Strategies for Effective Human-Computer Interaction: Sixth Edition* (6 ed.). Pearson. Retrieved from https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces

Appendix

	Description	URL
(A)	GitLab Repository	https://gitlab.doc.gold.ac.uk/smiah008/pcc
(Ai)	GitLab Dev Folder	https://gitlab.doc.gold.ac.uk/smiah008/pcc/public_html/
(B)	Adobe XD Prototype	https://xd.adobe.com/view/92380219-f418-4622-a98f-2e36122fdb4f-c76b/
(C)	Project Blog (Tumblr)	https://smiah008.tumblr.com/