(human factors 2)

designing user interfaces

comp314 • 27 march 2007 • daniel sandler
(human factors 2)
Last time
Witnesses:

S. Sickling
T. B. Story

Latham Sholes,
By James Densmore,
Attorney.
MEMEX

NLS, MOUSE

SMALLTALK, DYNABOOK, GUI

QUICKDRAW, MACINTOSH

Vannevar Bush

Doug Engelbart

Alan Kay

Bill Atkinson
Human factors

- How **people** interact with **tools**
  - the Human interface: senses, muscles

Goal

- Make better **tools**
  - ↓ fatigue, confusion, errors
  - ↑ accuracy, performance, fun
Human-computer interface (HCI)

- Human factors as applied to computing systems
HCI ≠ aesthetics

HCI ⊆ aesthetics

- Not a “layer of paint” at the end!
- User experience is factored into successful computer systems from the beginning
- Pretty is nice, but usable is paramount
Motivation

Intrinsic  "natural incentive"

- An essential part of successful systems
- Develop your sensitivity, appreciation
- Add to your skill set

Extrinsic  "artificial incentive"

- Part of your Project 3 grade will depend on your user interface design
- ZOMG!
user interface design is hard mysterious boring for beret-wearing latté-drinkers not my problem
Today’s lecture:

DESIGNING USER INTERFACES
(YOU CAN DO IT)
Agenda

Principles of user interfaces
  ✤ examples from real apps

Design methodology
  ✤ tools you can use in project 3
Inescapable facts of life

- Always applicable to your project

Principles are factories for patterns

- Previous solutions that work pretty well
- Sometimes applicable to your project

We’ll look at 4 principles

- Patterns that result
1. accessibility
accessibility

ability to **physically use** the interface
case study: colorblindness
case study: fitts’ law

Paul Fitts, 1954

Situation: user needs to point to a target
   - Original motivation: airport cockpit (!)

Difficulty (time to acquire)
   - \( D = \log (k + a \times w^{-1}) \)
     - proportional to **distance to target**; inv. prop. to **size of target**
     - Smaller, further targets harder to hit
Applications of Fitts

“mile high menus” — 1981

task bar — 2001

“mile high menus in office” — 2006
(assumption?)
2. simplicity
Obligatory Einstein quote

- “As simple as possible, but no simpler”

However

- “If ease of use was the only valid criterion, people would stick to tricycles and never try bicycles.”

Lesson

- Tradeoffs
case study: vista shutdown

plus:
physical power button
close laptop lid
Fn+<foo>

[Joel Spolsky, Choices = headaches, 2006]
Case study: frequency response measurement
pattern: progressive disclosure

Examples

❖ tree view
❖ tabs
❖ expansion & collapse

Let the user choose the level of detail

❖ (but remember, choices = headaches)
3. consistency
The simplest principle of all!

When in doubt,

- Do what **others** have done
- Do what **you** have already done

Beware

- “A foolish consistency is the hobgoblin of little minds.” —Emerson
case study: iTunes 7
The “principle of least astonishment”

- It should do what the user guesses it will do
- aka: “least surprise”
- This helps explain why consistency is so important: users are always guessing
- “mental models”
4. metaphor
metaphors: how we understand the world

Essential conceptual aids

- Connect mental models (what you understand) to external models (what is going on)

ARGUMENT = WAR

- attacking and defending positions; mustering arguments; shooting down ideas; winning and losing

TIME = MONEY

- quit wasting it; budget your time better; spend it wisely

See: George Lakoff & Mark Johnson, *Metaphors we Live By*, 1980
UI metaphors work the same way

Connect user’s mental model to system's functional model

❖ (Adapter pattern?)
UI metaphors?

FILE SYSTEM?
- OFFICE / DESK
  - Files; folders; trash; the Notepad

HTML/HTTP?
- DOCUMENT
  - Web page
- WEB
  - Links between pages
  - Crawlers/spiders
    - (vestiges of “spiderweb” sub-metaphor)
case study:
the web browser
What’s in a browser?

off the top of your head; no cheating

this is what we call a “wireframe”
History
design exercise
UI design lifecycle

1. Requirements
2. Design
3. Prototype
4. Test
5. Loop to earlier steps as necessary

(it will be necessary)
exercise: create a wireframe

Project: screen saver called “Message”

- draw a string many times
- in random positions on screen
- in one font
- at random sizes
- in various shades (luminosities) of one color

Your job: design the config interface
Message parameters

- string can be fixed text, e.g. “Hello”
- string can be output of a shell program, e.g. “fortune”
- delay between paintings can be adjusted between 10 sec and 1 hr
- color saturation can be adjusted from “gray” to “color”
- max luminosity can be adjusted from “dark” to “light”
- background can be solid or 2x2 or 4x4 pixel checkers

Task: In groups, create a wireframe for the configuration window for this screen saver; one of you will copy it to the whiteboard for discussion.
3/27/07: “Doodle faster!”
3/27/07: Finished designs
(The students had only about 15 minutes to create and whiteboard their wireframes.)

For audio of the design review, visit
https://sys.cs.rice.edu/course/comp314/07/lectures/23-ui-design/
Parting notes

I’ve only scratched the surface
  (of the surface)

Further reading
  - Check online lecture notes (forthcoming)
  - Project 3
  - You will need to specify your UI like any other part of your program, and defend your decisions
  - Usability will count for the final grade!
(fin)