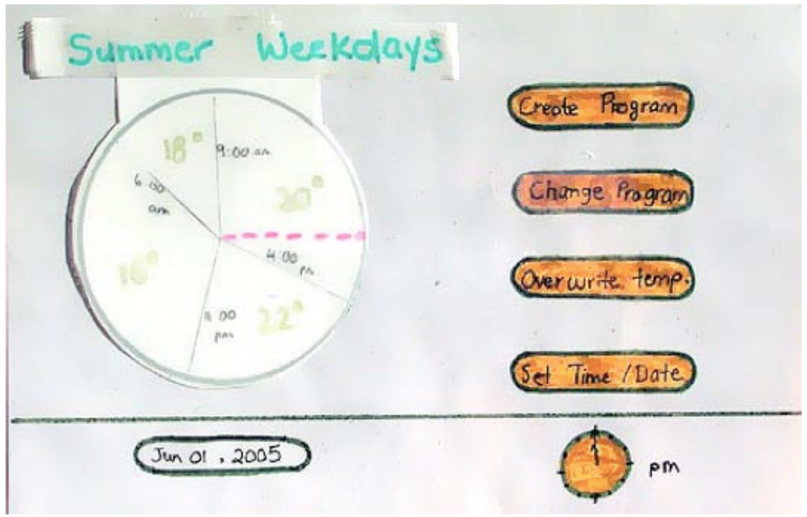
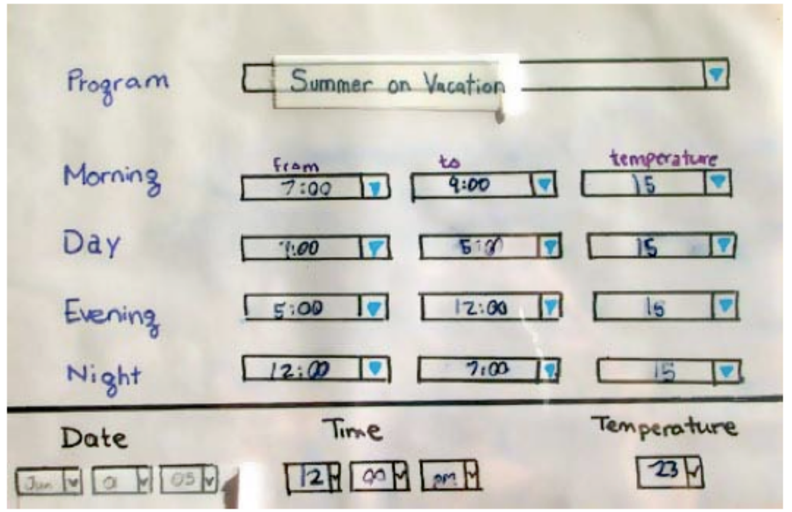
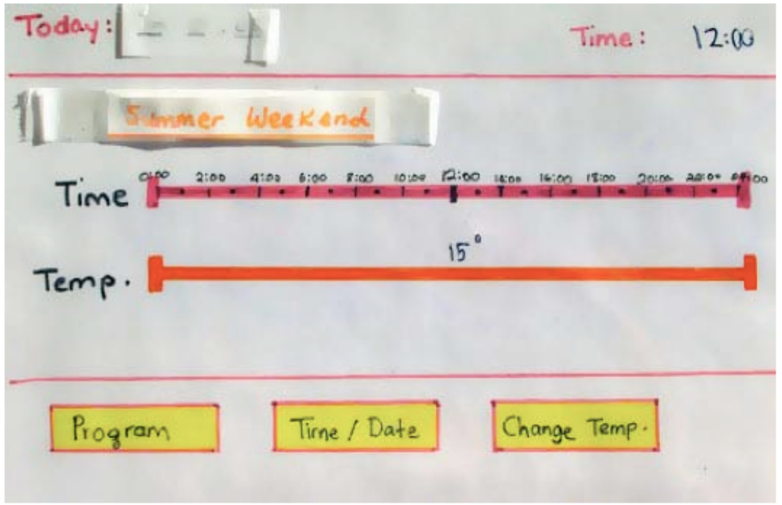
CS181DT Class 13: Prototyping







Three different paper prototypes for the same idea

http://courses.csail.mit.edu/6.831/2014/readings/L10-prototyping/#paper-prototypes

Class 13 agenda

- ZC
- Congrats on finishing Part 1 of the course!
- Lecture: From design goals to designs: prototyping!
- Break
- Studio: Paper prototypes (wireframes)

Announcements

- I’m at a conference next week - class Monday is **on Zoom**, and class Wednesday will be watching a video I’ll record (but I highly encourage you to come to the room and use it as project work time)
- Monday’s seminar will be on Zoom

You've made it through part 1!

1 Making ✅

Make things with many tools to establish proficiency at using tools

4 personal making assignments

- Maker movement
- Design activism
- Analog fabrication
- Digital fabrication
- Creative coding

2 Tools

Make a tool

(Final project, Wizard-of-Oz prototype)

- Brainstorming
- Needfinding
- Prototyping
- Software systems design
- Evaluation

3 Craft

Be critical of computational tools

(Final project implemented)

- Critical design
- Accessibility
- Materiality
- Art
- Power & politics

Revisiting learning goals

The **learning outcomes** of this course are that students will:

- develop a critical understanding of past and present making technologies and design tools, and their roles within current cultural and social contexts ✅ *(partially)*
- make expressive and interactive objects and tools that critique and advance computing culture ✅ *(partially)*
- establish proficiency with the fundamental concepts, methods, and practices of physical modeling, sketching, form giving, and hands-on making across a range of materials ✅
- be able to evaluate tools and interactive systems with quantitative and qualitative methods understand and critique social, ethnographic, engineering, and design-oriented research practices ❌
- improve their capacity to motivate and frame HCI research questions and contributions ❌
- improve their presentation and feedback skills through in class studio critiques ✅

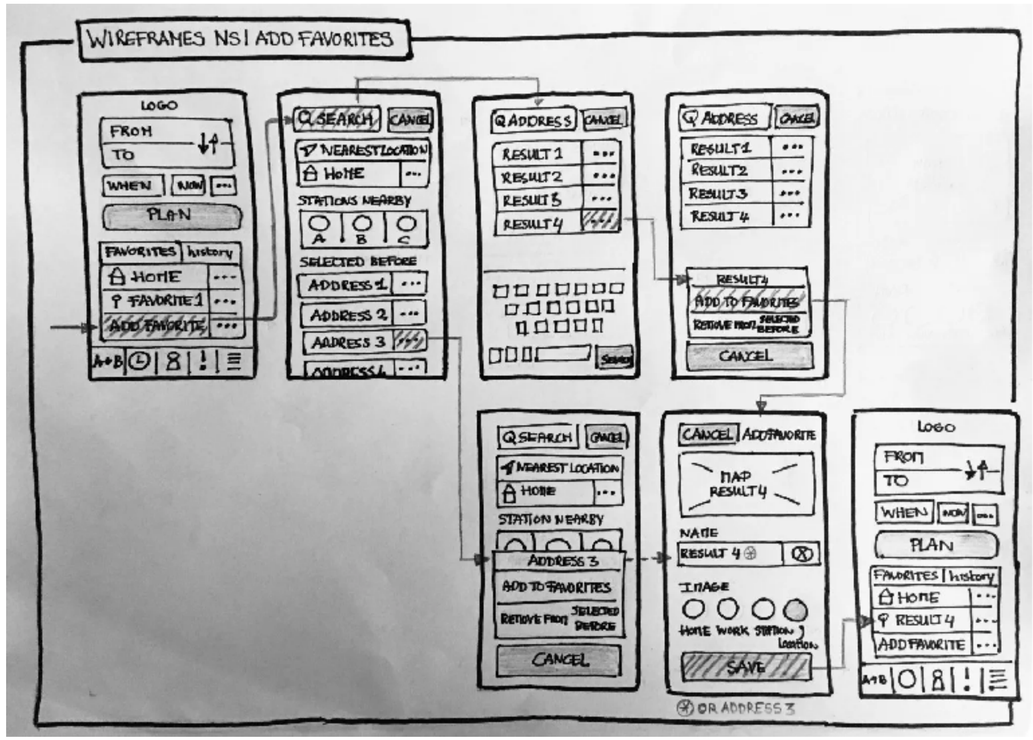
Prototypes

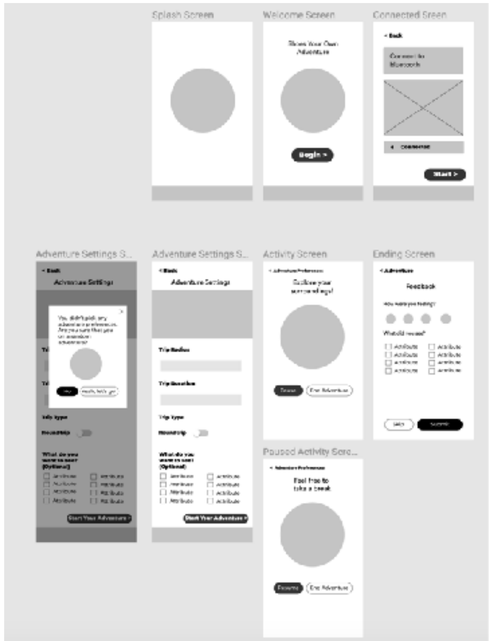
From design goals to design

- Prototypes are instantiations of ideas that can be informed by your user needs and design principles
- Prototypes force you to make design decisions
- This class we'll be making low fidelity *paper prototypes.* Benefits of paper:
  - Faster to build
  - Easier to change
  - Focuses attention on big picture rather than nitpicking UI details

Wireframe

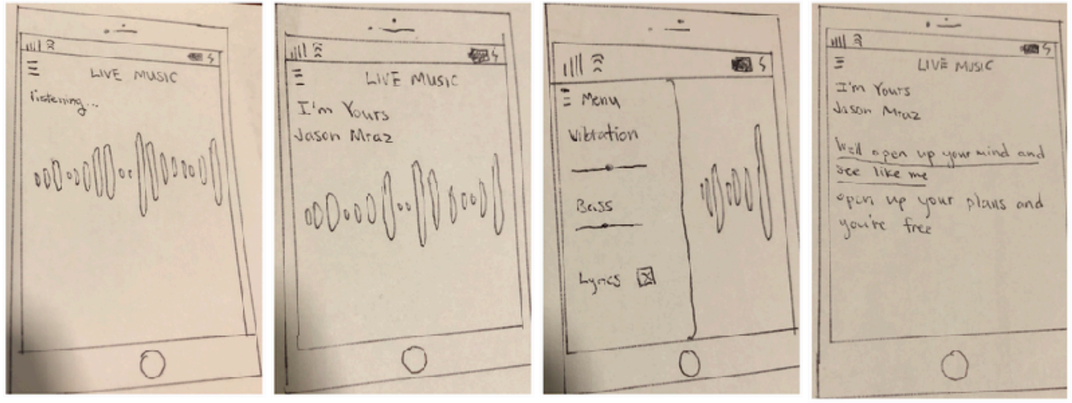
- A wireframe is an **architectural representation** of a user interface like a screen, window, dialog box or web page.
- Wireframes visually depict the basic layout of an app or website. They typically lack style, color, and graphics as their purpose is to test **functionality** and determine **priority of content.**

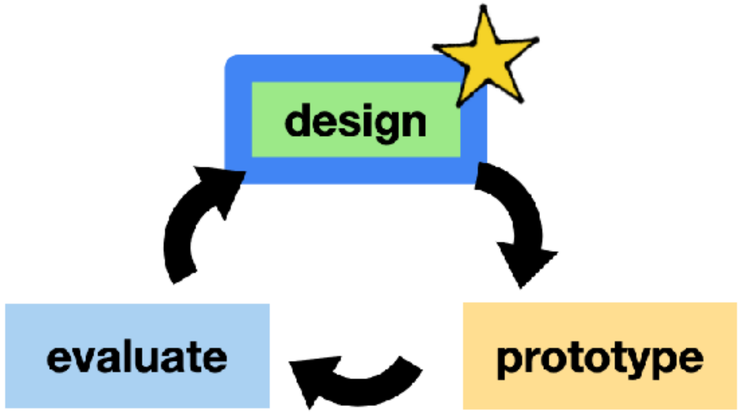




Sketch vs wireframe

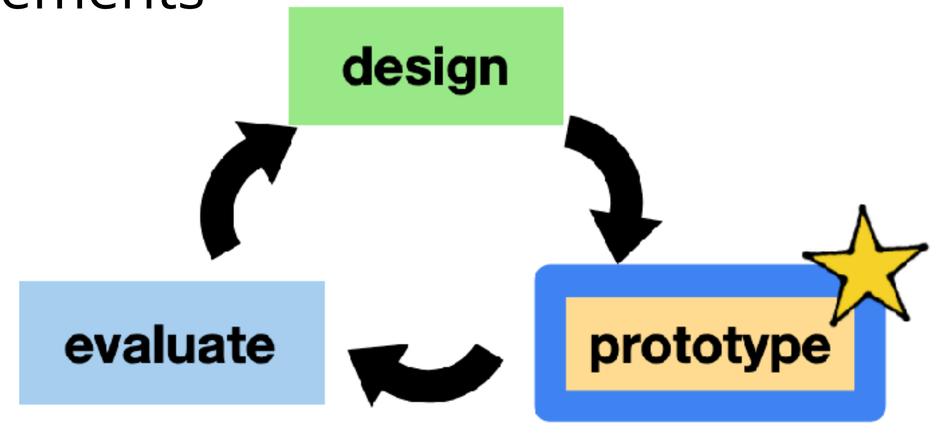
Sketch





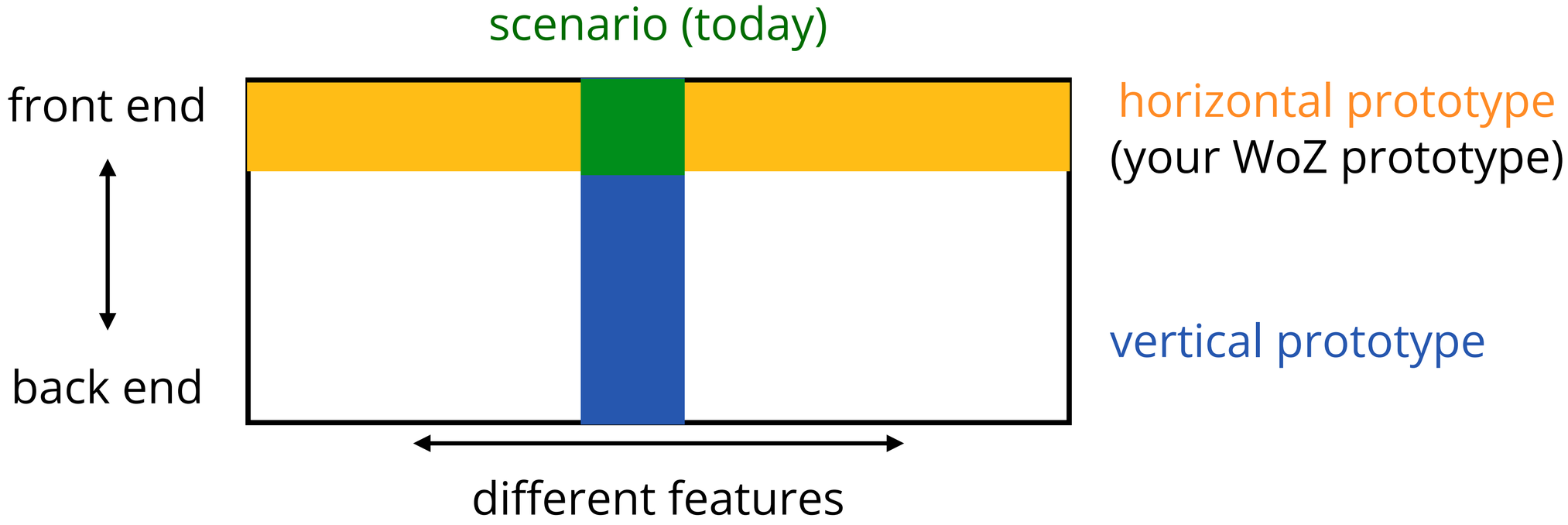
Wireframe

A wireframe fleshes out more details than a sketch, such as specific UI elements



Prototype fidelity is multidimensional

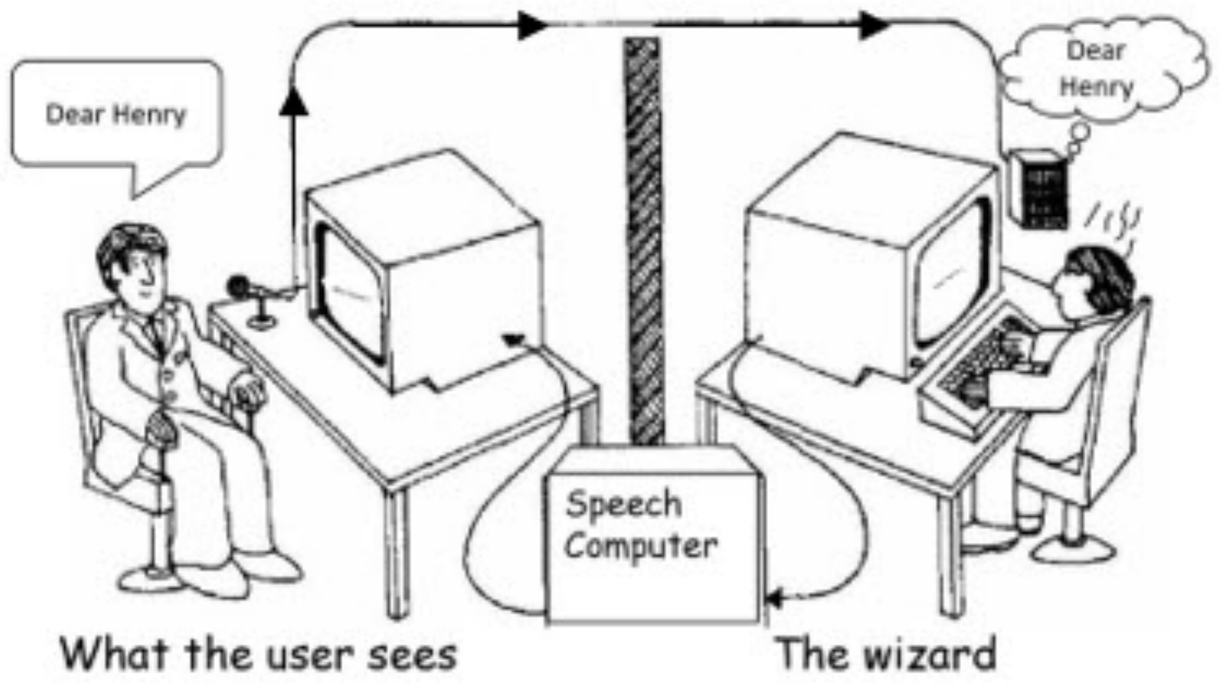
- Breadth (horizontal) - how many features does your prototype cover?
- Depth (vertical) - degree of functionality (how responsive to user inputs is your prototype? Or does it just have “canned” options)



The Wizard of Oz technique

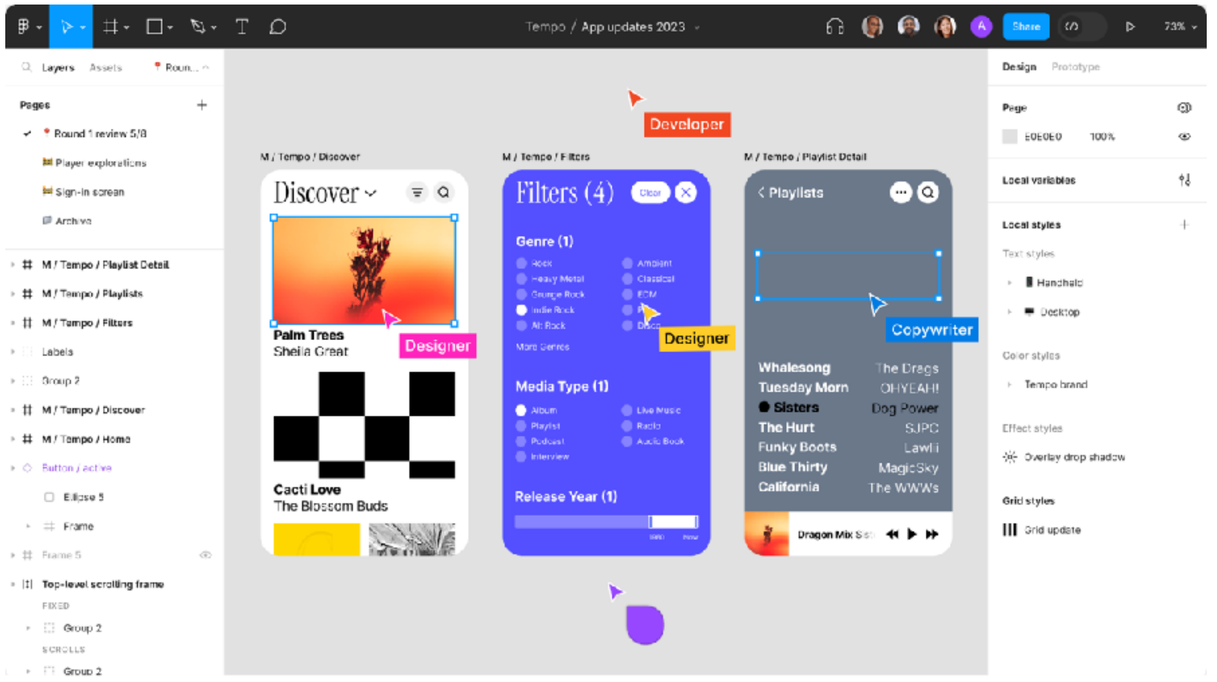
- A setup where subjects interact with a computer system they believe to be autonomous, but the system is actually operated by an (unseen) human being

Wizard of Oz testing – The listening type writer IBM 1984



Your project timeline

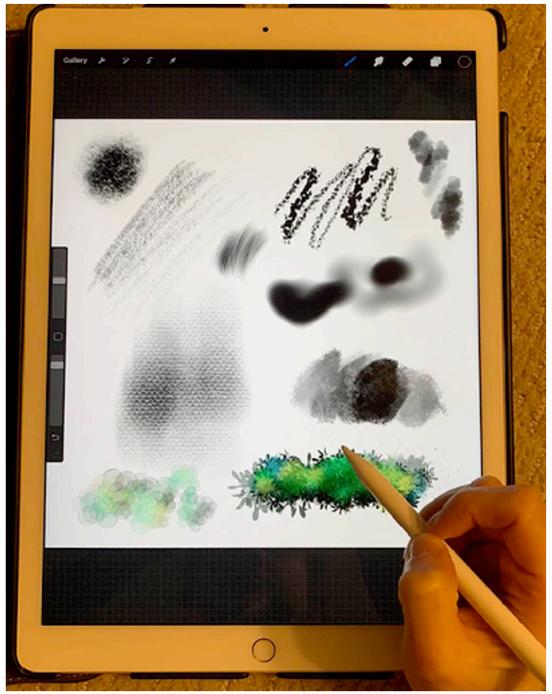
- Today: get started on paper prototyping your main **scenario**
- Oct 28: make & evaluate more scenario paper prototypes
- Nov 6: use **Figma** to make a Wizard of Oz breadth prototype

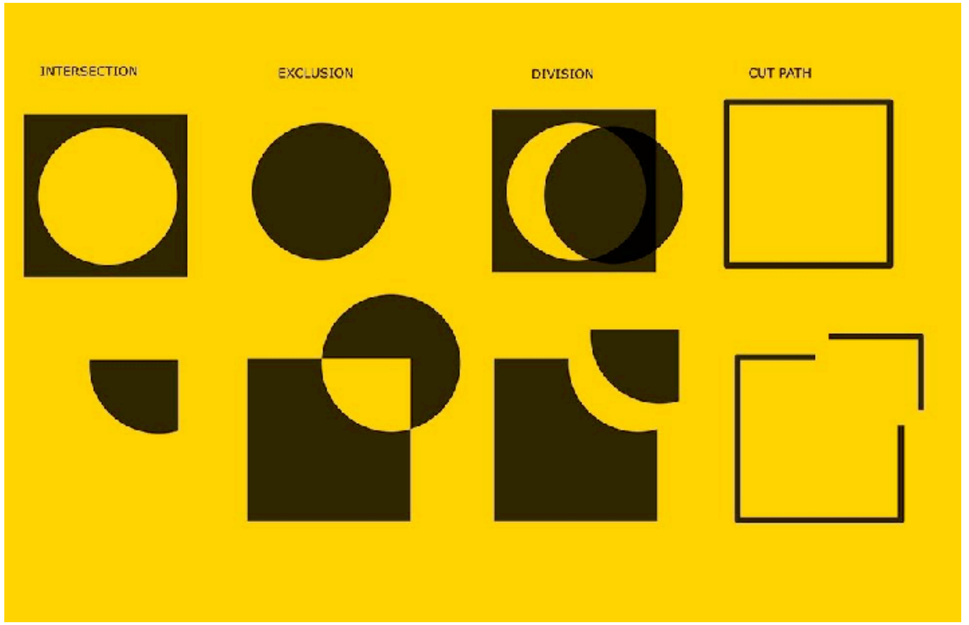


Going back to mental models...

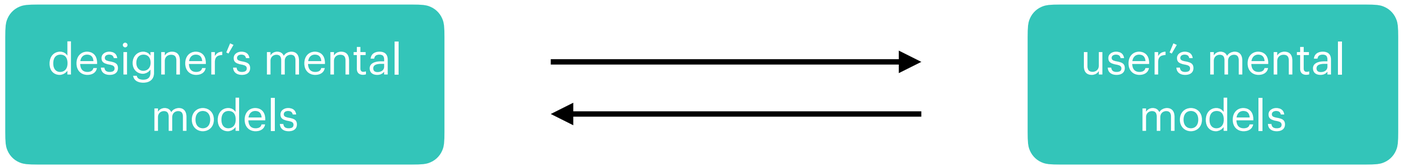
Mental models

- An internal representation of external reality that influences our behavior (Kenneth Craik, 1943)
- Different ways of thinking based on the constraints of the tool
- As we use new tools, we develop new mental models
- Ex: Procreate (mental model closer to drawing) vs Inkscape (new vector and boolean mental model)





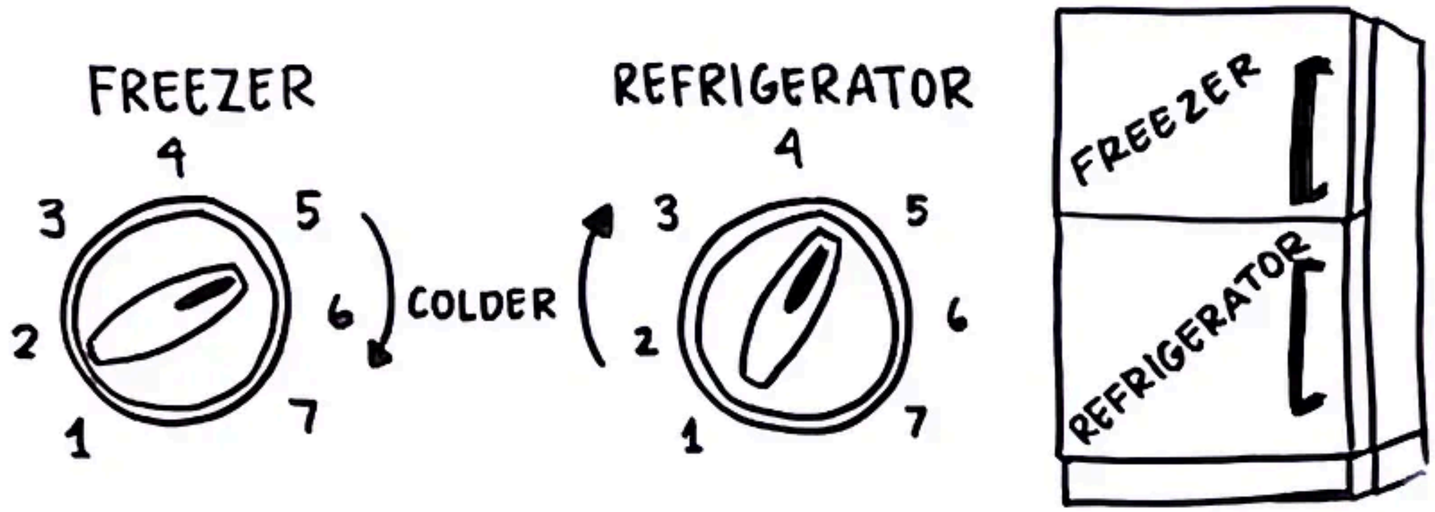
Conceptual model



- As designers, we want to minimize the gap between a designer's mental model and the user's mental model
- That's the purpose of an *interface:* to communicate our design intent without the need for written instructions/documentation
- "Walk up and use" cases

Conceptual model example

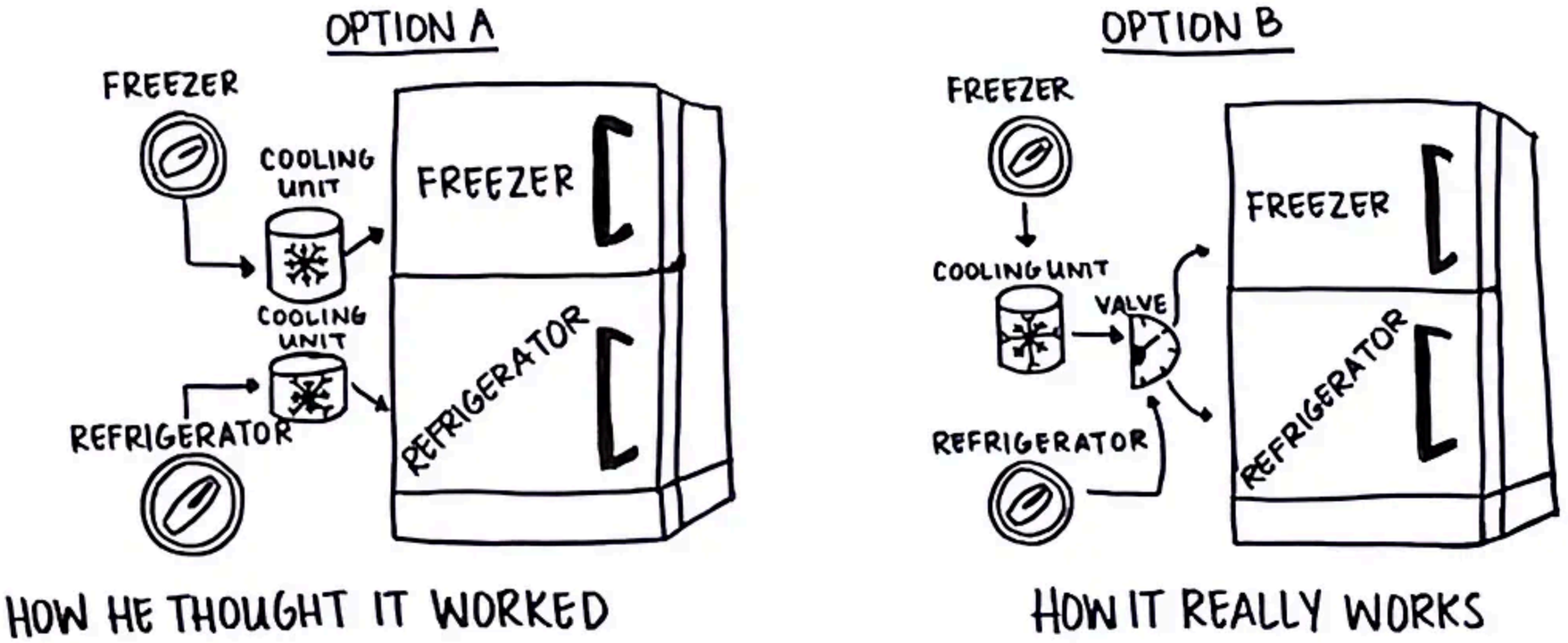
- In *The Design of Everyday Things*, Don Norman gives a fridge example. He wanted to make his freezer colder, so he turned the freezer dial. But it made his refrigerator colder too.



Drawings from https://www.factoftheday1.com/p/march-14-mental-models-with-an-example

Conceptual model example

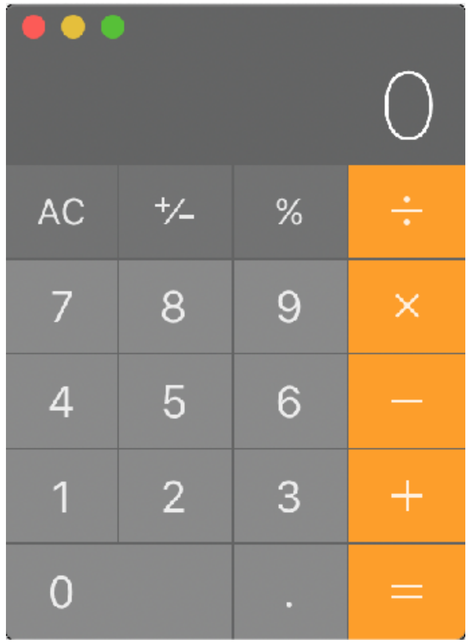
- In reality, the controls were shared: a difference in the designer's versus user's mental models.



Drawings from https://www.factoftheday1.com/p/march-14-mental-models-with-an-example

Skeumorphism

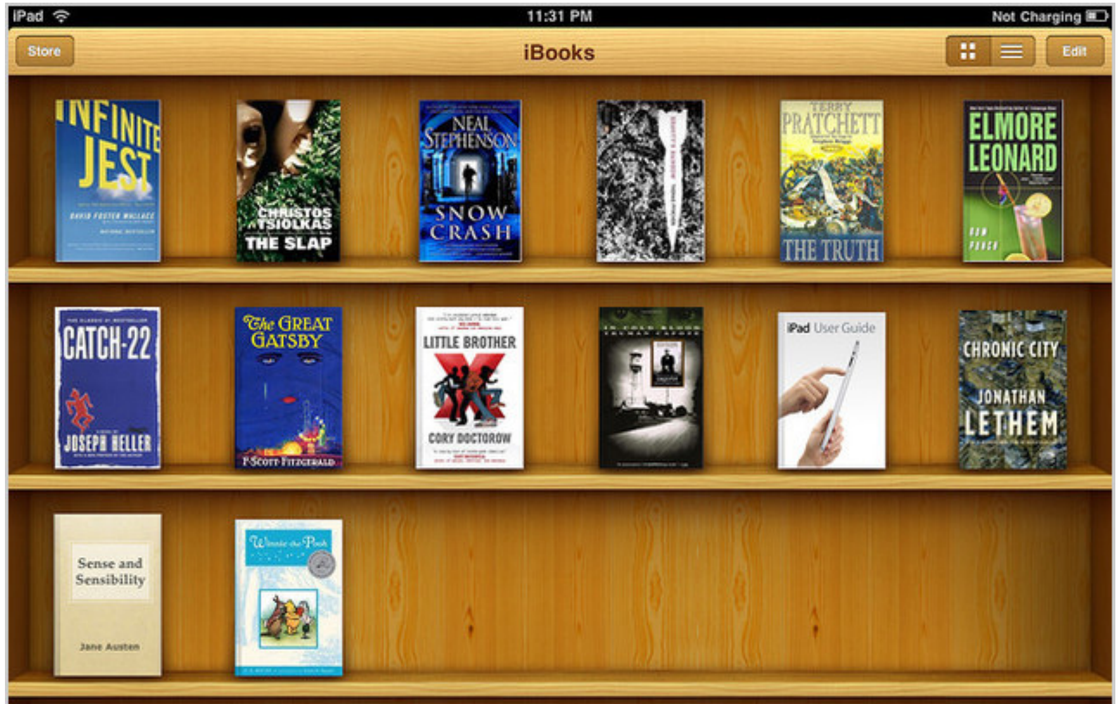
- Using metaphors is a design strategy to help leverage the familiar, but risks in resulting in naive conceptual models
- Skeumorphism is a kind of metaphor where digital GUIs (graphical user interfaces) were made to resemble analog equivalents

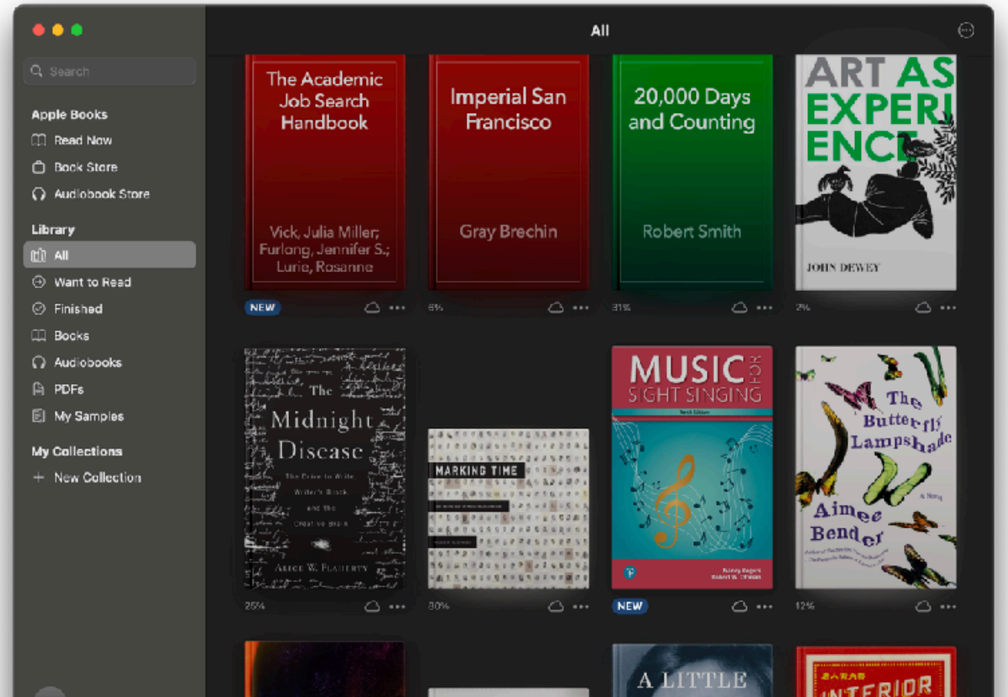


Skeumorphism

- **Pros**
- Reuse learned knowledge
- Repurpose conceptual models
- Clearly shows affordances

- **Cons**
- Limits what users can do
- Implies capabilities that my not exist





3 Design Principles for good design

1. Make controls visible

2. Make mappings clear

3. Provide feedback

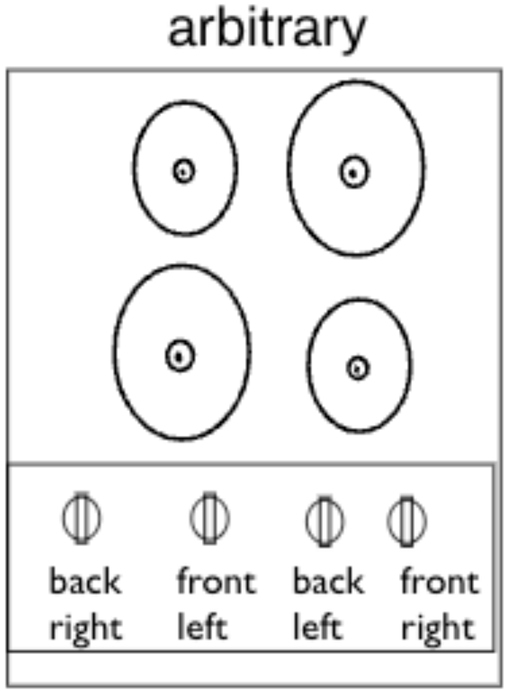
Make controls visible

Poor visibility (how do you set an alarm?)

Too much visibility

Make mappings clear

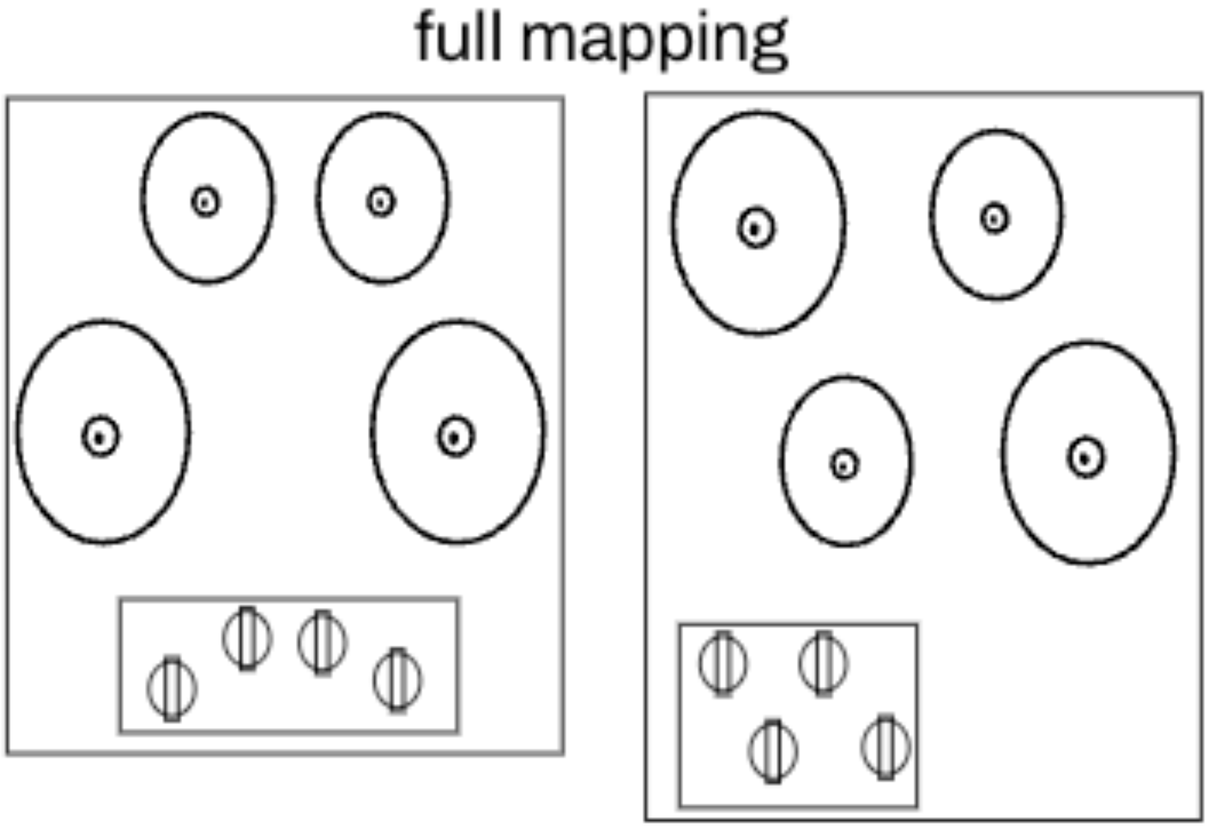
Good mapping for car seat controls



24 possibilities!
Requires:
visible labels
memory

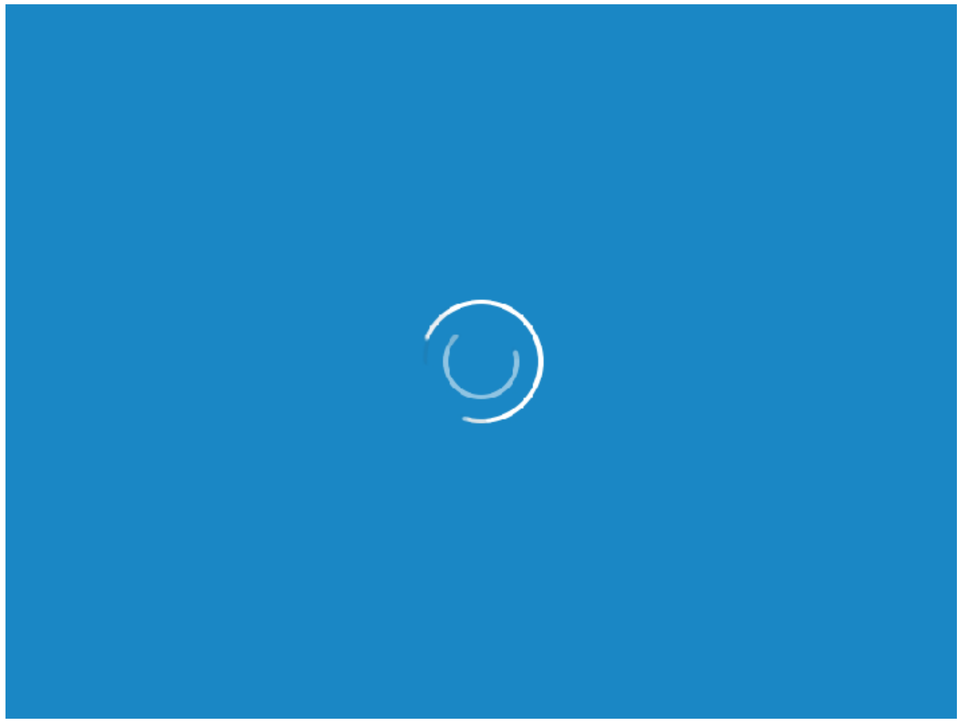
Bad mapping for stove top

Make mappings clear

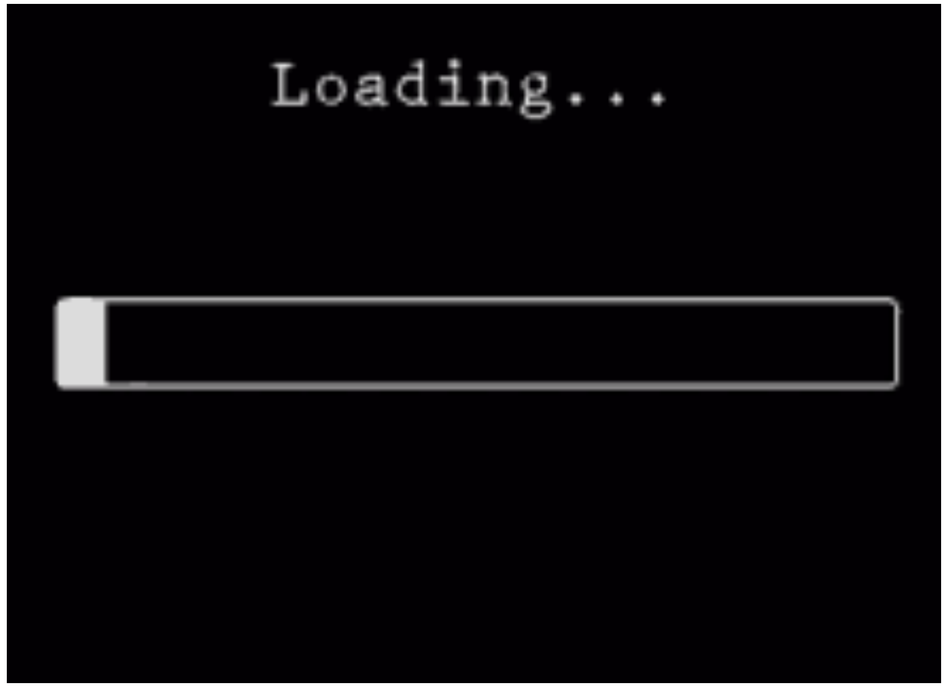


Better: controls placed in the same orientation as burners

Provide feedback



Bad: when will it be done??



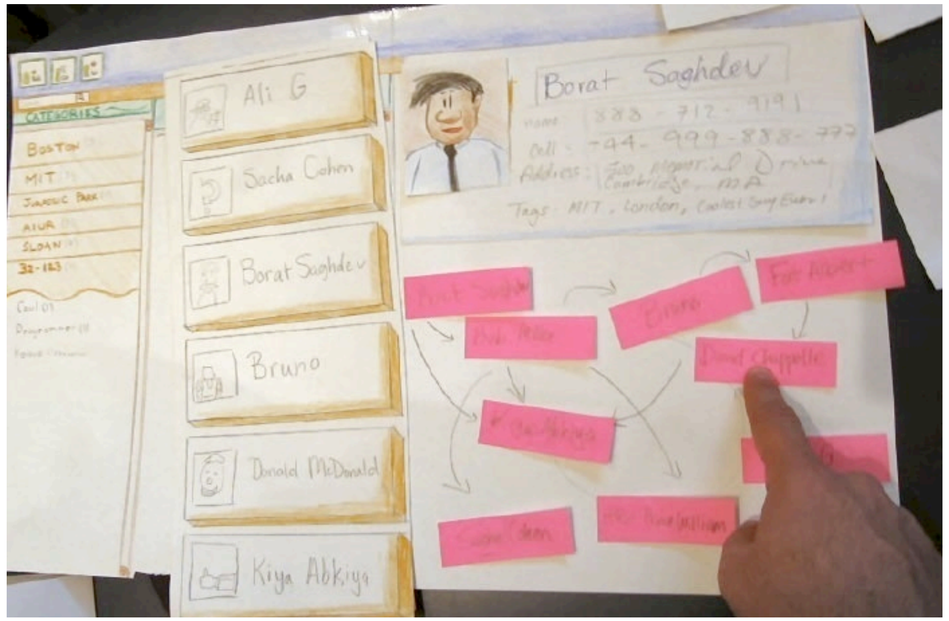
Better: some relative information

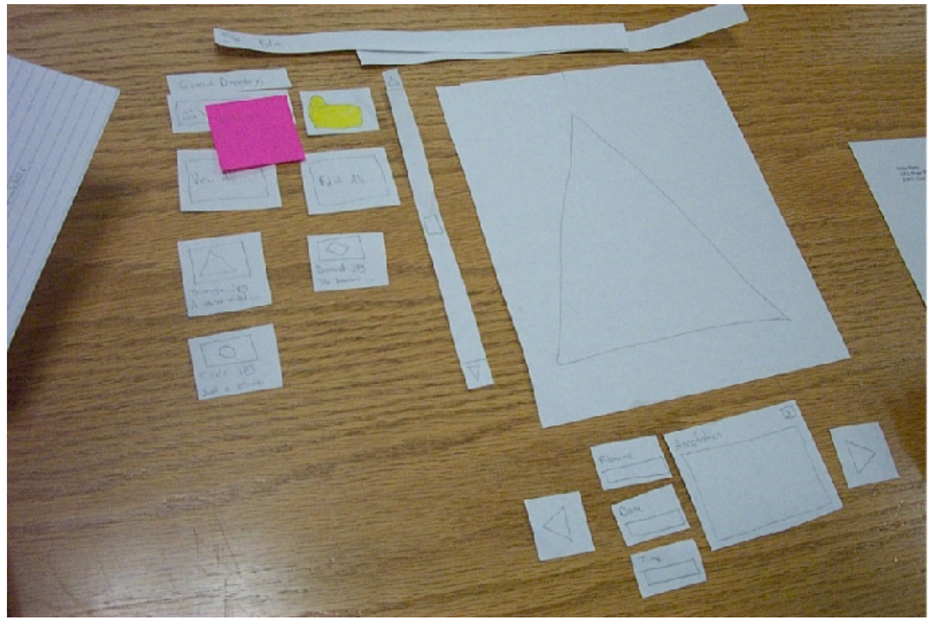
Summary

- Prototypes force you to make design decisions quickly and cheaply. “Fail fast, fail often” before you sink in engineering time
- Much of “good” design stems from matching your and the user’s mental models. To do so,
  - **Make affordances** and controls **visible** but not over-exposed
  - Provide clear **mappings** between controls (inputs) and outputs
  - Provide clear and timely **feedback** in response to user input

Your first paper prototype

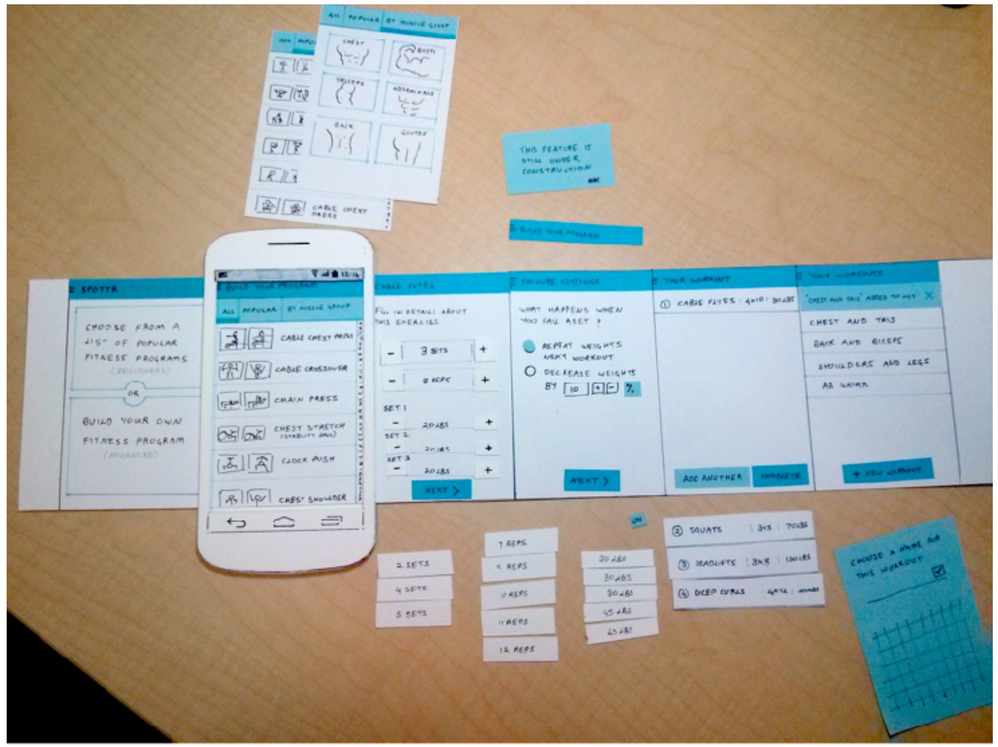
Paper is easier than code

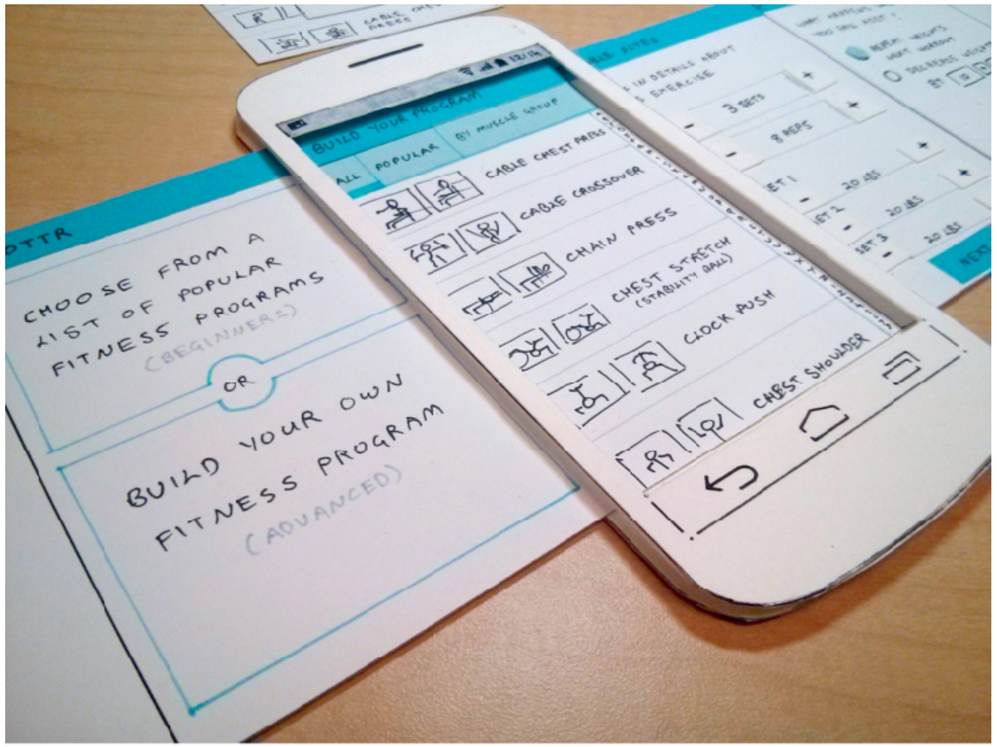




- Making with paper is faster and cheaper than implementing with code
- Use post-its or cut out smaller sheets of paper to replace UI elements

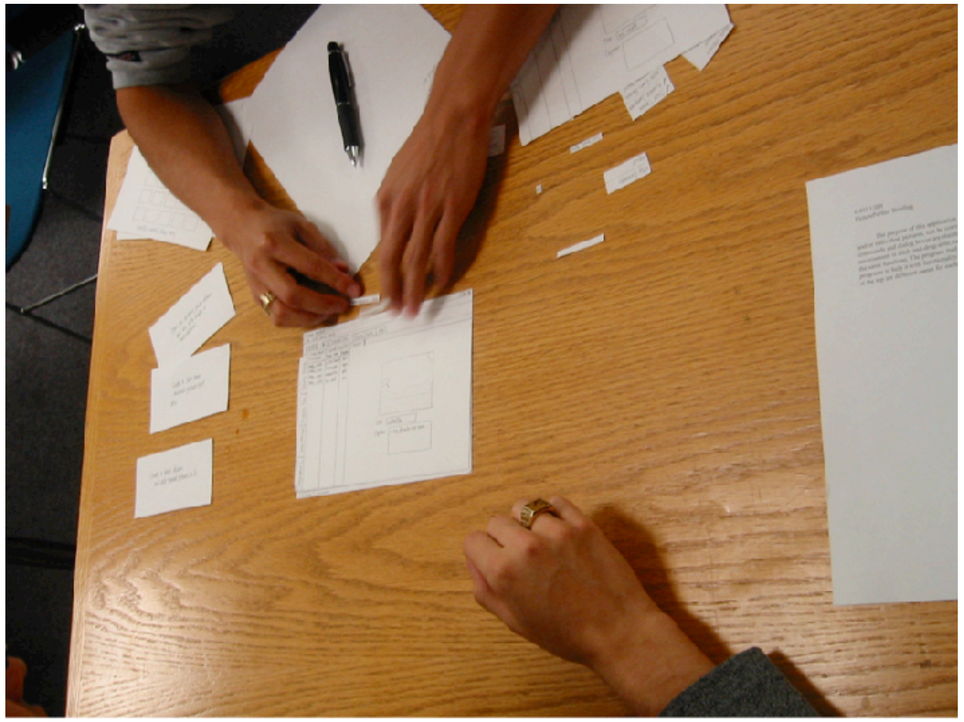
Other ways of interactivity

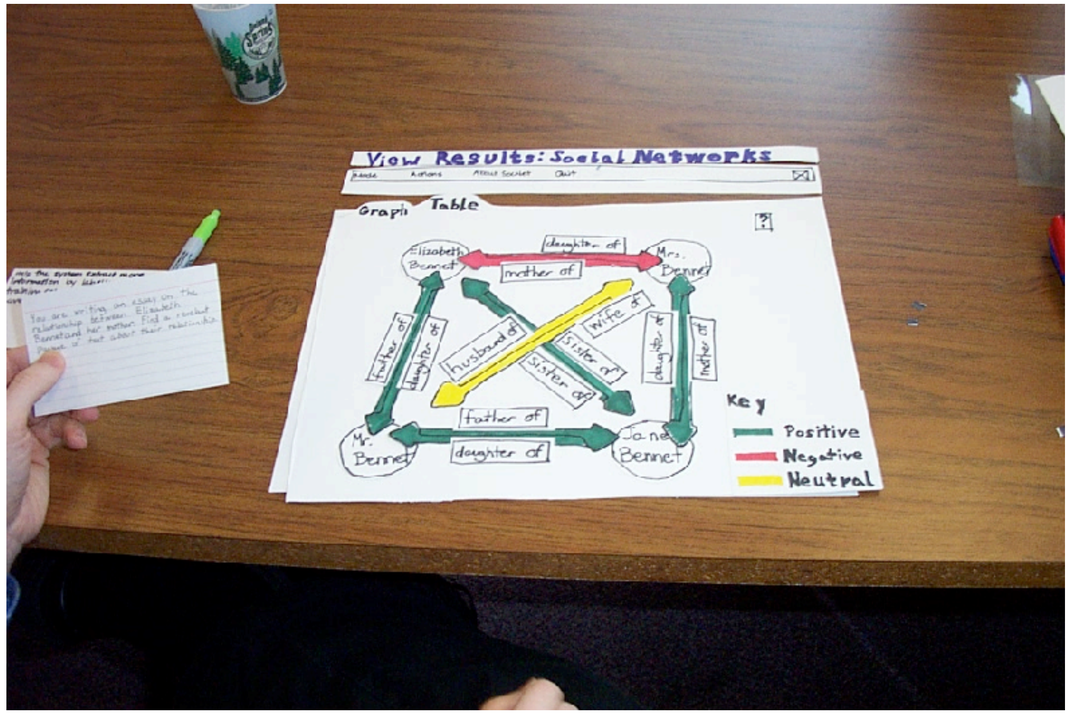




- Make a “frame” (like a phone or laptop screen) and replace screen content inside

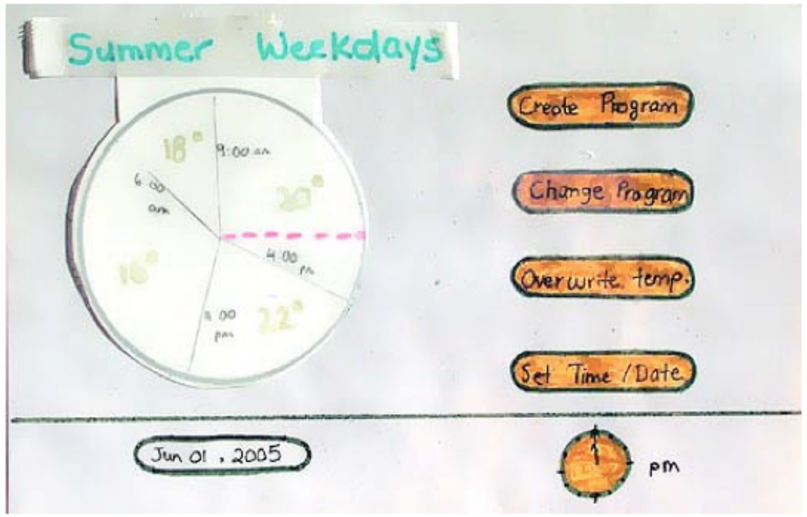
Size matters

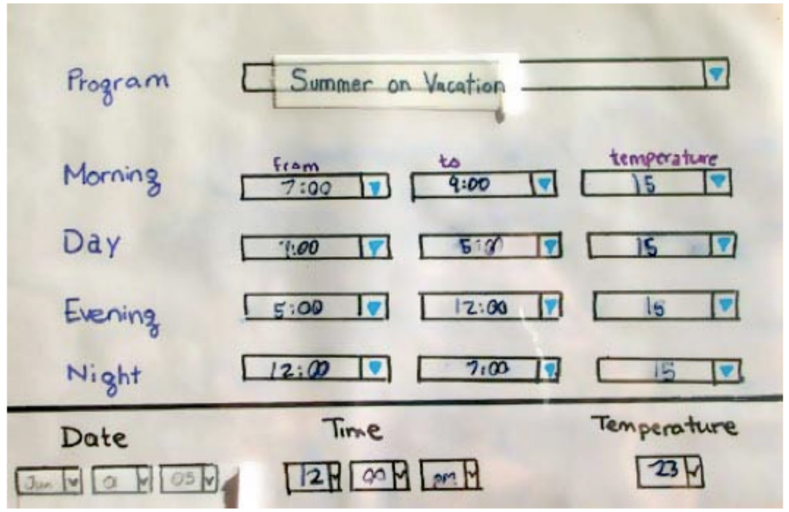


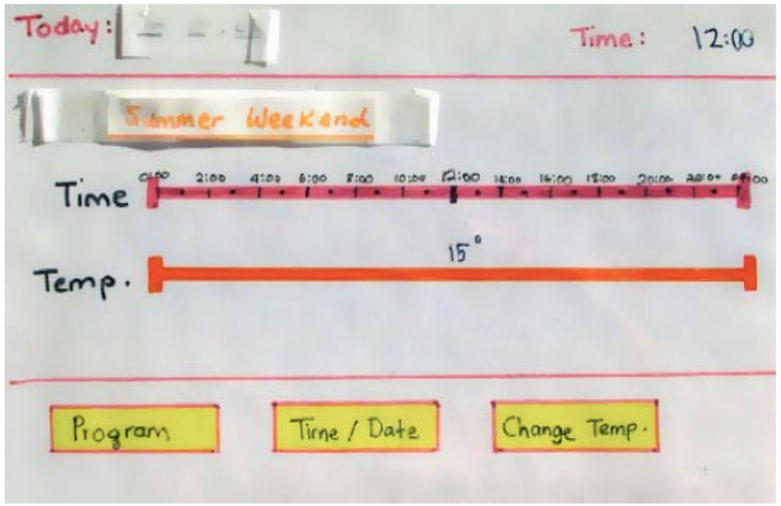


- Bigger prototypes are easier to manipulate and switch out screens
- If you’re making a web app, aim for a full 8.5x11 sheet of paper (a laptop screen size)
- Write big and dark: markers are better than pencil

Multiple alternatives generate better feedback







- When a design is presented with others, people tend to be **more ready to criticize** and offer problems, which is exactly what you want in the early stages of design.

Tohidi, Buxton, Baecker, Sellen. “Getting the Right Design and the Design Right: Testing Many Is Better Than One.” *CHI 2006*.

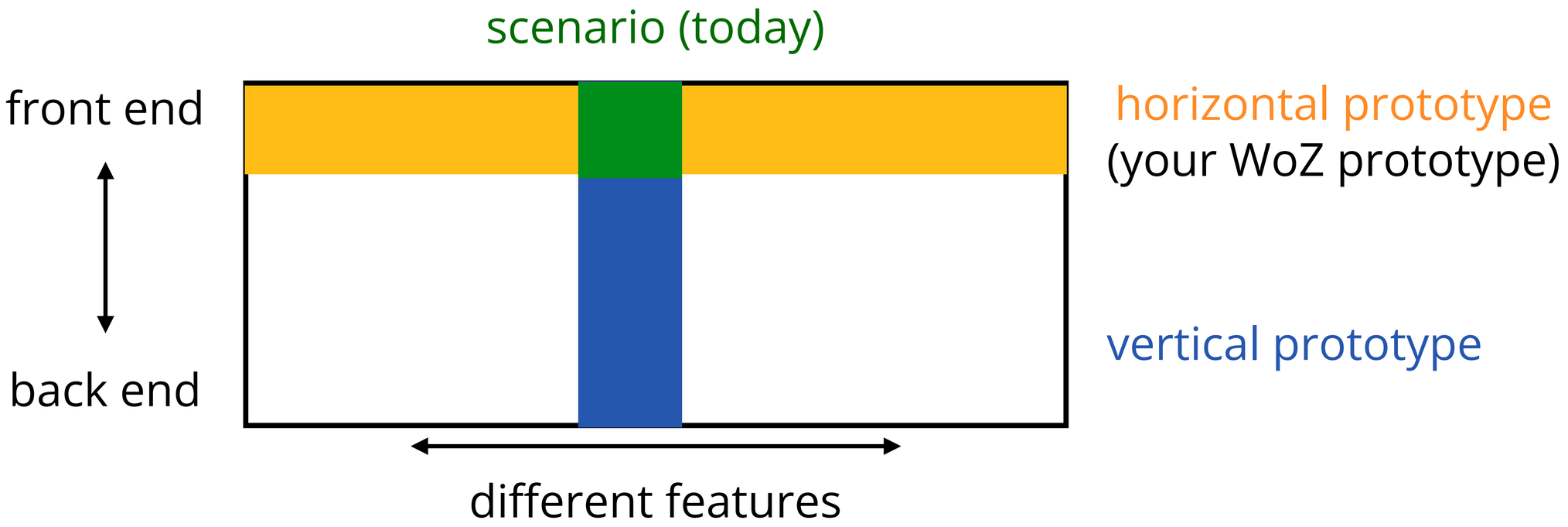
What can and can’t you learn?

- **Functionality:** Does your tool do what it needs to do? Is it missing features?
- **Conceptual model:** Do users understand it?
- **Navigation & task flow:** Can users find their way around?
- **Terminology:** Do users understand symbols or how to interact with UI elements?
- **Screen content:** Are you displaying the right information at the right time?

- Can’t learn…
  - Look & feel
  - Response time
  - Exploration << deliberation: users don’t explore as much with paper prototypes

Step 1: Pick your scenario

- 2 min: As a group, decide what the most important scenario for your tool is (what do users want to do, and how will your tool accomplish it?)



Step 2: Everyone make a paper prototype

- Everyone should make their own first stab at a wireframe to accomplish this task scenario (more ideas generates better feedback!)
    - What elements are on the screen? How does the user interact with them? (**visible affordances**)
    - What options are available to the user? What parameters are important? (**mappings**)
    - How does interacting with the elements change the UI? (**feedback/flow**)
    - How does the user know when the scenario is done? (**feedback**)
- Your next assignment (after needfinding interviews) is to flesh out these prototypes

Class 13 recap

- TODOs:
  - **Mon** (class is on Zoom - I will post the link on Canvas + Slack)
    - RRs due before class
    - Needfinding milestone due 11:59pm
  - **Weds**
    - No synchronous class, please watch a video I will record on motivating your project and writing introductions
    - I highly recommend you use the class time to meet up with your group and work on your paper prototypes - you'll have a *video* prototype due the following Mon (to be explained)