

# PSAM 1028

## CD Foundations: Interaction

<b>Program</b>	School of Art, Media, and Technology: Communication Design
<b>CRN</b>	<b>2102</b>
<b>Semester</b>	<b>Fall 2025</b>
<b>Meeting Day</b>	<b>Monday</b>
<b>Meeting Time</b>	<b>12:10pm - 2:50pm</b>
<b>Building/Room/ Zoom</b>	<b>Parsons 2 W 13th — Room 602</b>
<b>Instructor &amp; Email</b>	<b>Andrés Cuervo <a href="mailto:cuervora@newschool.edu">cuervora@newschool.edu</a></b>
<b>Class Website</b>	<b><a href="https://cwervo.github.io/cdff25/">https://cwervo.github.io/cdff25/</a></b>

## Course Description

CD Foundations: Interaction is designed to introduce students to programming as a creative medium—as a way of making and exploring. The coursework focuses on developing a vocabulary of interaction design principles which can then be applied across a range of platforms. Students are encouraged to experiment with various media, tools, and techniques, ultimately producing a portfolio of interactive and visual projects designed for the screen. An emphasis is placed on typography as it applies to a screen context, research-based problem solving and a learning-through-making approach to technical skill building. Historical and current interaction design precedents will be discussed. This course is intended for non-communication design majors, as an introduction to the discipline.

## Readings

1. Casey Reas, Chandler McWilliams, and LUST, *Form+Code in Design, Art, and Architecture*
2. Kimberly Elam, *Geometry of Design*
3. Armin Hofmann, *Graphic Design Manual*
4. Robert Bringhurst, *The Elements of Typographic Style*
5. Frank Chimero, *The Shape of Design*
6. Leah Buley, *The User Experience Team of One*
7. Compiled by Laurel Schwulst, *Very Interactive Library*
8. Paul Ford, *What is Code?*

## Course Outline

### **Unit 1 Week 1-4: Working methods**

The first segment of Core Interaction will focus on the tools and concepts required for building interactive experiences. We'll use the languages of the web because they're accessible and immediately open up new modes of communication for designers, but the concepts will be transferable to any screen-based or interactive media.

In weeks 1-4 we will focus on:

- File management (naming, organization, file paths)
- Setting up and starting a new project
- Tools (code editor, inspector, git/github)
- HTML/CSS basic concepts and syntax
- Figma (components, prototyping, grids, canvas sizing)

Sample projects: *Interview, Expressive Text, All HTML*

### **Unit 2 Week 5-8: Digital canvas**

In our second segment, we'll investigate how designing for the digital canvas differs from other media. We will aim to understand the inherent complexities and how to use them to create compelling digital experiences.

In weeks 4-8 we will focus on:

- Typography with HTML/CSS
- CSS selectors (cascades, combining, parent/child, pseudo)
- HTML structure (box model, dissecting a web page)
- Layouting (position, float, flexbox, grid)
- Designing for the digital canvas (how big is a browser?)

Sample projects: *Flags, 25 Variations*

### **Unit 3 Week 9-11: Designing for interaction**

Thinking about a website as a series of linked pages, we'll take the concepts we used to make individual web pages and apply them to larger systems. We'll explore how our systems can be designed to flex, rather than break, under a wide range of variables while still maintaining the original intent of the design.

In weeks 6-9 we will focus on:

- Multi-page systems
- Programming basic user interactions (:hover, basic JS click, etc.)
- Time-based design (interactive states, storyboarding, prototyping)
- User models (entering and receiving data, user flows, UX patterns, ways of navigating)

Sample project: *Stories as Networks*

### **Unit 4 Week 12-15: Networks**

Because a website lives in a larger network of apps, websites, devices, and contexts, our final segment will explore how our website lives online. We'll take the work we've done this semester and explore self-publishing and making our work public by putting our work on the internet.

In weeks 10-15 we will focus on:

- Putting a website online (hosting, Github, custom domains)
- Accessibility
- Asset creation (video, image optimization, WebGL)
- Metadata (search, social)
- Connecting to other web services

Sample project: *Living Collection*

## Learning Outcomes

*By the end of the semester, students will be able to:*

1. Use a basic vocabulary of interactive media to both give and respond to critique productively.
2. Create compelling interactive experiences through more careful and inspired interpretation/translation of content (i.e. develop great design concepts)
3. Demonstrate an understanding of the iterative making process in interaction design, using incremental methods such as prototyping, user research and evaluation to build toward more advanced work.
4. Conceptualize a product, object, or experience for the web and realize it through coding.
5. Evaluate the difference in designing interfaces for different kinds of devices, their limitations and specific user situations including responsive websites and apps for mobile.
6. Evaluate how typography and its variables are applied to interactive systems to facilitate orientation, support usability and create consistency.
7. Research historic and current design precedents to contextualize your own work.
8. Be able to archive and document work that is printed, on screen or time based in a reflective manner for learning portfolio.
9. Combine your artistic creativity with technology related to the internet.
10. Demonstrate a comprehension of skills, methods, techniques and processes to realize interactive systems, particularly systems for dealing with unpredictable, variable, and ever-changing content.

## Assessment Criteria

15% Attendance & Class Participation  
15% Unit 1 Projects: Interview, Expressive Text, All HTML  
15% Unit 2 Projects: Flags, 25 Variations  
25% Unit 3 Project: Stories as Networks  
30% Unit 4 Project: Living Collection

## Attendance, Grading and Work Submission Standards, Program Policies, Making Resources, and University Policies

All CD classes adhere to the same program and university policies:

[https://docs.google.com/document/d/1u358io8doX\\_SVVMGqIM\\_oH5V00IccneYu4Ww-uE55QM/edit?us](https://docs.google.com/document/d/1u358io8doX_SVVMGqIM_oH5V00IccneYu4Ww-uE55QM/edit?us)

[p=sharing](#)

## Sample Projects

### Interview

Partner up with one of your classmates. Interview your partner. Create a series of linked web pages that introduces your assigned partner to the class. Your project should consist of at least 10 individual pages, and should reveal at least 10 facts about your partner, other than their name.

### All HTML

Create a series of 5 unique web pages which focus on one or two HTML tags each (no CSS whatsoever, even inline.) The idea of this assignment is to see how html tags look by default, and to use (or misuse) them in an inventive way.

### Expressive Text

Find or compose a piece of writing that can be experienced on a browser. The length of the text should be approximately one printed page. The text can be a poem, manifesto, quote, etc. Layout the text so the experience of reading changes based on scrolling through it.

### Flags

Vexillology is the study of flags. All flags have a shared visual vocabulary (stars, stripes, crests, crosses) and there are specific terms to describe these things. Flags are always of interest to designers because they communicate a lot of meaning with very simple, graphic forms. They're also a perfect thing to make with code. Most of the lines are straight and most of the shapes are described with simple geometry. Start by doing a little research on flags.

You will be given a set of flags. You will be using HTML/CSS to recreate these flags with code. Research your assigned flags and do some planning as to how they can be created. What size should they be? What elements cannot be created using code (and how will you make them)?

You will display all of your flags on one page. What code can be reused between your flags? Give special consideration to how you will present your flags on a single page (and what additional code needs to be written to accommodate that).

Rules: You may only use images for an Emblem or Charge.

### 25 Variations

Choose a published poem, no more than about a printed page. You will be making 25 formal variations using this poem. Each variation will be a separate webpage. For each variation, you will focus on changing the poem's form through shifts in composition, typography, hierarchy, scale, and pacing. The poem's content must remain intact throughout all 20 variations. This project is divided into 5 levels of increasingly looser constraints of 5 layouts each.

### Stories as Networks

Jorge Luis Borges was an Argentinian writer famous for his short stories that deal with labyrinths, dreams, religion, and mathematical ideas (particularly set theory concepts like infinity and cardinality). His

circuitous and meandering prose, full of allusions and vivid imagery, is a good way to think about the web as a network that has many nodes and many connections that continuously folds upon itself. It is the act of navigating through this maze that brings meaning to the web experience.

Read all three short stories from *The Maker* (1960): “Parable of the Palace”, “Everything and Nothing”, “His End and His Beginning”.

Choose one story and set the text using HTML and CSS so that the reader will not only be able to read the story but also experience your interpretation of the story. You may use one page or multiple pages to convey this experience. This experience will be viewed on both large and small (mobile) screens. You should use no more than two typefaces. Representational images are not allowed.

- How does hyperlinking (internal and external) enhance the meaning of the story? - How does your interpretation of the story shape the way the information is presented? - Does the experience change when viewing the page on a larger or smaller surface area? If so, how?
- Linearity versus non-linearity
- Pay attention to typographic details: special characters, leading, words per line, etc.

### **Living Collection**

With the use of the collection amassed in the previous part of the class, you’ll be designing and building a website to contain and display that collection.

Keep in mind that you’re not designing a fixed, unchanging website: you’re designing a system that can expand or contract to show this collection as it changes. The design of your site should also in some way reflect your design perspective, particularly what you’ve discovered about your interests and working methods during this course.

Also, think about how your content—and design concepts related to this content—will help you to give form to your site. Other questions to consider: Does the design of the site somehow respond to new content? Rather than being a neutral vessel, how can the design that you use to organize your collection change when the collection itself changes? For example, do colors on the site change in response to the kind or amount of content posted to the site? Does the arrangement of elements or the grid change? Does the site’s navigation change to highlight the most current content?

### **Minimum Requirements**

- Content Addition & Protocol Area(s): The site must contain a
- “mobile-first” area that allows the user to understand the protocol used for the creation of the collection and to allow them to add to this collection, as well.
- Collection Display Area(s): The site must contain an area (designed for larger screens) that allows the user to view your collection.

The process is as follows:

- Create a hierarchy: What are the specific kinds of information that the site will contain/display? Which information should be more prominent? Less prominent? How can you give order/structure to the information?
- Create a User Flow: Map all of the screens/states of your site that shows what information will be displayed, and how the user would navigate from one screen to another. You shouldn’t be thinking of visual design at this point, just of how your site is organized and the way the user will

move through it.

- Wireframes / layouts: Create wireframes that show the structure of your site, and layouts that show how each screen will look. Show a full range of examples of the kind of content your site will contain, interactive effects like hover states, and how the site will change in response to different content. How can you take advantage of differences in screen sizes?
- Functional prototype: Using your layouts as your guide, build your site, test it, adjust the design and the code, and continue tweaking. You'll undoubtedly encounter problems with the design that you didn't initially anticipate, which is why multiple iterations are crucial. Use the qualities and opportunities inherent in the web to your advantage: interactive cues, generation of page code using JS (i.e. not only static HTML/CSS), links to external sites, video and audio, and soliciting user input all can be used to create a powerful experience. Be sure to check your design decisions against your content and your intentions/goals to avoid going overboard. Carefully consider the site's form as it relates to the content.

## Exercises

### Analog Programming

In this assignment, we will try to answer the question "what is a program?" at its most fundamental level. Part 1:

1. Create a design for a concert poster that is interactive and also that you find visually compelling. It must include at least one original image and any information that you deem necessary.
2. Based on the questions like those posed above (but probably not limited to), in everyday language (and words, only), write a 'program' that allows someone to replicate your design. Pay attention to the text as well as the form of this program.
3. Print out your instructions.

Part 2

1. Trade programs with a partner without showing them your design.
2. Using the program you've been given, create a new poster design
3. Present your designs side-by-side with your partner. Show your programs and explain the logic behind them. Where did your program run without 'errors'? Where did your program create an unexpected result? Consider how you will present all of this information.

Discussion

- How does a program enforce standards (rules) vs. facilitate expression (variables)?
- How will your partner get the assets required to complete the design (fonts, images)?
- What assumptions do we need to make about tools? Workspace? Software settings?
- How do you solve problems when the original designer isn't available?

### Git Poetry

In this assignment, we will write a poem together in class. As a demonstration, I will write the first line of our poem, and using Git, commit and push it to a Github repo that we all have access to. We'll go around the room, with each consecutive person pulling the latest version of the code, adding a new line to the poem, and commit and pushing it to the repo so that the next person can continue our poem.

### Abstracted Web

Reference layouts of a chosen category of website(social media, news, streaming, etc) and recreate the basic layout of five sites using CSS and HTML. Tie these bootlegs together with a home page explaining your process and concept.

## Materials and Supplies

Laptop

Camera

Software: Git/GitHub, VS Code, Figma

## Sample Schedule

Week 1	Working Methods
Asynchronous	Lecture: Computers, files, and networks
Synchronous	
Homework	Interview
Supplemental Resources	

Week 2	Working Methods
Asynchronous	Lecture: Tools, file management, version control
Synchronous	Git Poetry
Homework	Interview
Supplemental Resources	

Week 3	Working Methods
Asynchronous	Lecture: HTML/CSS basic concepts and syntax
Synchronous	Demo
Homework	All HTML
Supplemental Resources	

Week 4	Working Methods
Asynchronous	Lecture: HTML structure (box model, dissecting a web page)

Synchronous	Demo
Homework	Expressive Text
Supplemental Resources	

<b>Week 5</b>	<b>Digital Canvas</b>
Asynchronous	Lecture: CSS selectors (cascades, combining, parent/child, pseudo)
Synchronous	Demo

Homework	Flags
Supplemental Resources	

<b>Week 6</b>	<b>Digital Canvas</b>
Asynchronous	Lecture: Layouting (position, float, flexbox, grid)
Synchronous	Abstracted Web
Homework	Flags
Supplemental Resources	

<b>Week 7</b>	<b>Digital Canvas</b>
Asynchronous	Lecture: Typography with HTML/CSS
Synchronous	Demo
Homework	25 Variations
Supplemental Resources	

<b>Week 8</b>	<b>Digital Canvas</b>
Asynchronous	Lecture: Designing for the digital canvas (how big is a browser?)



Synchronous	<i>TBD</i>
Homework	25 Variations
Supplemental Resources	

<b>Week 9</b>	<b>Designing for interaction</b>
Asynchronous	Lecture: Programming basic user interactions (:hover, basic JS click, etc.)
Synchronous	Demo
Homework	Stories as Networks
Supplemental Resources	

<b>Week 10</b>	<b>Designing for interaction</b>
Asynchronous	Lecture: Time-based design (interactive states, storyboarding,

	prototyping)
Synchronous	Digital Materiality
Homework	Stories as Networks
Supplemental Resources	

<b>Week 11</b>	<b>Designing for interaction</b>
Asynchronous	Lecture: User models (entering and receiving data, userflows, UX patterns, ways of navigating)
Synchronous	Demo
Homework	Stories as Networks
Supplemental Resources	

<b>Week 12</b>	<b>Networks</b>
Asynchronous	Lecture: Putting a website online (hosting, Github, custom domains)
Synchronous	<i>TBD</i>
Homework	Living Collection
Supplemental Resources	

<b>Week 13</b>	<b>Networks</b>
Asynchronous	Lecture: Accessibility, Metadata (search, social)
Synchronous	<i>TBD</i>
Homework	Living Collection
Supplemental Resources	

<b>Week 14</b>	<b>Networks</b>
Asynchronous	Lecture: Asset creation (video, image optimization, WebGL)
Synchronous	<i>TBD</i>
Homework	Living Collection
Supplemental Resources	

<b>Week 15</b>	<b>Networks</b>
Asynchronous	Lecture: Connecting to other web services
Synchronous	<i>TBD</i>
Homework	Living Collection
Supplemental Resources	