

UI design principles

Lecture 19

CS 2112 Fall 2017

Goals and non-goals

- **Goals:**
 - efficient, easy, enjoyable completion of task
- **Non-goals:**
 - Exposing functionality with minimal code
 - Providing any 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.”

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

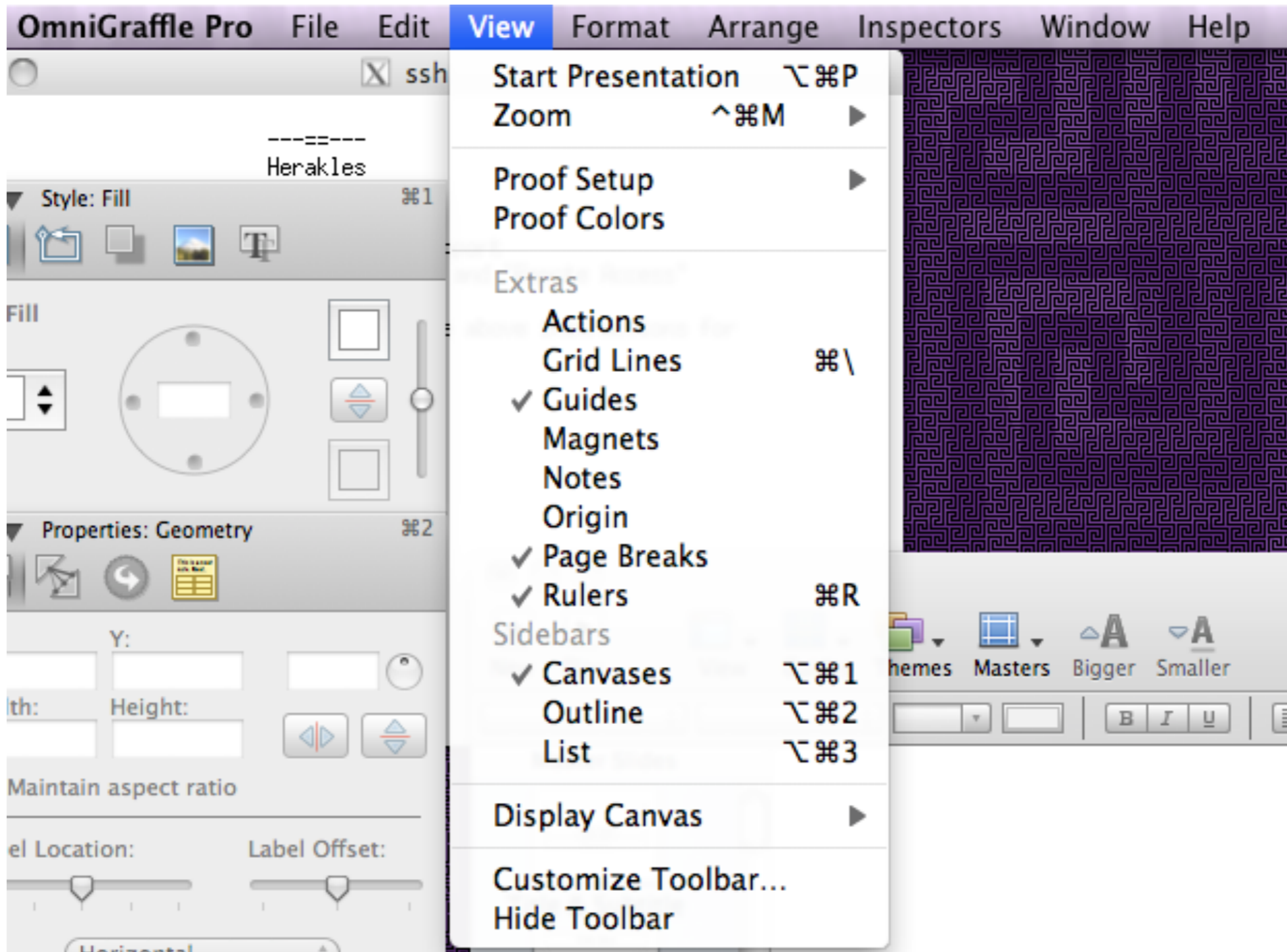


Novice users

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



Discoverability



No loaded guns



Frequent/power users

- Optimize for *efficient interaction*
- 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



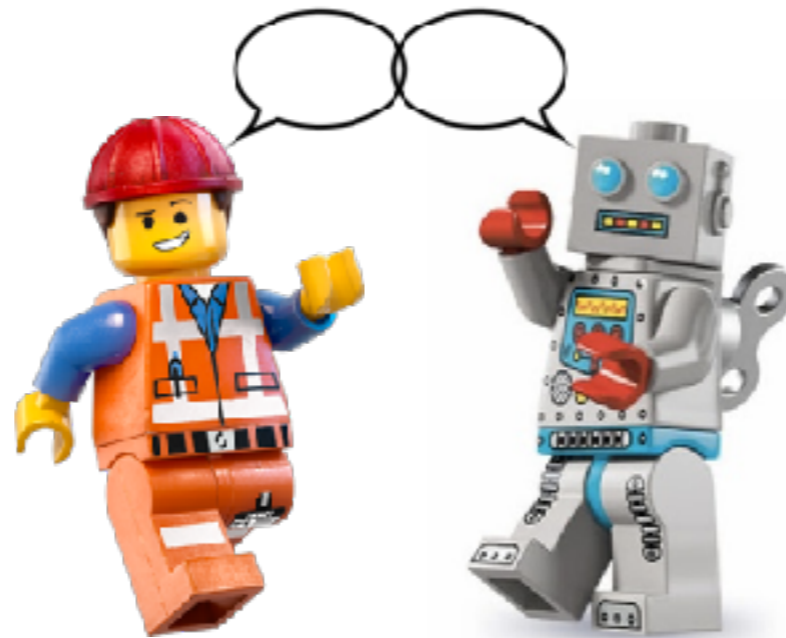
Expert UI



Know your user.



Principle 2: UI is a dialogue



UI: good conversation partner?

- Ratify actions quickly
- Be responsive (e.g., highlighting affordances)
- Show progress on longer actions



Conversations

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



Interaction paradigms

- Direct manipulation: the UI *is* the underlying data/behavior model
 - User view: Model = View = Controller
 - Implementation: Model \neq View \neq Controller
- I/O: UI generates output when input provided (UI \neq model)
 - e.g., menus, submitted forms, command shells



Direct manipulation vs. I/O



Know your user. UI is a dialogue.



Interaction time scales

- 1/60s: biologically imperceptible: faster than neurons
- 1/30s: 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.
- 10s–1 min: User becomes distracted and productivity declines. App needs to support parallel activities.
- >1 min: Significant loss of productivity. User leaves for coffee.

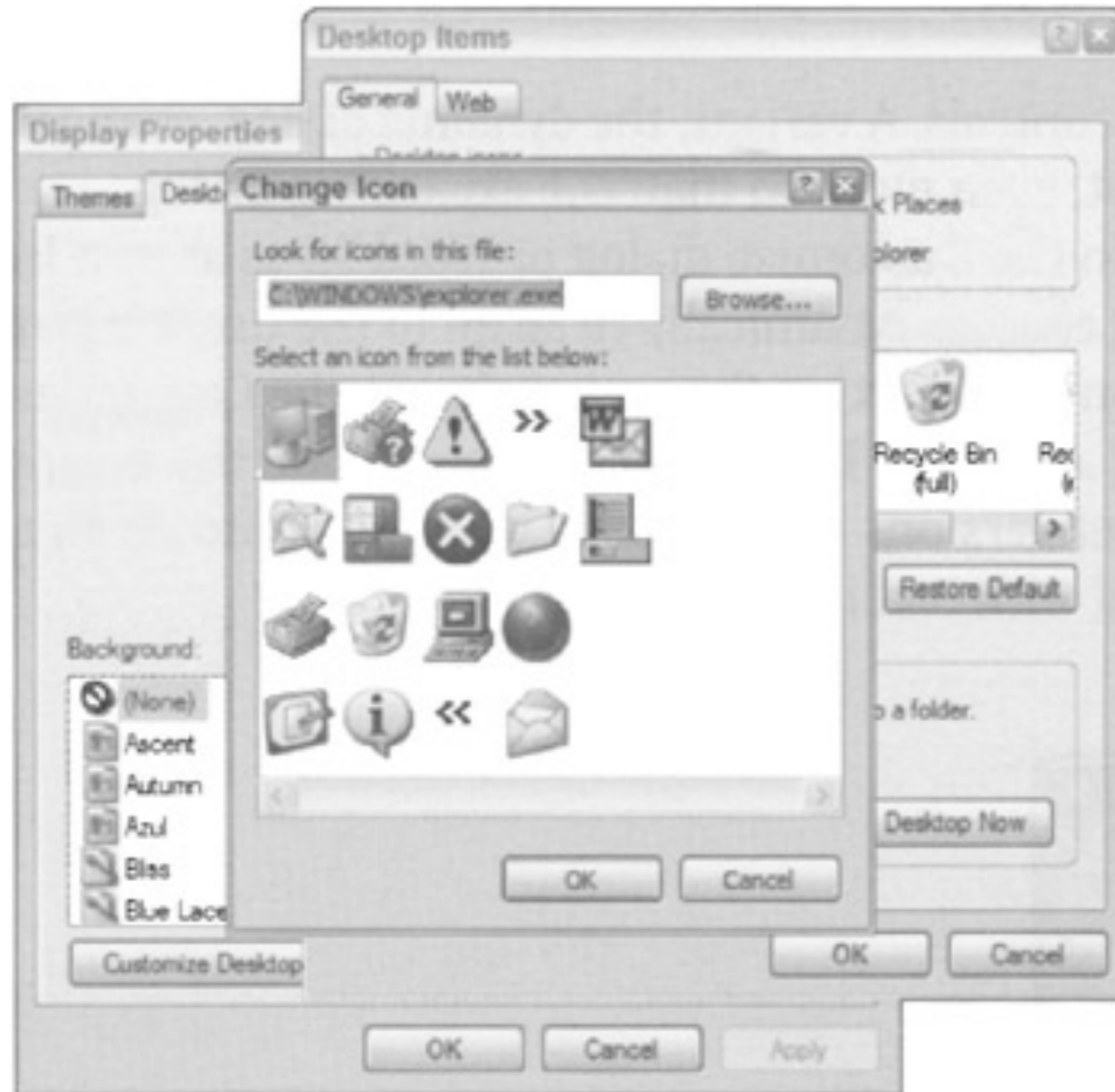


Modes

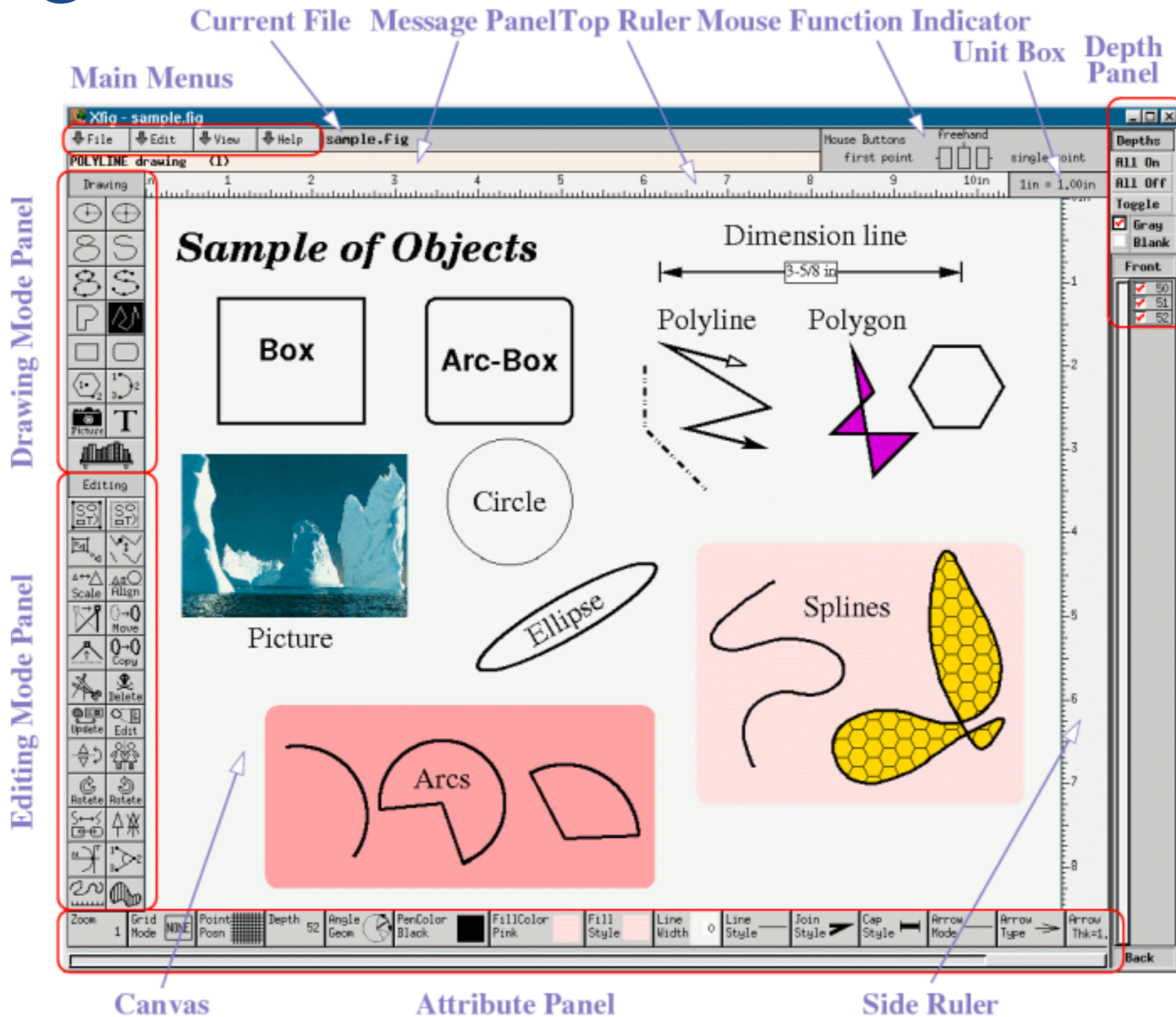
- Modes: states of UI that restrict interactions.
- Good: restricted context-sensitive vocabulary simplifies user interaction
- Bad: can be confusing and can trap users
- Moral: use judiciously



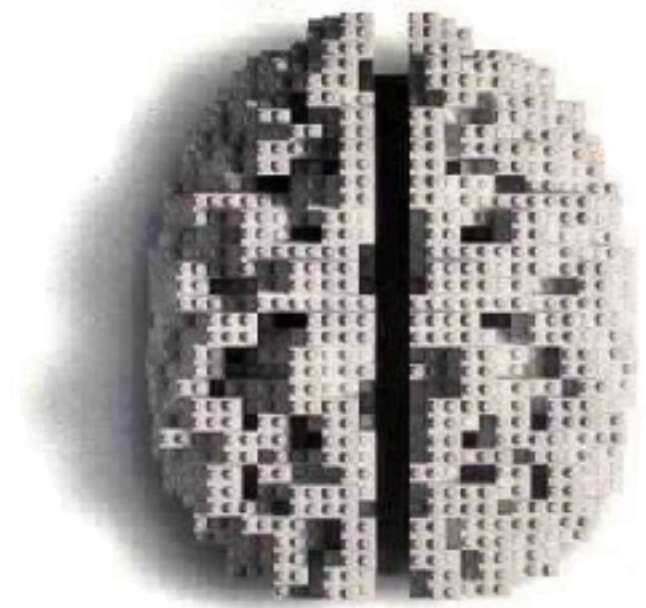
When modes go bad: cascading dialogs



xfig: the context-sensitive mouse



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

Rule of 7

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

⇒ Avoid long menus, arrays of buttons



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

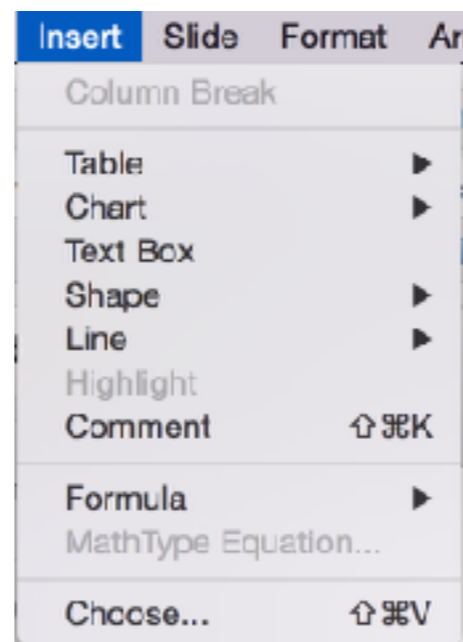


Muscle memory

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

⇒ action needs to activate functionality should be consistent

- e.g., gray out menu items instead of removing them

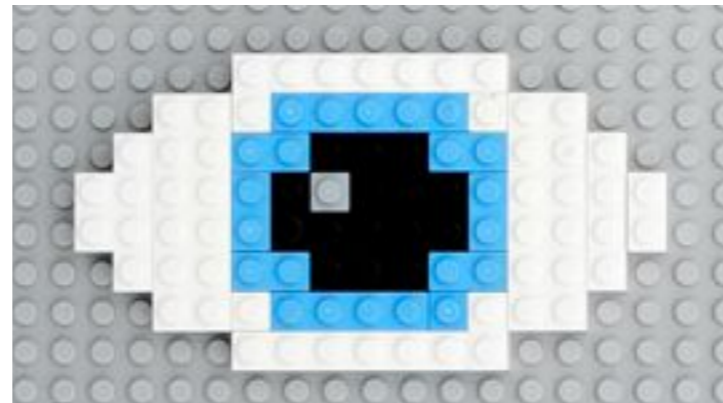


Context-sensitive help

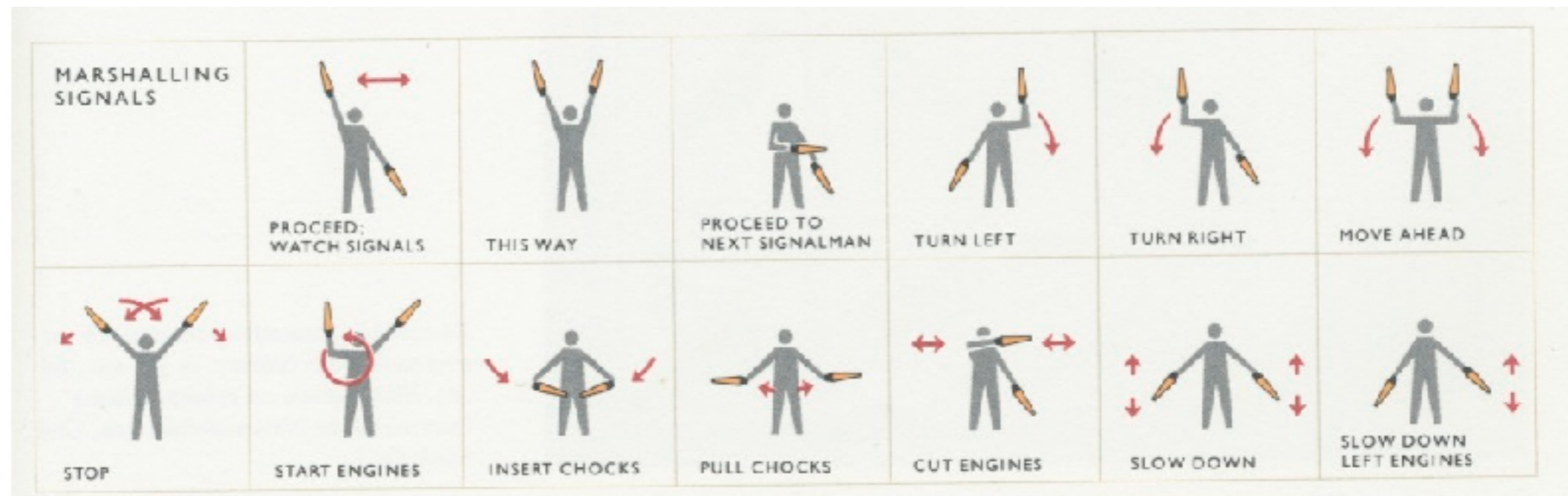
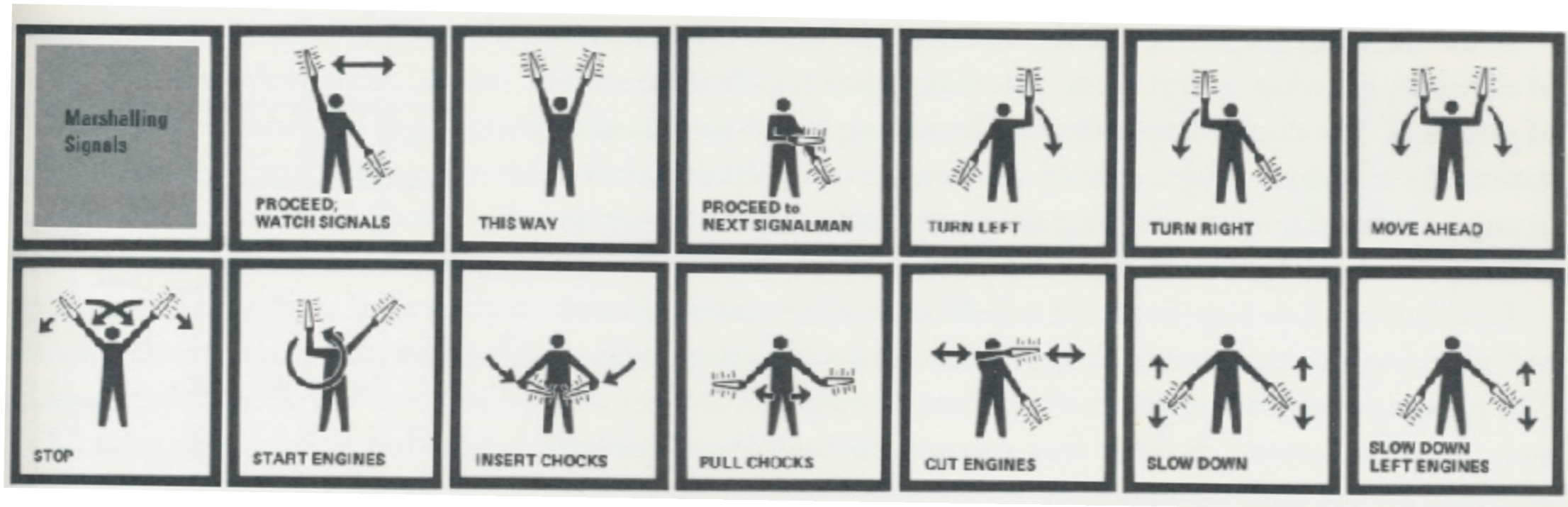
- Help should be about what user is doing now.
 - ⇒ task-focused rather than feature-focused (unlike most modern apps!)
 - ⇒ modes provide context



Principle 4: Visual design matters

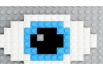


Avoid visual clutter

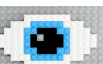
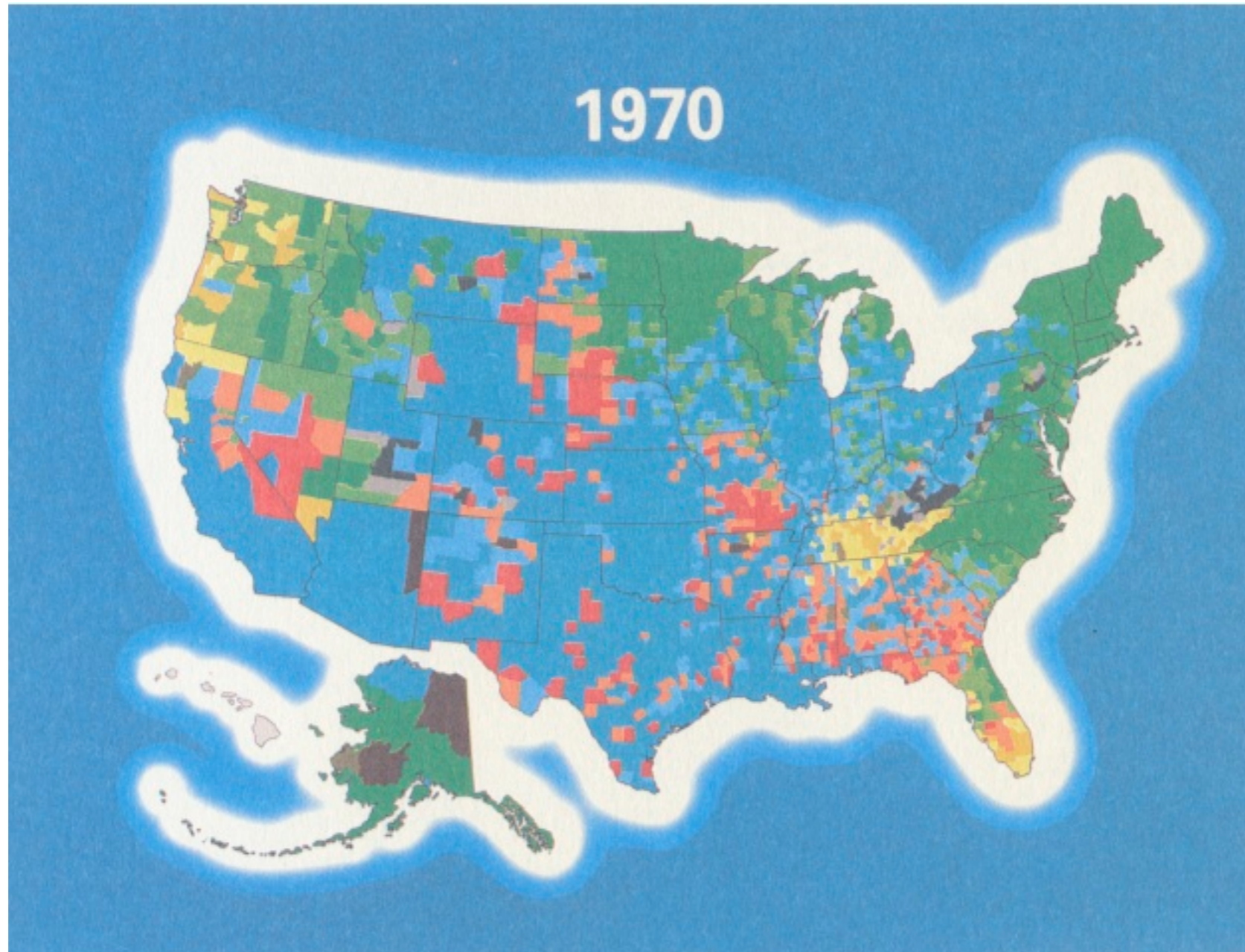


Avoid visual clutter

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



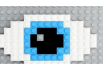
Good use of color and contrast?



Use high contrast, avoid chromatic aberration

**This text is probably
not very pleasant to
read.**

And it gets harder if the font size is small.



Visual consistency

- For novice users, be 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** for developers



Visual features

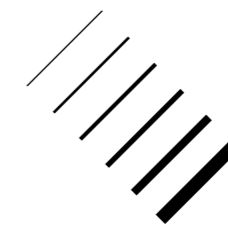
- Shape: up to 15



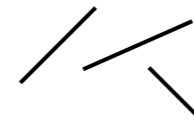
- Color: up to 24



- Size, length, thickness: up to 6



- Orientation: up to 24



- Texture

- Differing color perception!

⇒ can only *complement* other sources of information



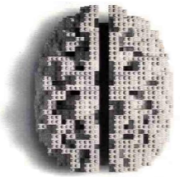
UI design principles



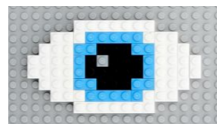
- Know your user



- UI is a dialogue



- Aid memory



- Visual design matters