UI design principles

Lecture 22

CS 2112 Fall 2024

Goals and non-goals

- **Goal**: Usability
  - Efficient, easy, enjoyable completion of tasks
  - Focus on user experience (UX) not programmer priorities
- **Non-goals**:
  - Exposing functionality with minimal code
  - Providing as many features as possible
  - Giving users what they think they want
    "If I had asked my customers what they wanted, they would have said a faster horse." –Henry Ford

Principle 1:
Know your user

Design to your user

- Frequent or occasional?
- Novice or knowledgeable?
- Training?
- Don't design for yourself—
  you are not the user
- Understand needs: talk to/watch users

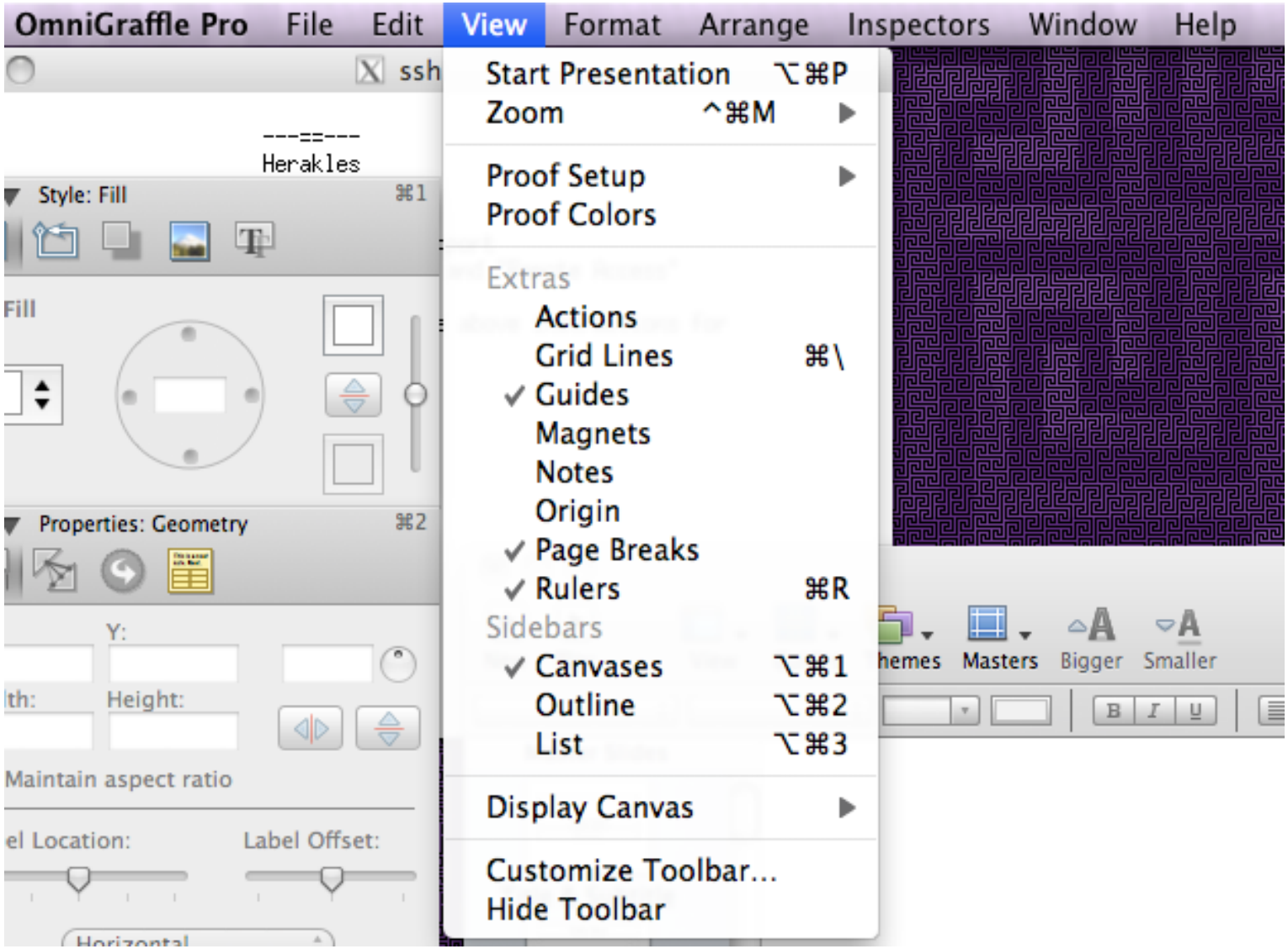
Know your user.

Novice users

- Gentle learning curve: **discoverability**
  - Standard way for user to find all functionality
  - vs: “long press”
- Protection from dangerous actions
- Clarity: simple displays, consistency with other applications and real world
  - E.g., using icons as metaphors

Know your user.

Discoverability



Know your user.

No loaded guns

Know your user.

Frequent/power users

- Optimize for *efficient interaction.*
- Usability ≠ user-friendly
- Powerful actions, short interaction sequences (e.g., hotkeys)
- Rapid response times
- Rich controls, shortcuts for common actions
- Exploit muscle memory
- Information-rich displays
- Customization and macros

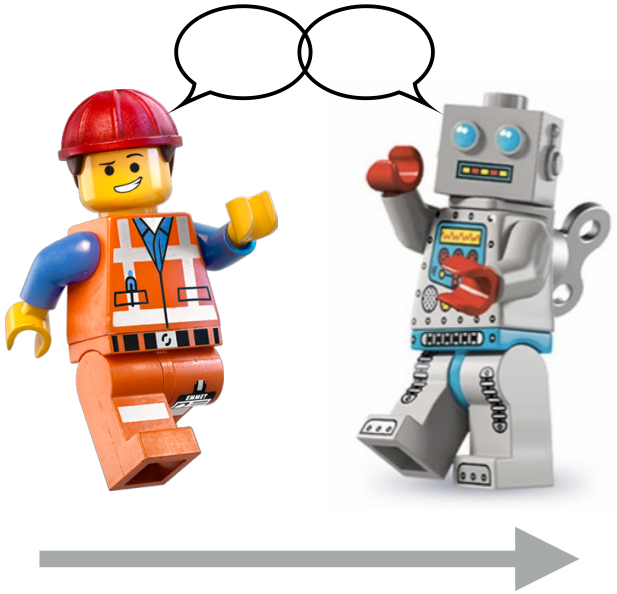
Know your user.

Power-User UI

feature-rich
information-dense
efficient actions

Know your user.

Principle 2:
UI is a dialogue



UI: good conversation partner?

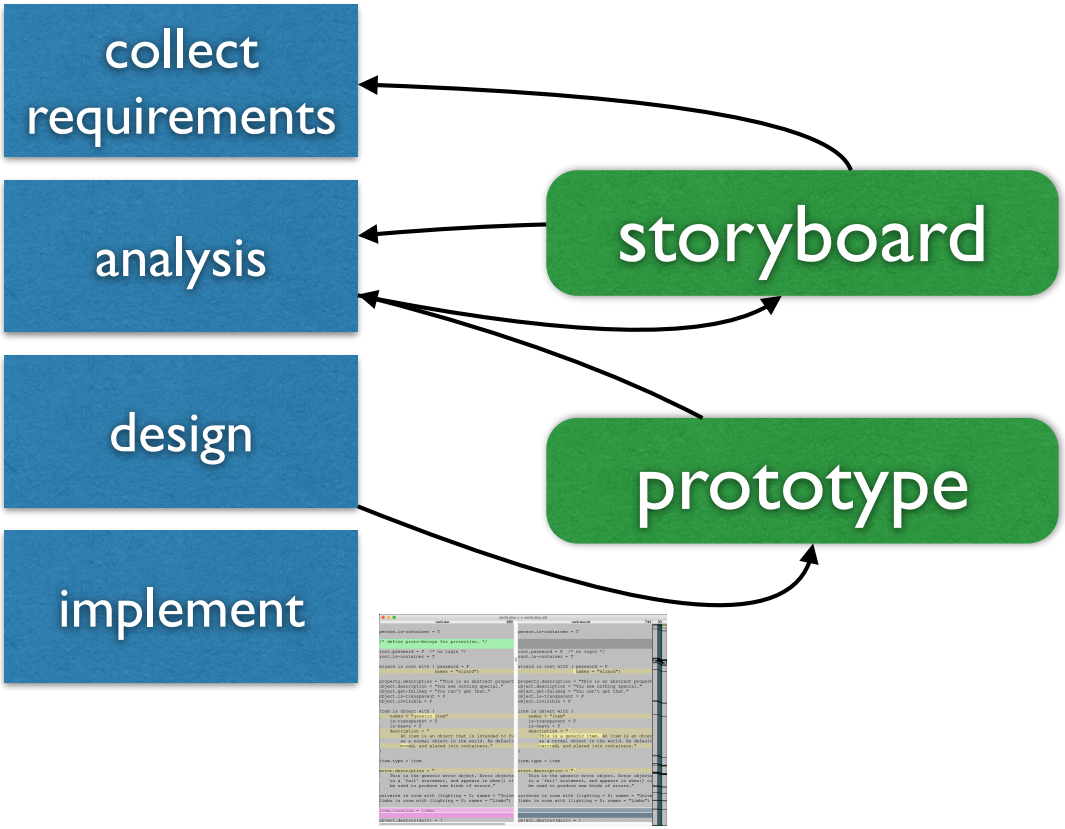
- Goal: good user experience (UX)
- Ratify actions quickly
- Be responsive
  (e.g., highlight buttons and other affordances)
- Show progress of longer actions

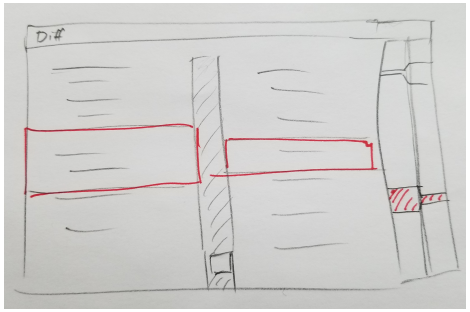
Know the conversations

- Identify and storyboard **use cases** to figure out what users will have to do
- Eliminate unnecessary user actions (e.g., needless confirmation dialogs)
- Short interactions with clear progress: **intermediate goal satisfaction**
- User testing to find your blind spots (as developer)
- Testing scripts help human testers to achieve coverage

Storyboarding and prototyping

- Avoiding design lock-in — early feedback loops avoid polishing a bad design.
- Storyboard 3 designs and discard 2!





Direct manipulation vs. I/O

Know your user. UI is a dialogue.

Interaction paradigms

- **Direct manipulation**: the UI *is* the underlying data/behavior model
  - User view: Model = View = Controller
  - Implementation: Model ≠ View ≠ Controller

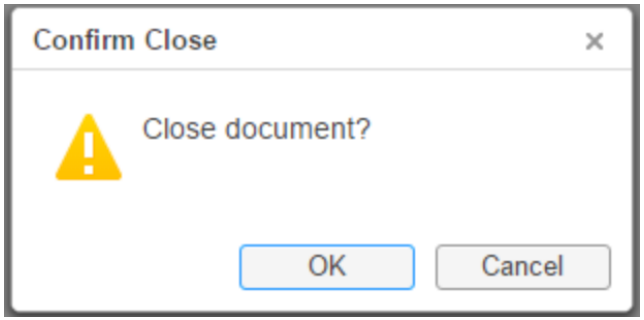
- **I/O**: UI generates output when input provided (UI ≠ model)
  - e.g., menus, submitted forms, command shells

Interaction time scales

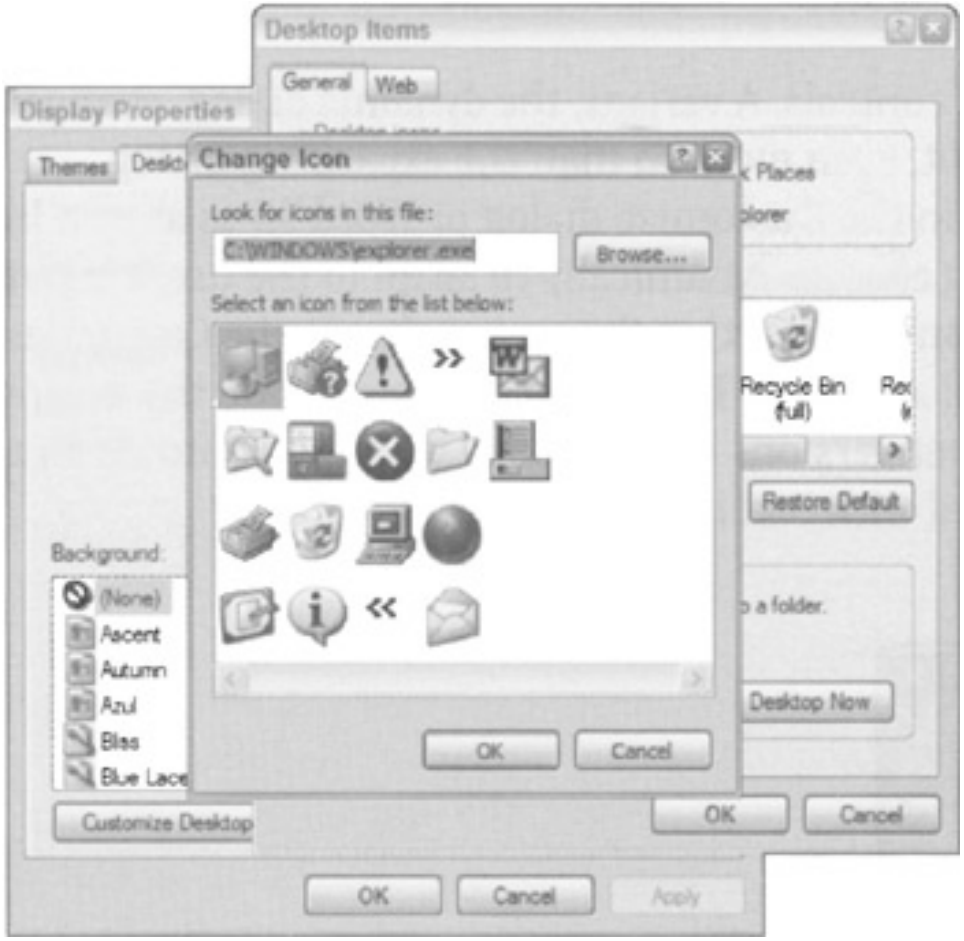
- 1/60s: biologically imperceptible: faster than neurons
- 1/30s: just fast enough for continuous-feedback tasks (e.g., mouse tracking)
- 1/10s: imperceptible delay for discrete actions, e.g. button clicks.
- 1/2s: fast but noticeable (ok for command-response interaction)
- 1/2s–5s: increasingly annoying but user stays focused
- 5s–10s: User starts to lose attention and productivity declines.
- 10s–1 min: User highly distracted. App must support parallel activities to keep focus.
- >1 min: Huge loss of productivity. User leaves for coffee, chats with friends.

Modes

- Modes: states of UI that limit the possible interactions.
  - Good: restricted context-sensitive action "vocabulary" simplifies user interaction on current task
  - Bad: can be confusing and can trap users
- Moral: use judiciously

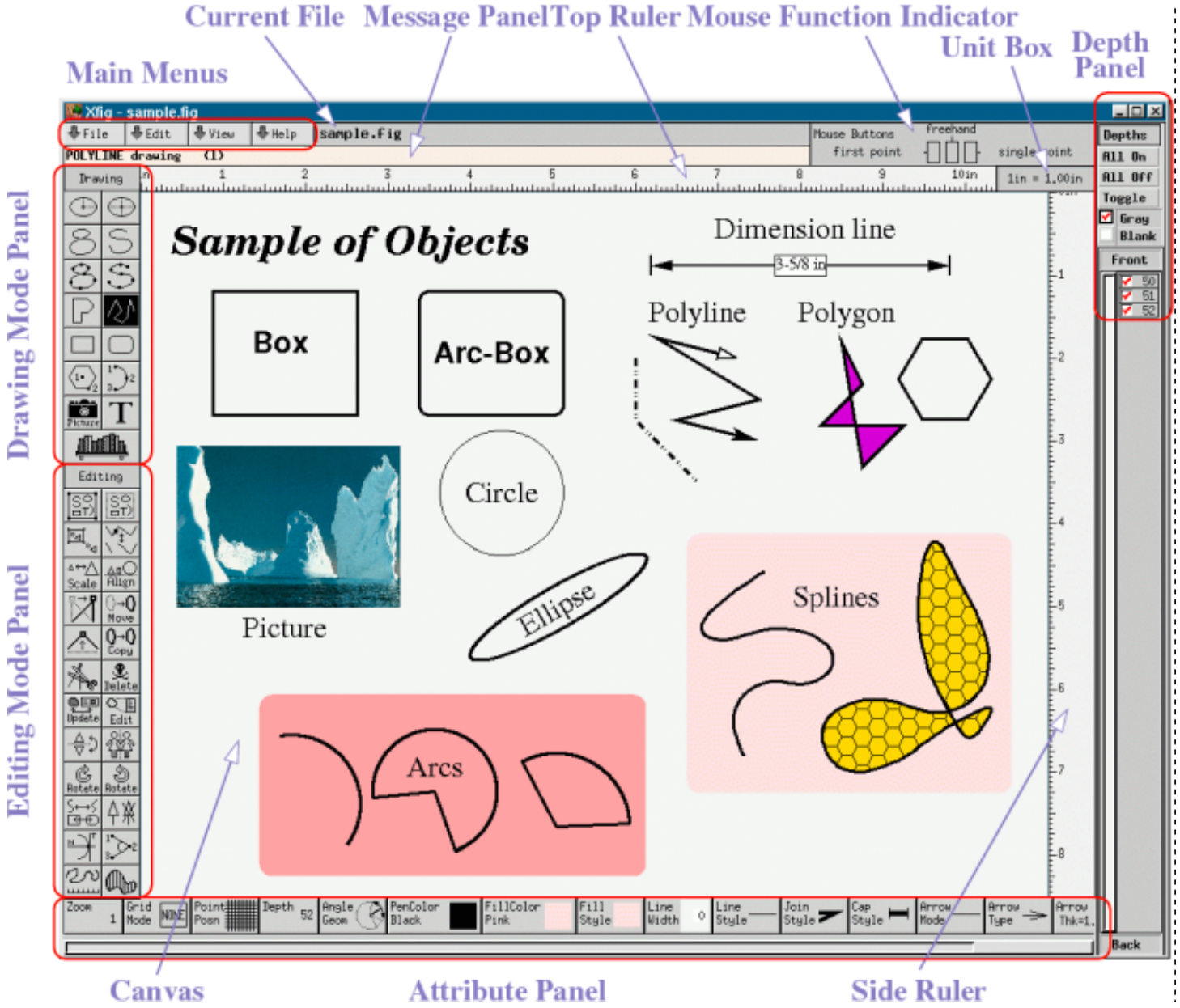


Modes gone bad: cascading dialogs



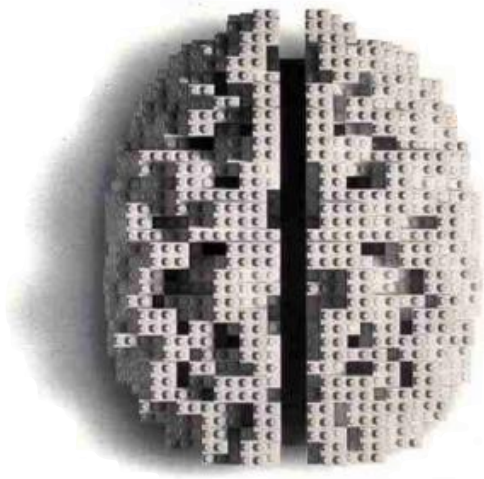
Know your user. UI is a dialogue.

xfig: the context-sensitive mouse



Know your user. UI is a dialogue.

Principle 3: Aid Memory



"The advantage of a bad memory is that one enjoys several times the same good things for the first time."

— Friedrich Nietzsche

Keep models simple

user interface

model

- Good models of data and operations are easier to learn and to remember
  - based on familiar abstractions
  - describable as narrow interface with simple spec

Trash model

```
class Trash {
    /** Move file to trash but remember where it was. */
    void discard (File f);

    /** Remove file from trash and
     *  restore it to its previous location. */
    void putBack(File f);

    /** Permanently delete all files in the trash. */
    void emptyTrash();
}
```

Simple, familiar, narrow

Rule of 7

- Humans can hold at most 7 things in their head at once

  ⇒ Avoid long menus, arrays of buttons

  ⇒ Avoid complex modes

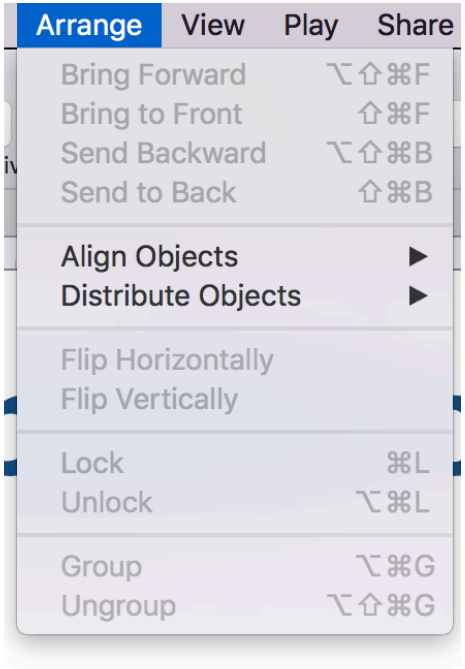
Spatial memory

- Human spatial memory is amazingly good (e.g., memory palaces).
  ⇒ Good UIs exploit it

- Each window or dialogue or mode is a "place" for interaction
  - make it a nice place to be
  - avoid unnecessary places/modes
  - make navigation easy, obvious
- Big-picture views strengthen spatial sense

Spatial organization

- Place things that belong together close by
  - related functionality
  - used in same workflows

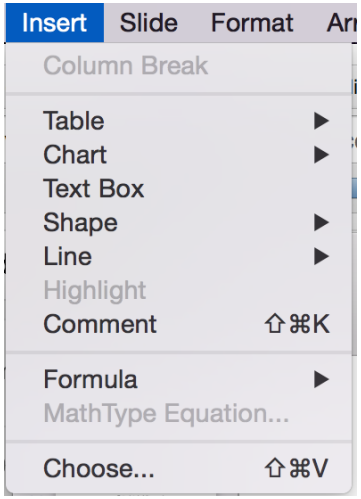


Muscle memory

- Frequent users don't need to look – UI is programmed into their muscles

$\Rightarrow$ action needed to activate functionality should be consistent

  - e.g., gray out menu items, don't remove them



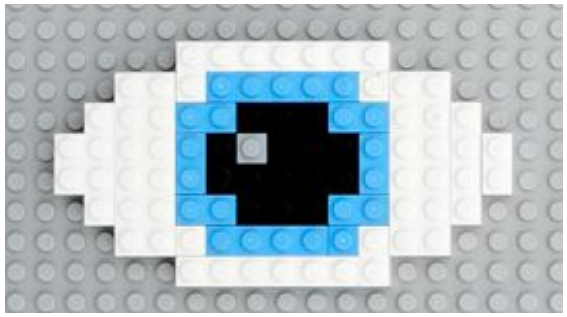
Context-sensitive help

- Help should be about what user is doing now.

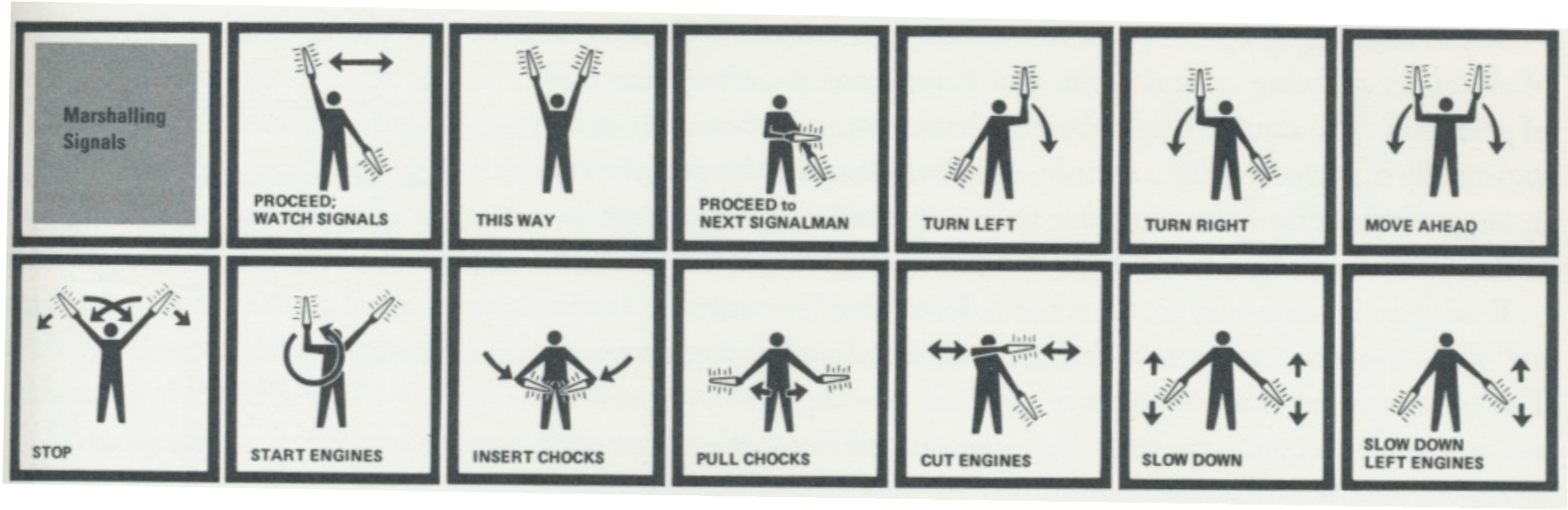
$\Rightarrow$ task-focused rather than feature-focused (unlike many modern apps!)

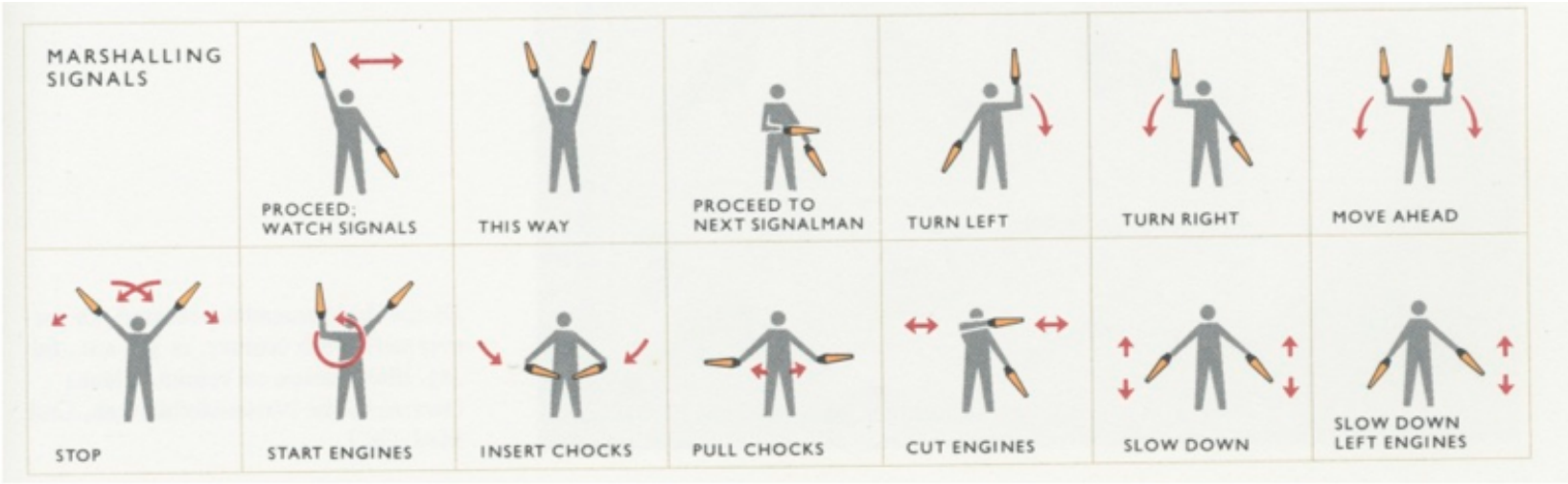
$\Rightarrow$ modes provide context

Principle 4:
Visual design matters



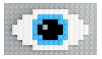
Avoid visual clutter





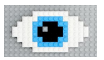
Tufte. Envisioning Information

Know your user. UI is a dialogue. Aid memory. Visual design matters.

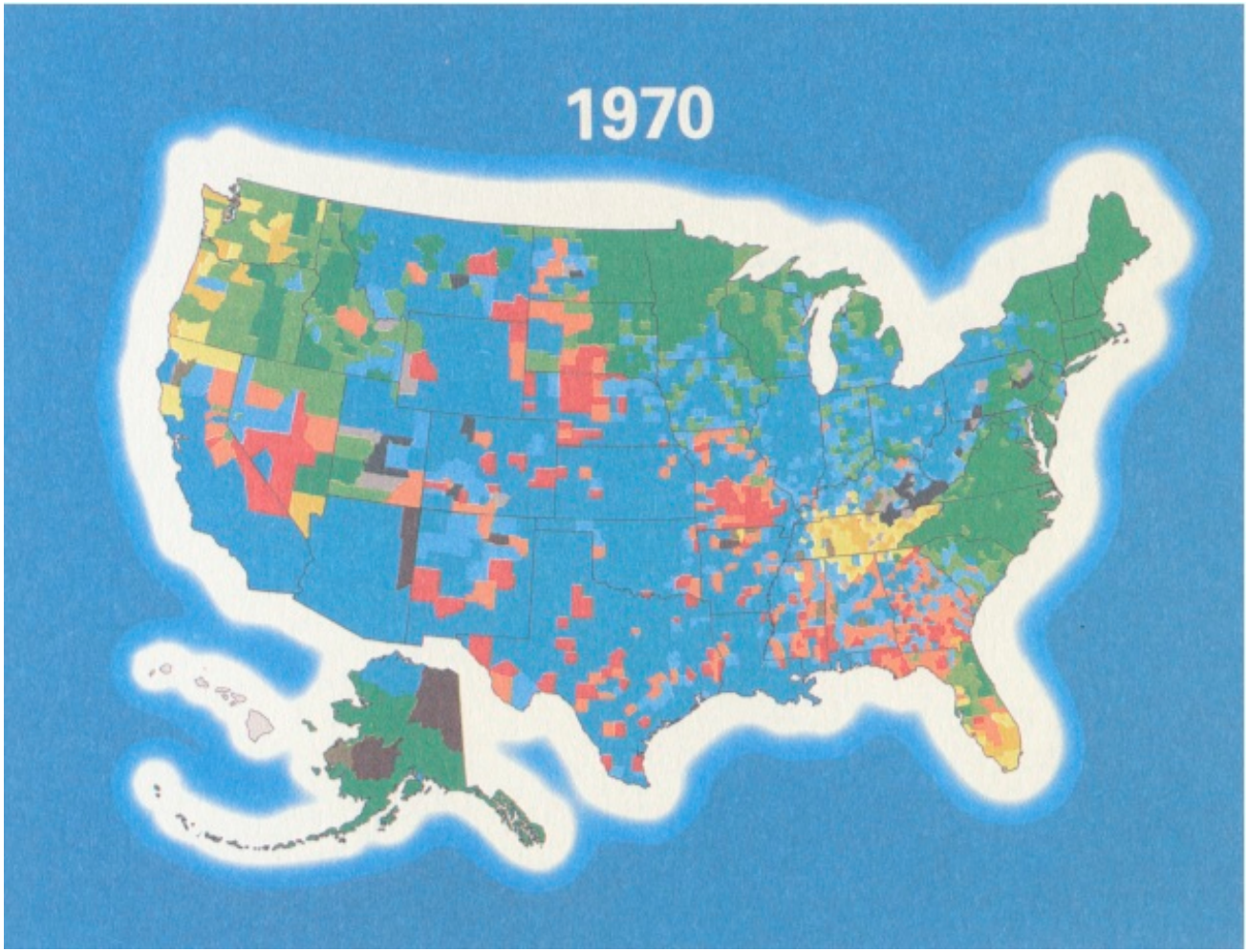


Avoid visual clutter

- Maximize information/“ink” ratio
- Use low-contrast separators to organize
  - space shading, color instead of lines



Good use of color and contrast?



Use high contrast, avoid chromatic aberration

Low-contrast text is not much fun to read.

And it gets harder if the font size is small.

Visual consistency

- For novice users, be *externally* consistent with existing apps and real world
- For expert users, be *internally* consistent
  - e.g., buttons that navigate vs. buttons that change state vs. buttons that expose new information
  - write **style guide** so developers maintain consistency.

COLORS

ON-SCREEN

Computers, televisions and other electronic displays use combinations of red, green and blue to simulate a large gamut of colors. Many displays are able to show millions of colors and can often display colors more vivid than printers.

ON-PRESS

Standard printing process, from desktop ink-jets to industrial offset lithography use combinations of cyan, magenta, yellow and black inks to simulate a large gamut of colors. This method is not able to produce the full color spectrum and lacks strongly-saturated green and orange.

CUSTOM INKS

The Pantone Matching System by Pantone, Inc. is a proprietary method of mixing inks to exacting standards. It allows highly-accurate color reproduction and is key in unifying corporate communications. PMS numbers, as listed at right, are to be used when designing and printing materials.

PRIMARY COLOR

The primary color is an off-black that screens to a subtle brownish color. It is to be used in one of the four manners shown at left.

This color allows for adequate contrast in type and photography while retaining approachability and warmth.

PMS 412 C 0 M 30 Y 66 K 98

100% 60% 15% MONOTONE IMAGERY

DO NOT

- Mix with other colors
- Screen at percentages other than sixty or fifteen

ACCENT COLOR

The bold and warm accent color provides variation to printed materials that builds hierarchy, captures attention and evokes the bold nature of GIRAFFE HEROES.

PMS 193 C 0 M 100 Y 66 K 13

100%

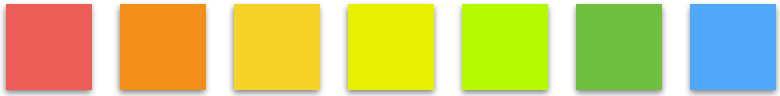
DO NOT

- Mix with other colors
- Use anything but 100%

giraffe project

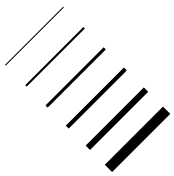
COLOR

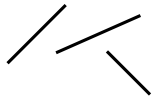
Visual features



- Shape: up to 15
- Color: up to 24
  - But: color perception varies!
  - 8% M, 0.4% F have altered perception

  ⇒ color should only *complement* other sources of information



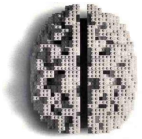


- Size, length, thickness: up to 6
- Orientation: up to 24

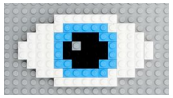
UI design principles

- Know your user

- UI is a dialogue



- Aid memory



- Visual design matters