

# Human-Computer Interaction

(HCI)

# Warm-up

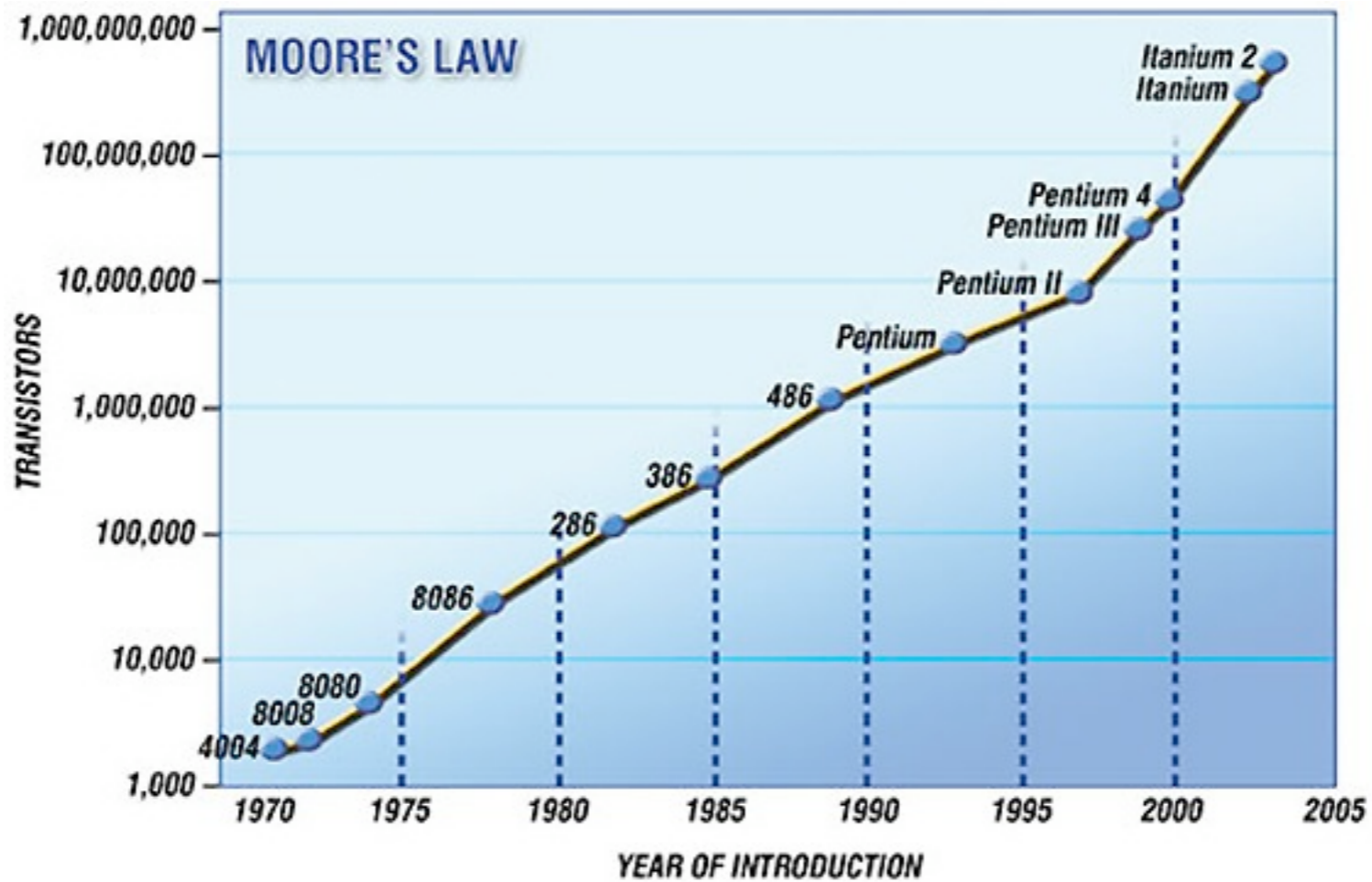
- On a piece of paper with your name you can turn in, answer these questions:
- Who do you think designs the look and feel of Windows or Photoshop? What do you imagine that process is like?
- What are some applications or websites you think are really easy/pleasant to use? Hard/unpleasant? What makes them that way?
- Do you think designing interfaces is hard or easy? Why?

“When the creators of software-based products examine their handiwork, they overlook how bad it is.

Unfortunately, their frame of reference is themselves, so they only make it easy to use for other software engineers, not for normal human beings.”

- Alan Cooper, *The Inmates are Running the Asylum*

# Moore's Law

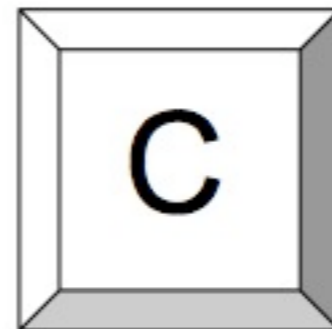
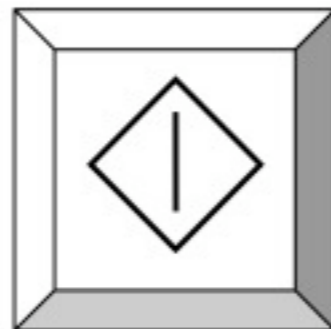


# Need for design

- Computers as cognitive prosthetics
- Computers are cheaper and faster but our abilities haven't changed
- Computing time is now much cheaper than human time
- Before, user had to bend to machine's will

# Poor design leads to...

- Frustration
- Wasted time
- Errors
- If you've never seen a copier, how do you copy?



# HCI is Inter-disciplinary

- Computer science
- Cognitive Psychology
- Social Psychology
- Ergonomics
- Linguistics
- Artificial intelligence
- Philosophy
- Sociology
- Anthropology
- Design
- ...

# Factors affecting design

<b>Organisation Factors</b> Training, job design, politics, roles, work organisation		<b>Environmental Factors</b> Noise, heating, lighting, ventilation
<b>Health and Safety Factors</b>	<b>The User</b> Cognitive processes and capabilities Motivation, enjoyment, satisfaction, personality, experience	<b>Comfort Factors</b> Seating, equipment, layout.
<b>User Interface</b> Input devices, output devices, dialogue structures, use of colour, icons, commands, navigation, graphics, natural language, user support, multimedia,		
<b>Task Factors</b> Easy, complex, novel, task allocation, monitoring, skills		
<b>Constraints</b> Cost, timescales, budgets, staff, equipment, buildings		
<b>System Functionality</b> Hardware, software, application		
<b>Productivity Factors</b> Increase output, increase quality, decrease costs, decrease errors, increase innovation		

# Usability

- Input format
- Feedback
- Visibility
- Perceived potential (is this door push or pull?)

# HCI Principles

- Put people first
- Understand how people use technology
- Develop tools to enable building good systems
- Design for efficient, effective and safe interaction
- Quality is contextual (fit to task)
- User reaction is the best test

# Steps in HCI

- Observation (what's the problem?)
- **Paper prototype**
- Video prototype
- User tests
- Iteration

# Paper Prototyping

- Test the design
- Choose between alternatives
- Minimize cost of experimentation



# Alarm Clock for world-traveling businesspeople

- Hardware or software? Phone?
- How do you know if the alarm is set?
- What colors do you use?
- How easy is it to set the alarm?
- How many buttons are there?
- How do you change the timezone?
- Differences with kids' alarm clock?

# Knowing your users

- Expertise
- Computing habits
- Location
- Social interactions
- Literacy
- Vision

# Understanding usability requirements

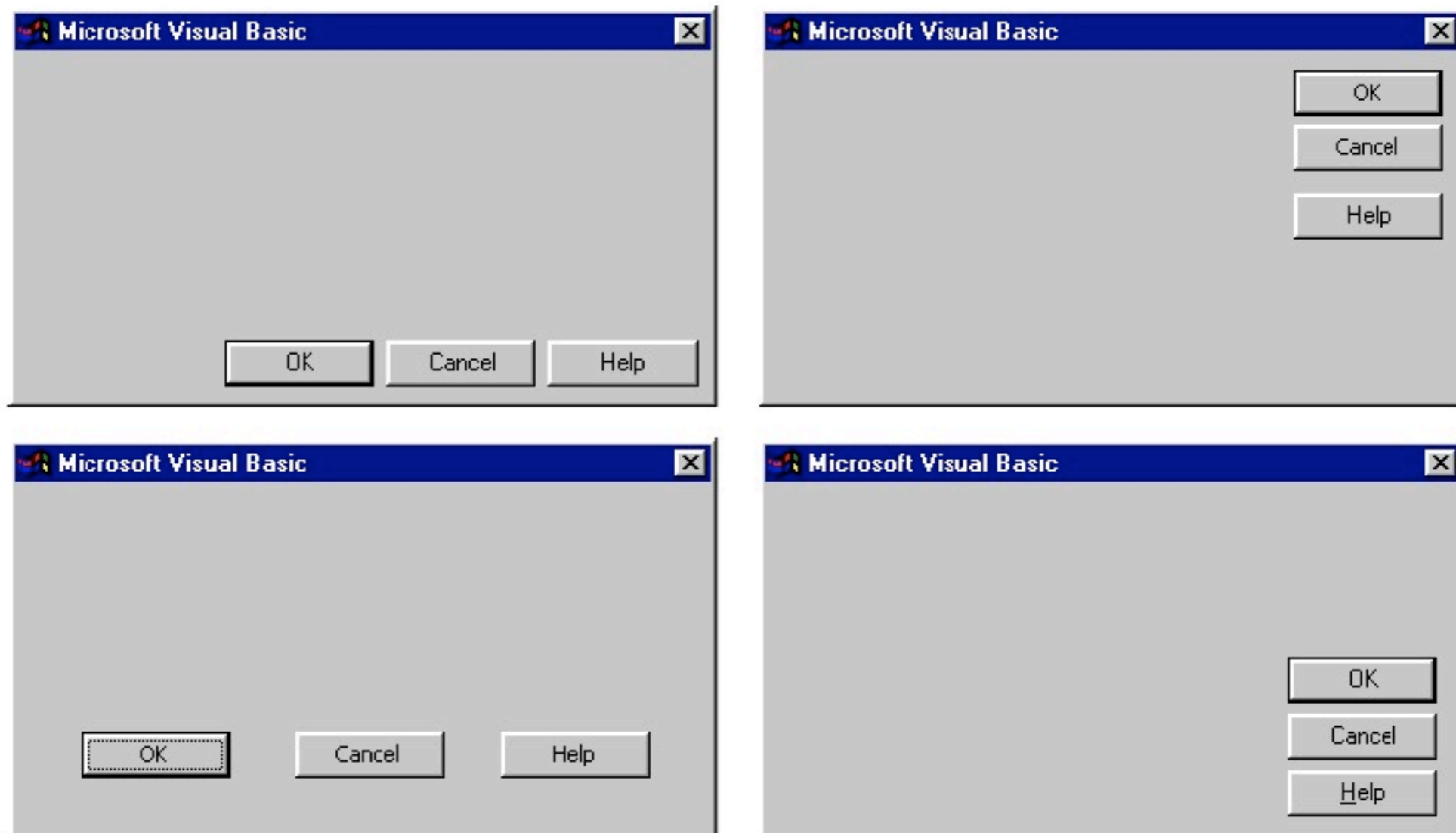
- Learnability
- Speed of performance
- Error rate
- Retention over time
- Subjective satisfaction
- ... often sacrifice one for the other!!

# Shneiderman's 8 Golden Rules

- Strive for consistency
- Enable frequent users to use shortcuts
- Offer informative feedback
- Design dialogs to yield closure
- Offer error prevention and simple error handling
- Permit easy reversal of actions
- Support internal locus of control
- Reduce short-term memory load

# Consistency

"consistency makes the interface familiar and predictable",  
The Windows User Interface Guidelines for Software Design, Microsoft Press



# Dialog closure

- Users should feel like task is complete for
  - satisfaction
  - relief
  - preparation for next actions

# Errors

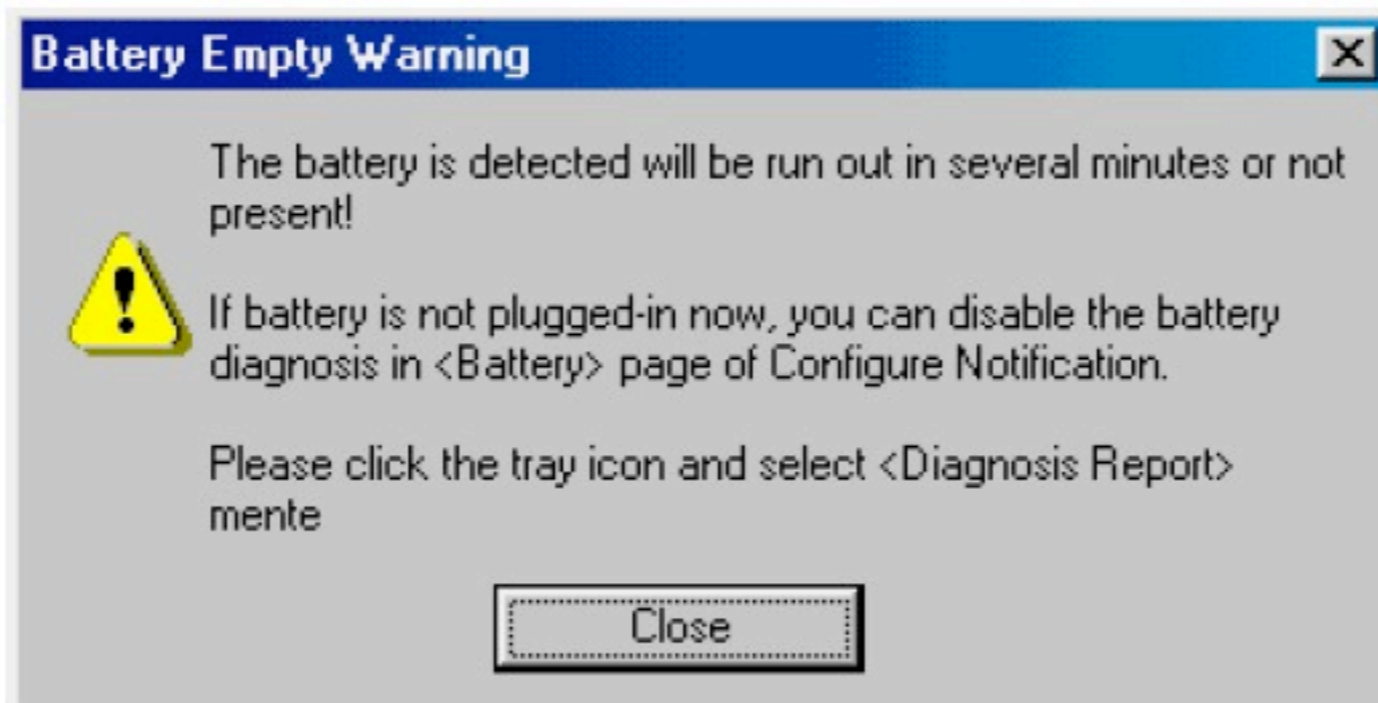
- What happened?

- W

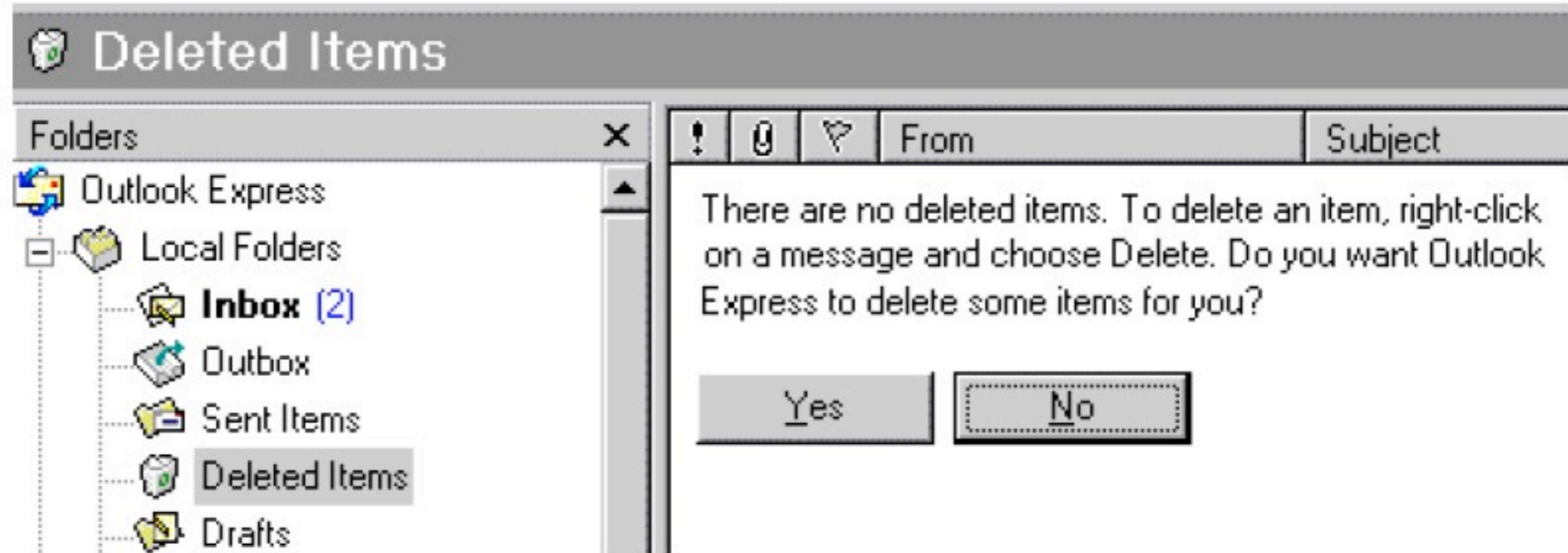
- Hc

- Hc

- Prevent by using menus instead of fill-ins, etc



# Locus of Control



# Memory Load

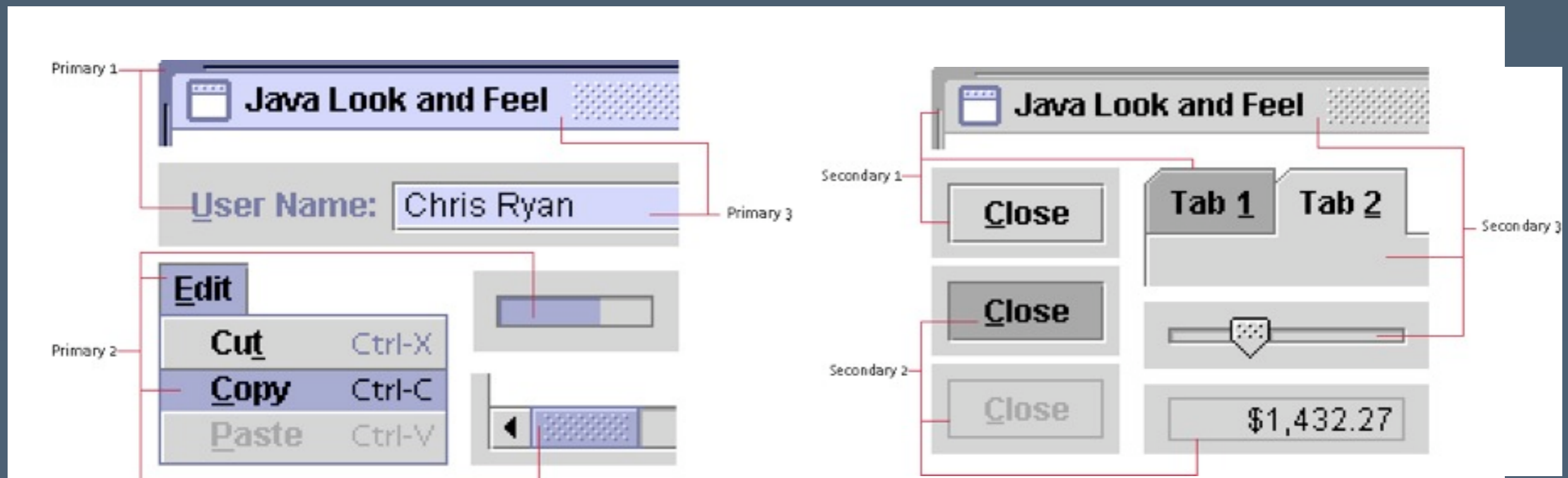
- Humans remember seven plus or minus two chunks
- Consolidate multiple page displays
- Don't make users note down codes
- Allow for training

# Desktop applications

- Must fit existing metaphors (windows, open, etc)
- Must integrate into operating system
- Must be consistent between screens
- Less and less relevant!

# Color (Java L&F)

- Six color semantic scheme
- Clean, consistent look
- Easy on eyes (mostly gray)



# Mobile

- Swiss army knife
- Multiple interaction models (speech recognition?)
- Integration
- Social
- Ubiquitous

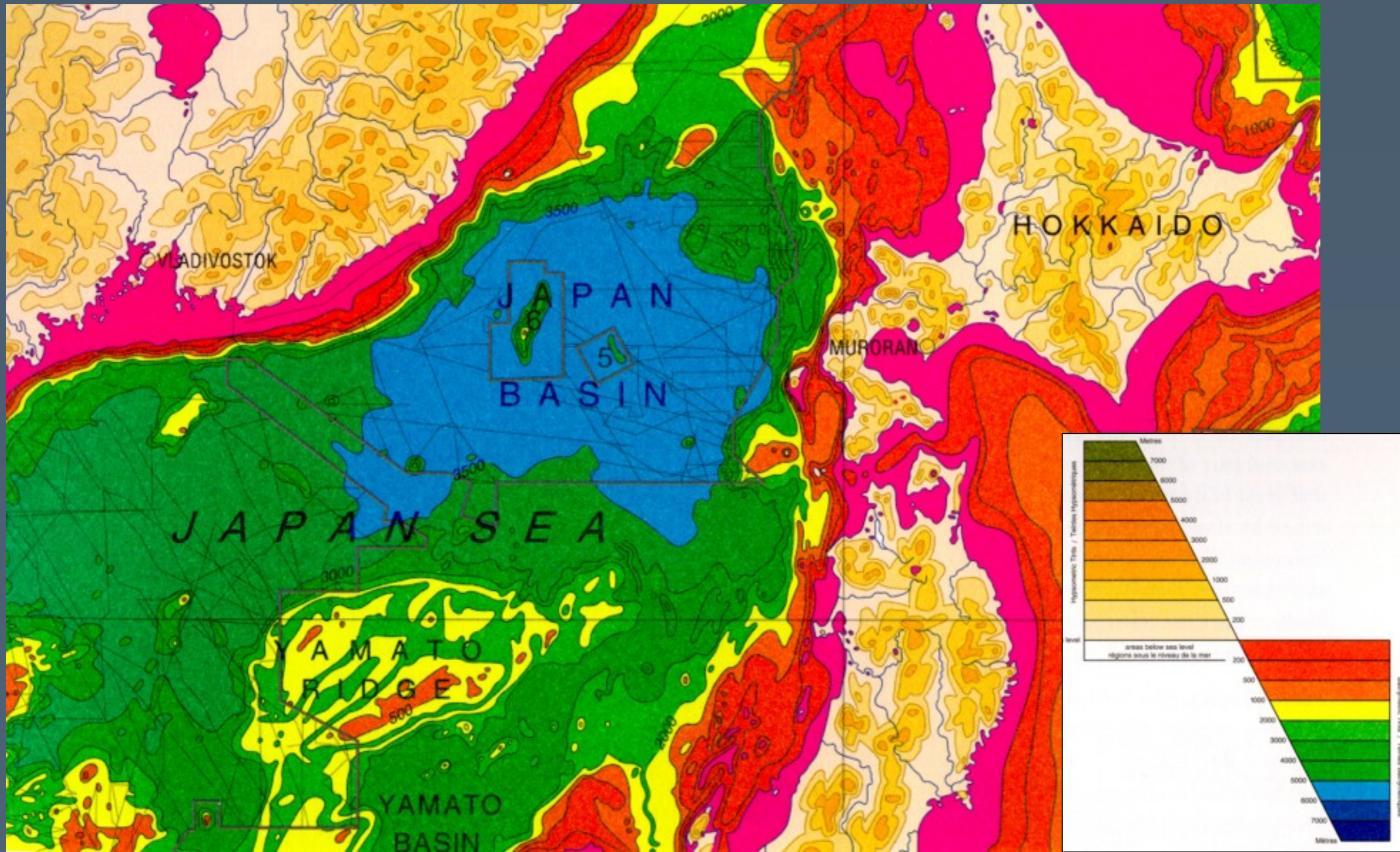
# Interaction styles

- Direct manipulation (gesture)
- Type in form
- Menu selection (touch vs keyboard)
- Command language
- Natural language
- Combination for different users

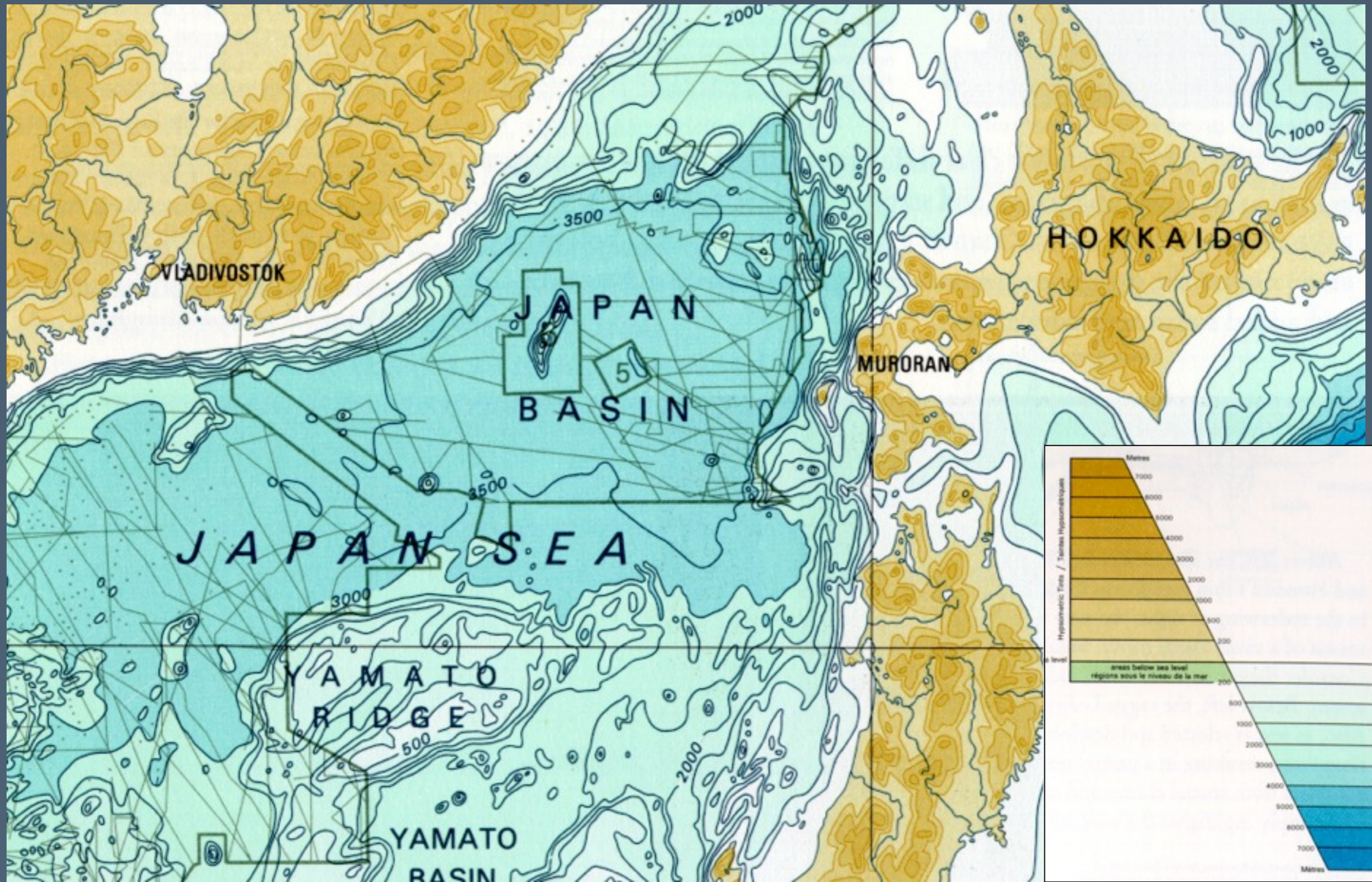
# Presenting information

- Color
- Fonts
- Size

# Color: Edward Tufte



# Color: Edward Tufte



# Marks of Typographic Style

What to do with CAPITALS in your text? SMALL CAPITALS are the best solution. If they aren't available, reduce the point size of the full-size CAPITALS slightly and letterspace so they aren't crashing into each other.  
capitals attract too much attention and break up the text

What to do with CAPITALS in your text? SMALL CAPITALS are the best solution. If they aren't available, reduce the point size of the full-size CAPITALS slightly and letterspace so they aren't crashing into each other.  
small caps are designed to work with lowercase letters

What to do with CAPITALS in your text? SMALL CAPITALS are the best solution. If they aren't available, reduce the point size of the full-size CAPITALS slightly and letterspace so they aren't crashing into each other.  
if you don't have small caps, set full-size caps smaller

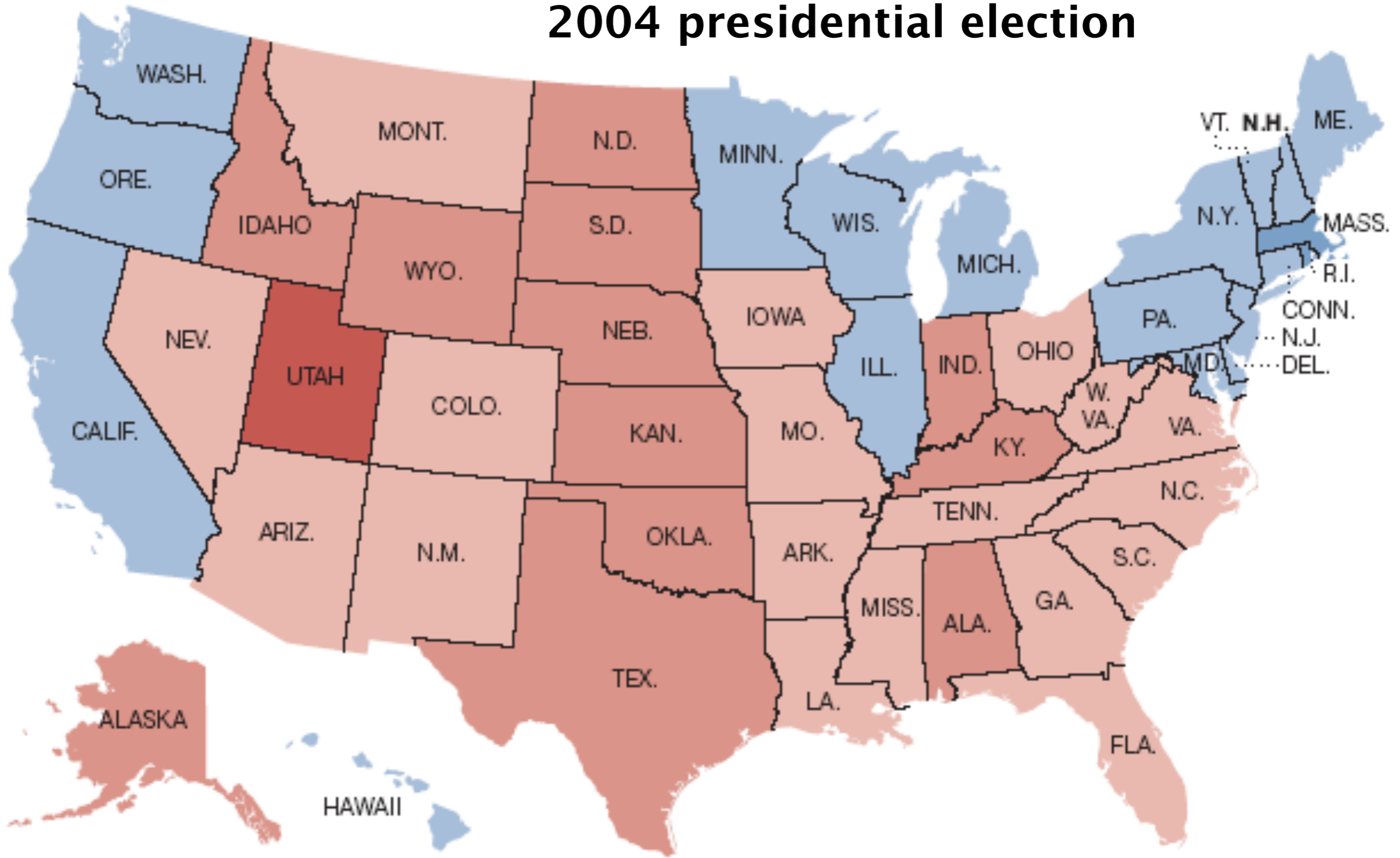
## Ligatures

office flirt file afflict effect  
office flirt file afflict effect

## Upper and lower case numbers

0123456789 0123456789

# 2004 presidential election

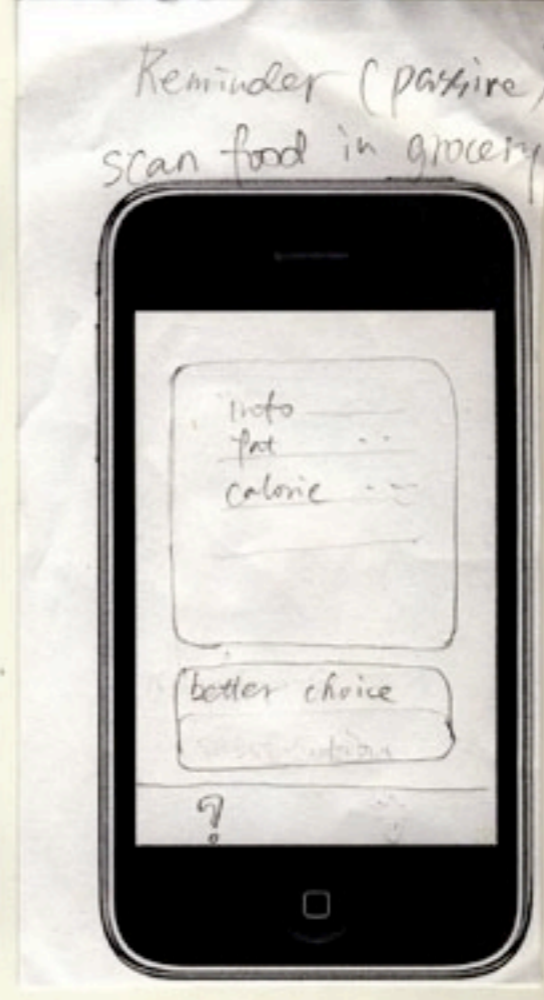
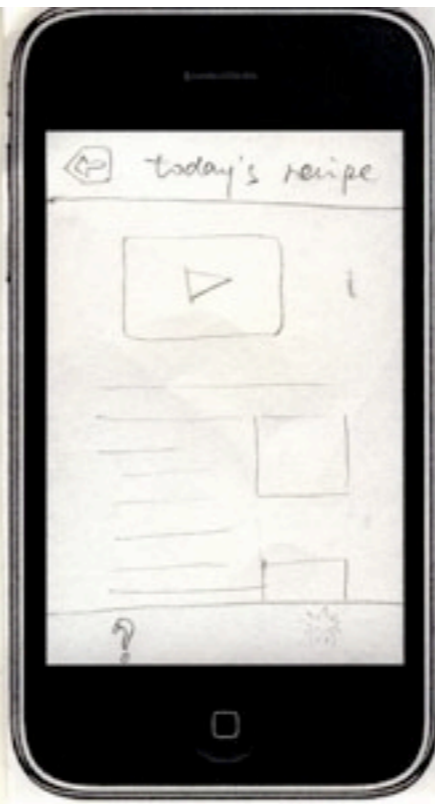


Matthew Ericson, NY Times



# Creating usable designs

- Most important: solve a real problem!
- Know your users
- Choose the right interaction model
- Try a lot of things early
- Get a lot of user feedback



# Problem statement

- You want to know where your friends are
- You want to show off how many places you go to
- Design a cellphone app that lets you
  - Tell your friends where you are (and ONLY your friends!)
  - Get points for going to new locations
  - See where your friends are
  - Discover new places and get tips about them from the community
  - Add friends