Principles of HCI Design

CS774 Human-Computer Interaction
Spring 2004

HCI Design Principles

- Design involves choices
  - Big choices - primary interaction styles
  - Little choices - button label, color, position
- HCI design is art, not science
  - No algorithm to make choices for you
- Principles help make choices
  - Heuristics -- rule of thumb, wise guess
  - etymology:from Greek heuriskein to discover, to find

Proliferation of Principles

- Shneiderman - 3 principles, 8 golden rules
- Norman - 7 principles
- Borenstein - 10 commandments
- Mandel - 3 golden rules
- Johnson (GUI Bloopers) - 9 principles, 82 bloopers *and* rules to prevent them
- How to learn what everyone has to say?

What have we learned so far?

- Knowledge in the world
  - Books, notes, bookmarks - organize the world
- Magic number 7
  - Keep the number in mind small
- Learn
  - Accommodate/assimilate/habituate
- Mnemonics
  - Use patterns, mental images, to organize your mind

Principles vs. Guidelines

- General Principles
  - Applicable to many interface systems
  - Big and small designs
  - Verified by research experience
- Widget Guidelines
  - Designed for specific GUI systems
  - Gives grammar and vocabulary for interface
  - Mostly for smaller decisions
  - Apple, Windows, Java guidelines

Process vs. Usability

- Process principles
  - Such as make prototypes, involve user, etc.
  - Focuses on what you do
- Usability principles
  - Such as provide feedback, be consistent, etc.
  - Focuses on what you create
  - Good for evaluation of interface

Chapter 2 Principles

- Principle 1: Recognize the diversity
  - Characterize users, tasks and interaction styles
  - Foreshadows rest of book - will get back to
- Principle 2: Use the Eight Golden Rules
  - Will do now
- Principle 3: Prevent Errors
  - Your homework

1. Strive for Consistency

- Make the elements of your interface consistent.
- What elements?
- Consistent with what?

Consistency

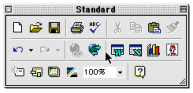
- With what?
  - Within your own program?
  - With other programs?
  - With the user's experience in the world?
- What elements?
  - Actions
  - Widgets
  - Data

2. Enable shortcuts

- Let the interface grow with the user
- How?

Shortcut options

- Keyboard short cuts
- Toolbars
- Re-configurable menus and toolbars



- Alternative representations

A counter opinion

- Jeff Raskin
  - One button mouse
  - Monotonous interfaces - only one way to do X
- Multiple options can result from
  - Legacy options
  - Management indecision (oh, include them all)
  - Myth of beginner/expert users
- Current GUIs mix of two inferior interfaces
  - Inefficient menu system + incomplete keyboard system

3. Offer informative feedback

- Let the user know what happened
- How?

Feedback

- Responsiveness
  - Do it quickly
  - If it's going to be a while, tell the user
  - Use watch cursors, progress bars
- When? What is a complete action to report on?
- How?
  - Change appearance of object (WYSIWYG)
  - Dialog boxes
  - Status of interface (grey out menu items, highlight, etc)

4. Design for closure

- Let the user know the task is done
- How?

Computers as Theatre

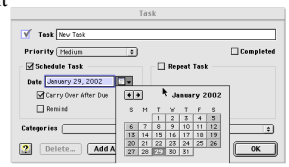
- Brenda Laurel - started as game designer
- Computer Screen as stage on which we enact scripts
- Think of interface in terms of
  - Narrative flow - beginning, middle, end
  - Actors/Agents - who saved your file? Who gave you the error message?
  - Stage with actors carrying out an action

5. Offer error prevention

- Design the system to prevent errors
- How?

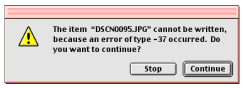
Avoid errors - control input

- Grey out inappropriate commands
- Forcing functions
- Control field input
  - No letters in number field
- Know causes/ frequency



Responding to errors

- Error messages should be in user's language
- Actions should be clear
- Avoid guessing user intent
  - PL/I and parens
- Do nothing (???)



6. Permit easy reversal

- Let the user undo actions
- How and why?

Easy undos

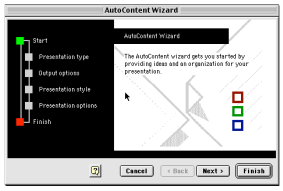
- Allow the user to feel safe
  - Less risk of damaging data
  - (Why is data ever damaged? Why not retain everything? Problem of the Mac trash can.)
- Correct errors
- Allow experimentation
- How
  - Revert to saved, undo (how long? What is a single action?), automatic backup files

7. Support internal locus of control

- Keep the user in control
- How?

User in control

- Avoid modal dialogs (with Java threads!)
- Avoid long guided sequences
- Be permissive
- Provide exits
  - Cancel, undo, interrupt, quit



8. Reduce short-term memory load

- Magic number 7+/-2
- How do we manage this?

Reducing memory load

- Rely on recognition, not memory (see and point)
- Provide cues (affordances, toolbars, menus)
- Visual clarity (Greek temple, not strip mall)
- Progressive disclosure (hide advanced stuff)