CS130(0) Style Guide

GENERAL GUIDELINES

Every assignment you hand in for this class should be written with a reader that is unfamiliar with the assignment and the course in mind. In order to get full style points, your assignments should feel like engaging portfolio-ready pieces that someone outside of this class can easily understand - this means the reader should be able to understand the premise of the project without reading the assignment handout. Additionally, the material should be interesting and engaging enough that a viewer would go out of their way to read it. A general rule of thumb is to consider the assignment as something you would post on your actual portfolio or submit as an example of your work to a job application. This content style guide offers some tips on how to achieve that.

Why is style important?

In industry, you'll often be tasked with communicating the value of your work. You might get questions from developers on how to implement your designs, or a product manager might not think your idea is worth executing. It's important to present your work nicely to make sure the high-quality content and hard effort you put in it is not lost in communication.

We want you to get practice presenting your work outside the context of the assignment and as are engaging works on their own. In industry, you'll also have to be able to present your projects as stand-alone pieces where a reviewer without context of the assignment you had would be able to understand your thought process and design choices.

How can I make my piece portfolio-ready?

The main thing is to make sure that someone outside of the class who has no knowledge of the handout can easily follow your write-up by providing context for your writing. This means having a paragraph at the beginning of your handin that explains the problem and sets context for the reader so they can understand what the purpose of your handin is without having read the assignment. Additionally, have transitions between the different sections of your handin; explain how you're moving from one section to another and why they're related.

Your pieces should also be engaging to the reader and something someone would read if they were just scrolling through Medium or another posting site. To achieve this, think about articles you read on Medium or other platforms, why you read them and why they're engaging. Be sure to also explain your reasoning in your handin and elaborate on your ideas so it makes sense to a reader who is not familiar with the assignment. Again, refer back to other articles you read by professional designers or UI/UX researches.

Additionally, below are some tips for good presentation and making sure that your work stands alone as a portfolio-ready piece. Please note that following all of these does not necessarily guarantee a perfect score.

Some Useful Tips:

For writing:

- Make sure the reader would be able to understand the assignment if they didn't read the handout of the assignment beforehand. This means don't reference the name of the assignment in any titles or text within the assignment and be sure to explain the premise of the project.
- Introduce the problem statement at the beginning of the handin. Explain what you're doing and why. Think of this like the introduction to an essay - you wouldn't write an essay without an intro!
- Have transitions between different sections to help the reader follow along.
- Proofread! Use spell-check.
- Be concise, and don't repeat information. Underline or italicize to indicate importance.
- Adhere to the 2-page limit. This applies to text only -- labeled figures may be added to the end of your hand-in. It is okay to add text in your visuals in moderation.
- Break long paragraphs into smaller chunks of text.
- Using big words might not be helpful. Aim not to impress your reader with your vocabulary, but with the quality of your work. If simpler words make your work stand out more, use those.
- Example:
  - "This user was discombobulated, riddled, and puzzled by this interface." → "This user was confused by the back button's placement."

For images:

- Don't use blurry images.
- For paper sketches, make high quality scans instead of taking photos by phone. Make sure your sketches are organized and clear, lines are straight (use a ruler), and everything is labeled to make it easier to understand the sketch.
- For digital illustrations (i.e. most wireframes and prototypes), save them as vector-based formats like svg, ai, or eps whenever possible. It's generally not advised to save your illustrations as png or jpg before embedding in your pdf.
- For normal photographs or native raster images (e.g. a profile photo of a persona, image of a waterfall), formats like png and jpg images are fine. Jpgs are lossy (the quality degrades when edited) and should be used only for photographs.
- Keep text separate from your images as much as possible. If you must include text inside an image, make the font size large enough to read.
- Use colors helpfully, but not gratuitously.
- If images are referenced in the text, label them helpfully with names like "Figure 1".

For code:

- Your code should be written in a way that allows another developer to quickly understand what's going on and add new features if necessary even if they haven't read the handout for the assignment.
- Any code taken from online or elsewhere must be clearly cited in either a comment or a README or you will be in violation of the collaboration policy.
- Use descriptive names for variables and methods.
- Avoid repeating code.
- Comment helpfully!