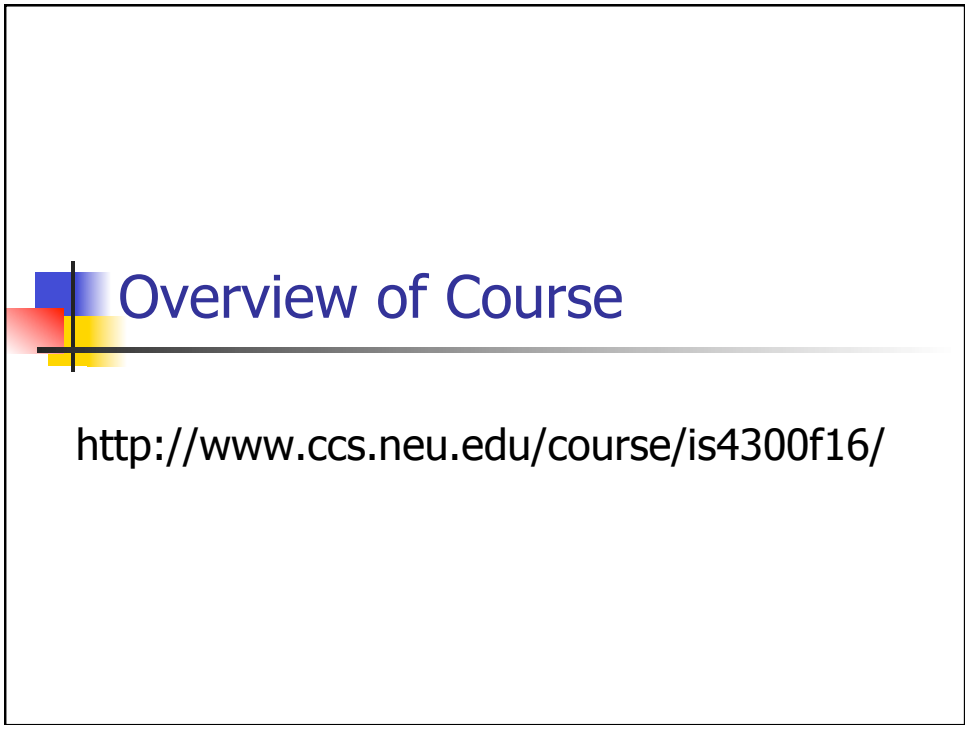
Human-Computer Interaction
IS 4300

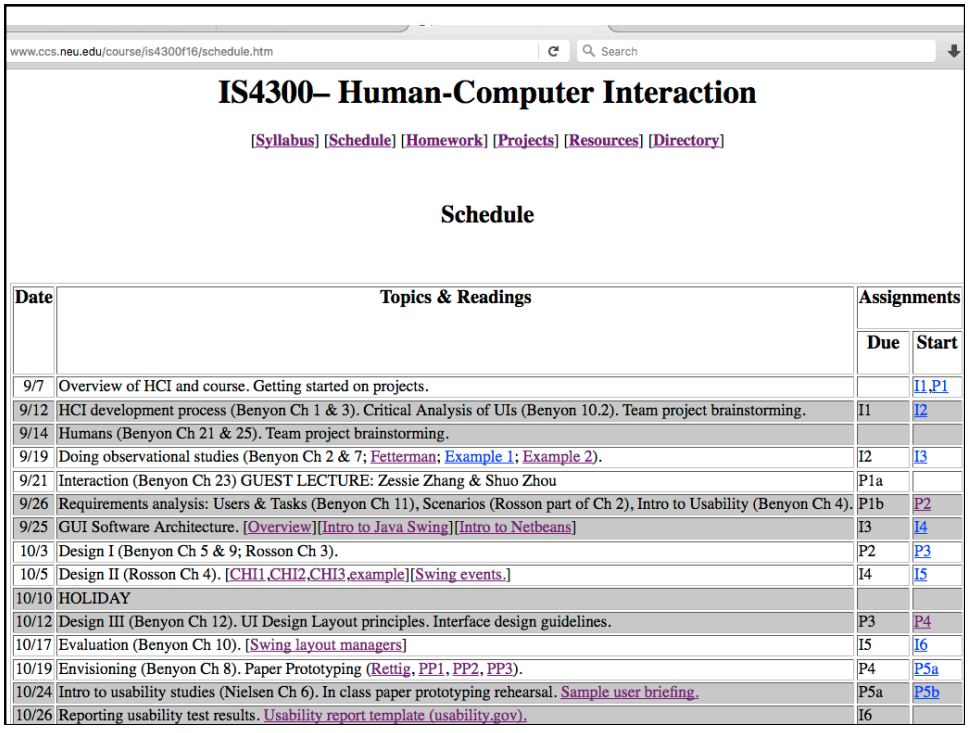
Overview for Today

- Brief review
- Readings
  - Designing Interactive Systems
  - HCI Development Process
- Critical Analysis of UIs
  - Heuristic Evaluation
  - Cognitive Walk-through Evaluation
- Homework I2
- Begin project brainstorming



Overview of Course

http://www.ccs.neu.edu/course/is4300f16/



www.ccs.neu.edu/course/is4300f16/schedule.htm

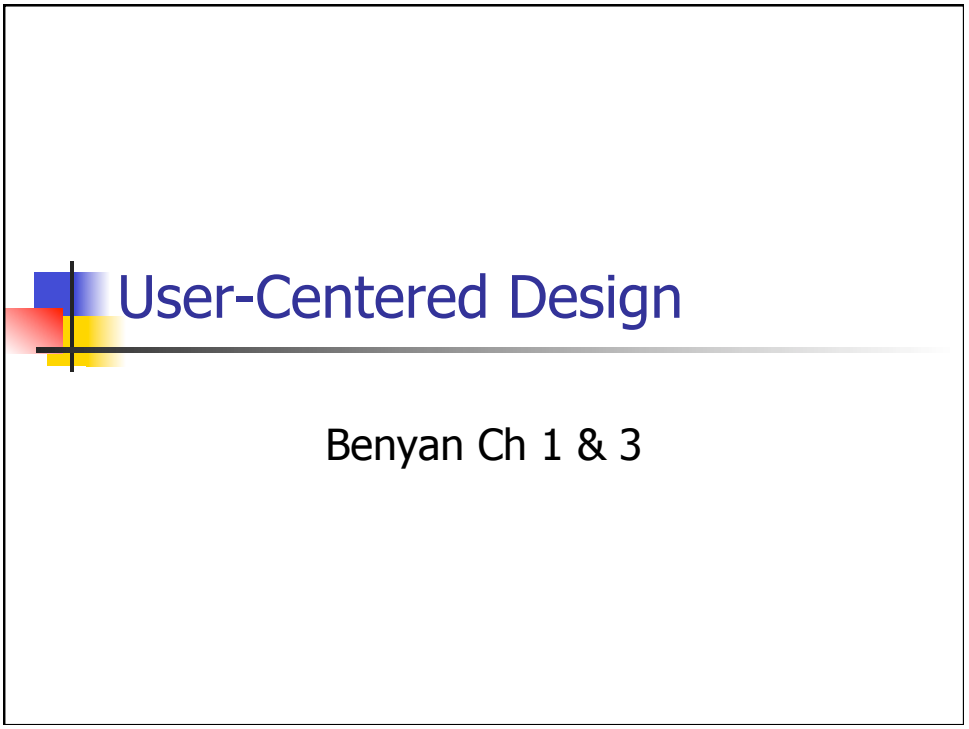
Search

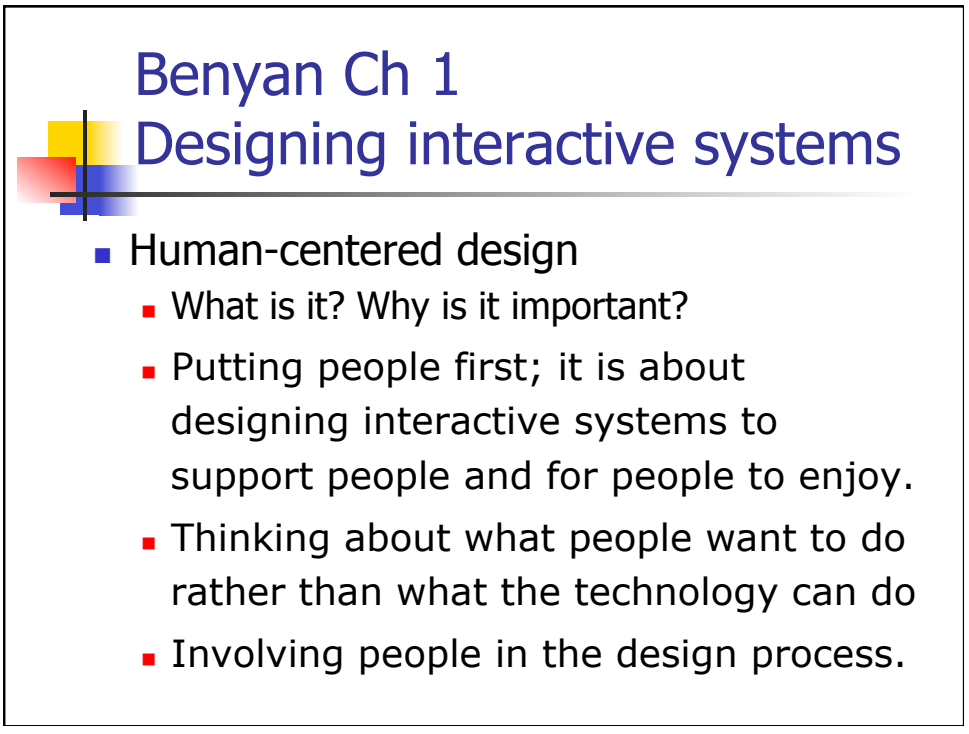
IS4300– Human-Computer Interaction

[Syllabus] [Schedule] [Homework] [Projects] [Resources] [Directory]

Schedule

Date	Topics & Readings	Assignments	
		Due	Start
9/7	Overview of HCI and course. Getting started on projects.		I1,P1
9/12	HCI development process (Benyon Ch 1 & 3). Critical Analysis of UIs (Benyon 10.2). Team project brainstorming.	I1	I2
9/14	Humans (Benyon Ch 21 & 25). Team project brainstorming.		
9/19	Doing observational studies (Benyon Ch 2 & 7; Fetterman; Example 1; Example 2).	I2	I3
9/21	Interaction (Benyon Ch 23) GUEST LECTURE: Zessie Zhang & Shuo Zhou	P1a	
9/26	Requirements analysis: Users & Tasks (Benyon Ch 11), Scenarios (Rosson part of Ch 2), Intro to Usability (Benyon Ch 4).	P1b	P2
9/25	GUI Software Architecture. [Overview][Intro to Java Swing][Intro to Netbeans]	I3	I4
10/3	Design I (Benyon Ch 5 & 9; Rosson Ch 3).	P2	P3
10/5	Design II (Rosson Ch 4). [CHI1,CHI2,CHI3,example][Swing events.]	I4	I5
10/10	HOLIDAY		
10/12	Design III (Benyon Ch 12). UI Design Layout principles. Interface design guidelines.	P3	P4
10/17	Evaluation (Benyon Ch 10). [Swing layout managers]	I5	I6
10/19	Envisioning (Benyon Ch 8). Paper Prototyping (Rettig, PP1, PP2, PP3).	P4	P5a
10/24	Intro to usability studies (Nielsen Ch 6). In class paper prototyping rehearsal. Sample user briefing.	P5a	P5b
10/26	Reporting usability test results. Usability report template (usability.gov).	I6	

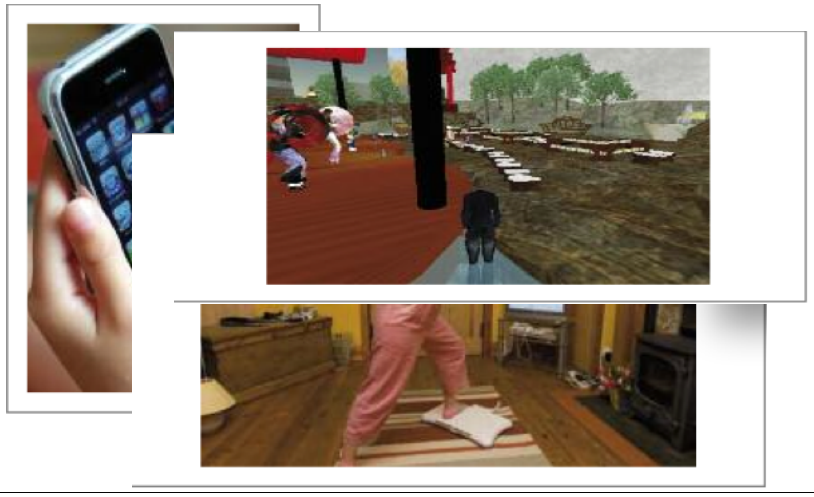

User-Centered Design
Benyan Ch 1 & 3


Benyan Ch 1
Designing interactive systems
Human-centered design
What is it? Why is it important?
Putting people first; it is about designing interactive systems to support people and for people to enjoy.
Thinking about what people want to do rather than what the technology can do
Involving people in the design process.

Benyan Ch 1
Designing interactive systems

- What kinds of products and contexts can you use human-centered design for?
  - Software for desktop
  - Websites, games, MP3 players
  - Environments in which many devices interact
  - Home, Work, Community

Benyan Ch 1
Designing interactive systems



Benyan Ch 1
Designing interactive systems

- What are some of the key concerns in human-centered design?
  - Design
  - Technologies
  - People
  - Activities and contexts

Benyan Ch 1
Designing interactive systems

- *Design*
  - both problem and solution need to evolve during the design process

Engineering Design
- Science rules

Artistic Design
- Creativity rules

Most projects have some aspects of each.

Benyan Ch 1
Designing interactive systems

- What disciplines inform what designers need to know about people?
  - **Psychology** is the study of how people think, feel and act.
  - **Ergonomics** is the study of fit between people and machines.
  - **Anthropology** is similar but focuses also on the study of culture, biology and language and on how these have evolved and changed over time.
  - **Sociology** is the study of the relationships between people in society, the social, political and other groups that they participate in, and the settings in which such relationships take place.

Benyan Ch 1
Designing interactive systems

- Other design concerns
  - Return on Investment
  - Safety
  - Ethics
  - Sustainability

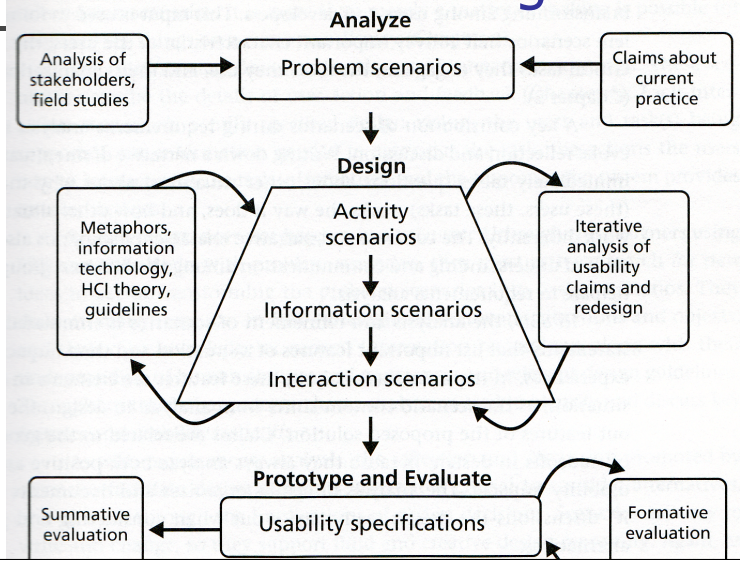
Benyan Ch 3
The process of … design

- Understanding
  - Requirements
  - Stakeholders
- Design
  - Conceptual
  - Physical
  - Envisionment
- Evaluation

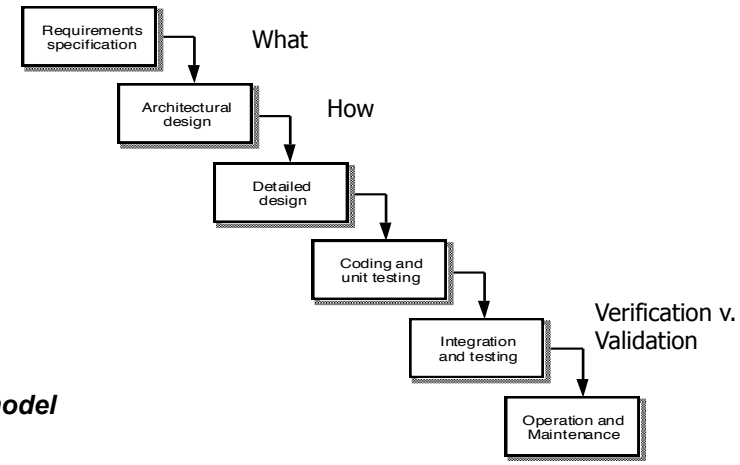
Benyan Ch 3
The process of … design

- Personas (?)
- Scenarios (?)
  - Stories - real-world experiences of people.
  - Conceptual - abstract descriptions
  - Concrete - specific design decisions and technologies added
  - Use cases - formal descriptions that can be implemented

Rosson & Caroll Scenario-Based Design

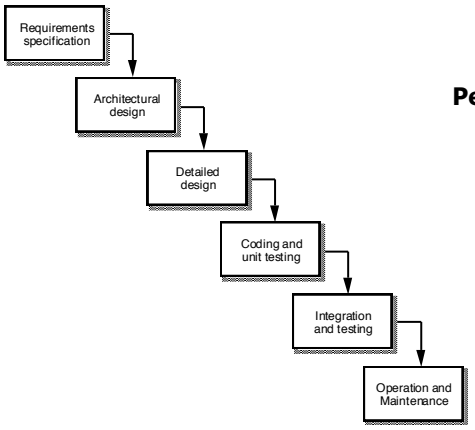


The Software Lifecycle



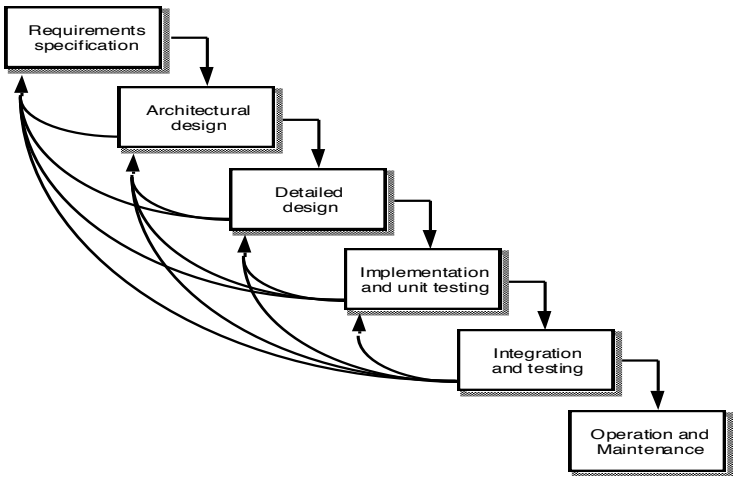
The waterfall model

Why doesn't this work for UIs?



People are insanely complicated.

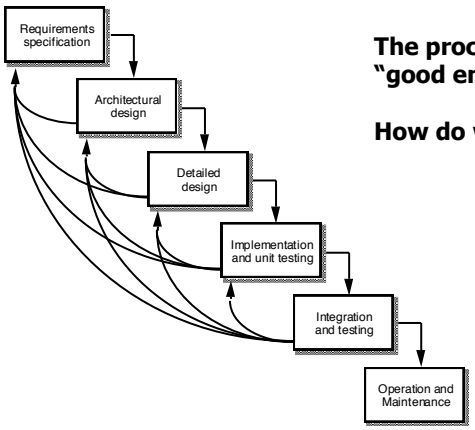
Lifecycle for UIs



User-Centered Design

- Try lots of stuff. See how it plays with the users.
  - Involve representative users in all stages of the development process.
  - Minimize the cost of and commitment to prototypes.
  - Users often can't tell you which alternative is "better" – have to test and measure.

Usability Engineering



The process by which we achieve "good enough" usability.

How do we know when we're there?

Usability Engineering

- Must define usability attributes (multi-dimensional)
- Must define specific measures for each
- Must define "good enough" (goal) levels for each
  - If appropriate, current & ideal levels for each
- Example attributes (measures?)
  - Learnability
  - Efficiency
  - Memorability
  - Low error rate
  - Subjectively pleasing

Critical Analysis of UIs

How do we know if a UI is "good"?

Evaluation Methodologies

- Expert
  - inspection methods
    - Heuristic evaluation
    - Cognitive walk through
  - models
- User testing
  - qualitative methods (observation, interviews, questionnaires, think aloud)
  - quantitative usability evaluation

Design Heuristics

- "Rules of Thumb" for improving usability
- Benyan: "Design principles" (§4.5)

Heuristics / Design Principles

- There are many "checklists" available
  - Nielsen's 10 design heuristics
  - Tognazzi's First Principles of Interaction Design
  - Gerhardt-Powals' cognitive engineering principles
  - Benyan's
  - etc

Nielsen's Heuristics
1. Feedback

- Clearly tell user effects of input actions
- Keep user informed of system state
  - Cursor change
  - Selection highlight
  - Status bar
- Feedback re: processing - response time
  - < 0.1 s: seems instantaneous
  - 0.1-1 s: user notices, but no feedback needed
  - 1-10 s: display busy cursor or other feedback
  - > 10 s: display progress bar

Feedback

Feedback

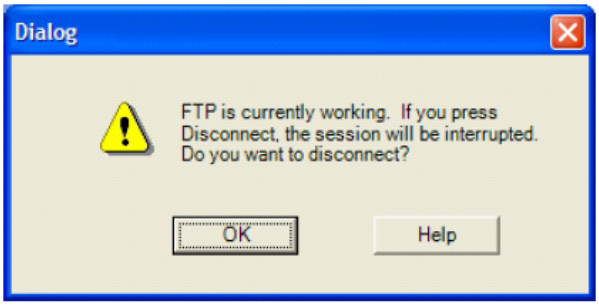
- Air France Flight 447, 1 June 2009, Airbus A330-200
- Stalled, crashed, killed 216 passengers and 12 aircrew
- Final report:
  - Initial cause: icing of airspeed sensors
  - Many feedback problems:
    - Inconsistency between the airspeed measurements
    - Incomprehension of the situation when the autopilot disconnection occurred,
    - The lack of a clear display in the cockpit of the airspeed inconsistencies identified by the computers
    - A failure to identify the aural stall warning
    - The appearance at the beginning of the event of transient warnings that could be considered as spurious
    - The absence of any visual information to confirm the approach-to-stall after the loss of the limit speeds

Nielsen's Heuristics
2. Speak the User's Language

- Use common words, not techie jargon
  - But use domain-specific terms where appropriate
- Don't put limits on user defined names
- Allow aliases/synonyms in command languages
- Use good metaphors
- Benyon: "Familiarity"

Nielsen's Heuristics
3. Clearly Marked Exits / Navigation / Freedom

- Provide undo
- Long operations should be cancelable
- All dialogs should have a cancel button



Nielsen's Heuristics
4. Consistency

- Principle of Least Surprise
  - Similar things should look and act similar
  - Different things should look different
- Other properties
  - Size, location, color, wording, ordering, …
- Follow platform standards
- Kinds of Consistency
  - Internal
  - External (aka "familiarity")
  - Metaphorical

Nielsen's Heuristics
5. Prevent Errors

- Selection is less error-prone than typing
- Disable illegal commands
- Description Error
  - different things/commands should look and act different
- Mode Error
  - Eliminate modes
  - Visibility of mode
  - Spring-loaded or temporary modes
- Benyon: "Constraints" & "Recovery"

Nielsen's Heuristics
6. Minimize User Memory Load

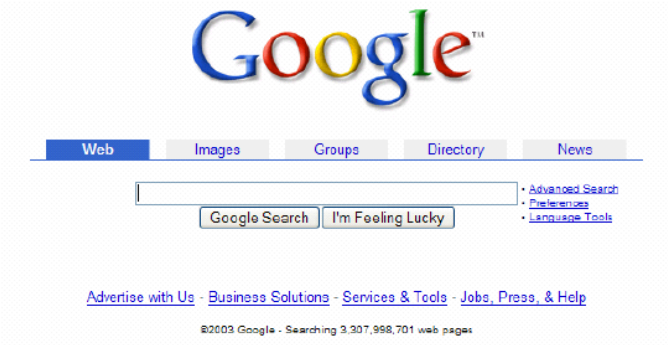
- Use menus, not command languages
- Use combo boxes, not textboxes
- Use generic commands where possible (Open, Save, Copy Paste)
- All needed information should be visible

Nielsen's Heuristics
7. Shortcuts / Flexibility / Accelerators

- Provide easily-learned shortcuts for frequent operations to improve efficiency
  - Keyboard accelerators
  - Command abbreviations
  - Styles
  - Bookmarks
  - History
- Benyon: "Flexibility"

Nielsen's Heuristics
8. Simple design

- "Less is More" / KISS
  - Omit extraneous info, graphics, features

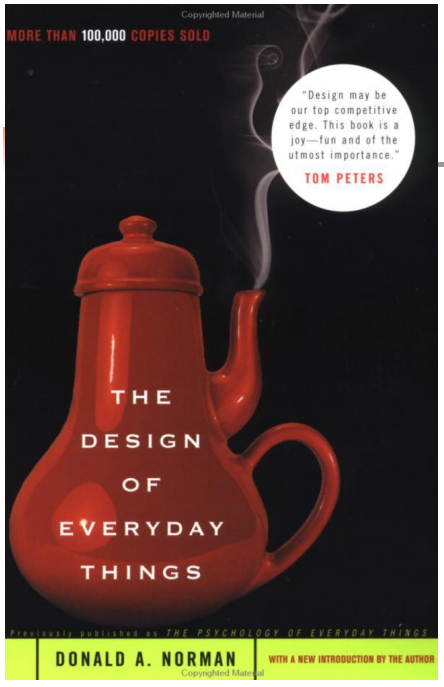


Nielsen's Heuristics
9. Good Error Messages

- Be precise; restate user's input
  - Not "Cannot open file", but "Cannot open file named paper.doc"
- Give constructive help
  - why error occurred and how to fix it
- Be polite and non-blaming
  - Not "fatal error", not "illegal"
- Hide technical details (stack trace) until requested

Nielsen's Heuristics
10. Help and Documentation

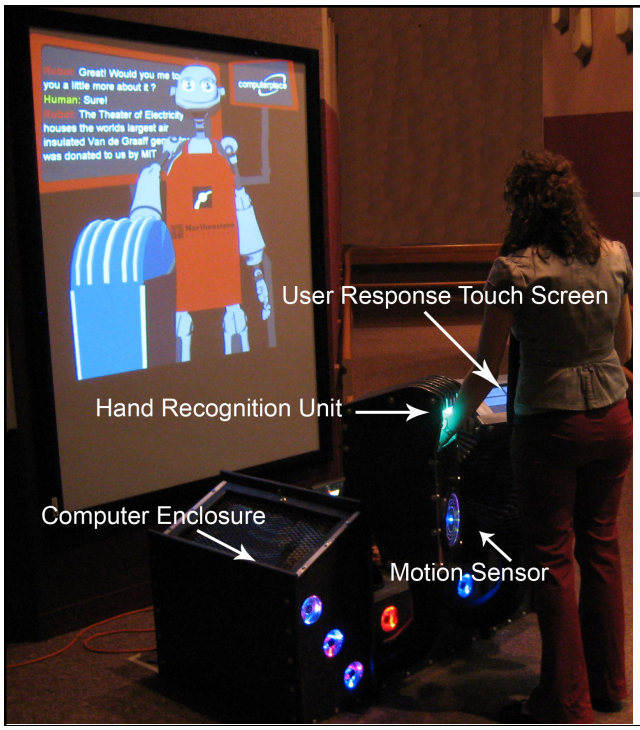
- Model
  1. Searching
  2. Understanding
  3. Applying
- Important features
  - Index
  - Overview map
  - Help visible while user is applying
  - Describe confirmatory feedback



A few additional rules/principles…

Affordances

- The fundamental properties of a thing that determine just how it could possibly be used.
  - Examples?
    - A chair affords sitting
    - Knobs are for turning.
    - Slots are for inserting things into.

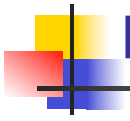


- How to get visitors to put their hand in the box?

Visibility

- *aka "Obviousness"*
- The correct parts must be visible.
- They must convey the correct message.
- Impacts learnability.

- How different from affordance?
- Examples?



Additional Heuristics from Beynon

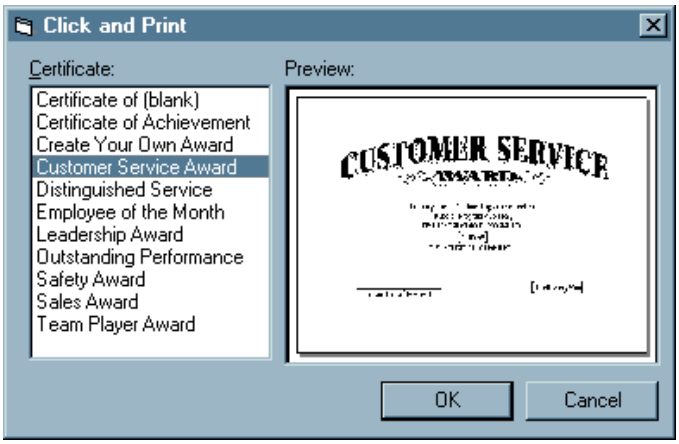
- Navigation – Enable users to move around the system
- Style – Design should be "stylish" and "attractive"
- Conviviality – System should be polite

Some (bad) examples…

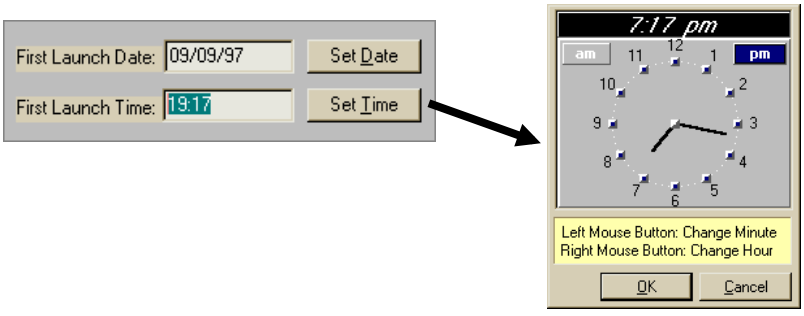
What's wrong with these?

Example 1

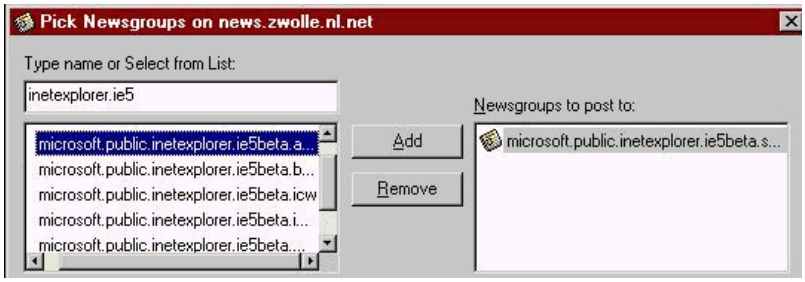
Example 1 - redesign



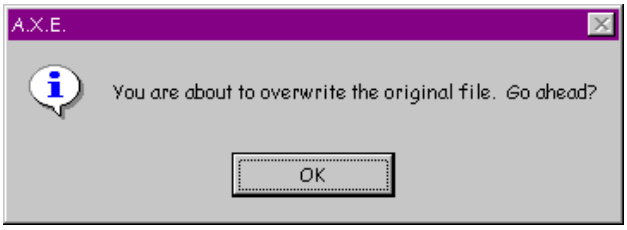
Example 2

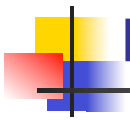


Example 3



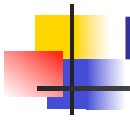
Example 4





Exercises in Inspection

- Break into groups
- One person must have a laptop or smart phone
- Pick an uncommon web site (preferred) or app
- Do a heuristic evaluation



Exercise 1

- "Heuristic evaluation"
- Critique a UI using Heuristics discussed
  1. Feedback
  2. Speak the User's Language
  3. Clearly Marked Exits
  4. Consistency
  5. Prevent Errors
  6. Minimize User Memory Load
  7. Flexibility / Shortcuts
  8. Simple Design
  9. Good Error Messages
  10. Help and Documentation
  11. Use Appropriate Affordances
  12. Visibility / Obviousness

Inspection methods

- Cognitive walkthrough
  - Identify a set of representative tasks
  - Walk through each step in the task and evaluate:
    1. Given current user goal, is the choice of action obvious (visibility)?
    2. After the action is taken, will users understand the feedback they get?

Exercise 2

- Cognitive walk-through
  - A more methodical approach to heuristic evaluation

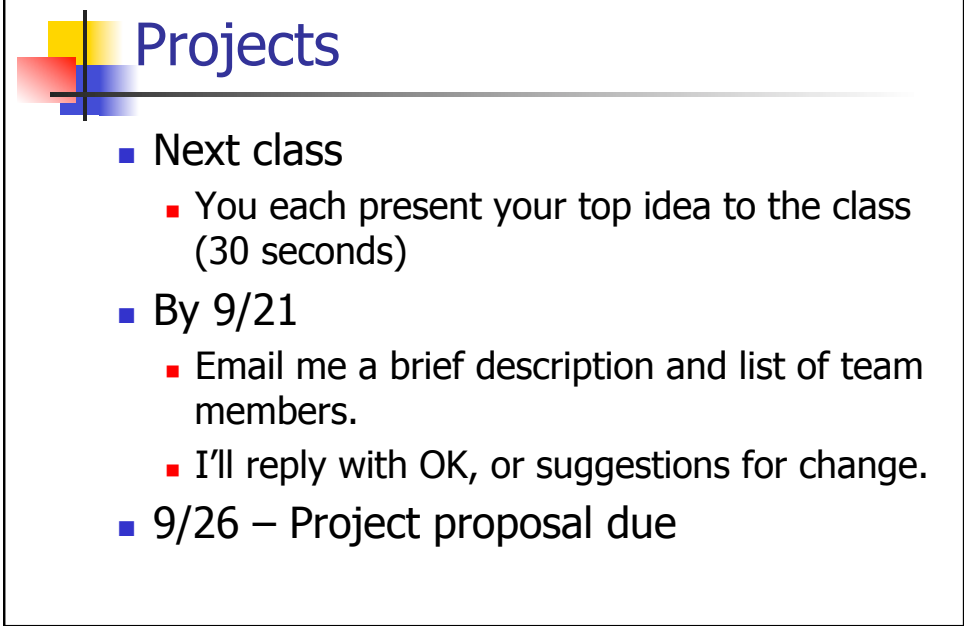
1. Define a task (as end goal, not how-to)
2. For each step (UI action)
   - Is the next action obvious?
   - Is the effect of the action taken obvious?

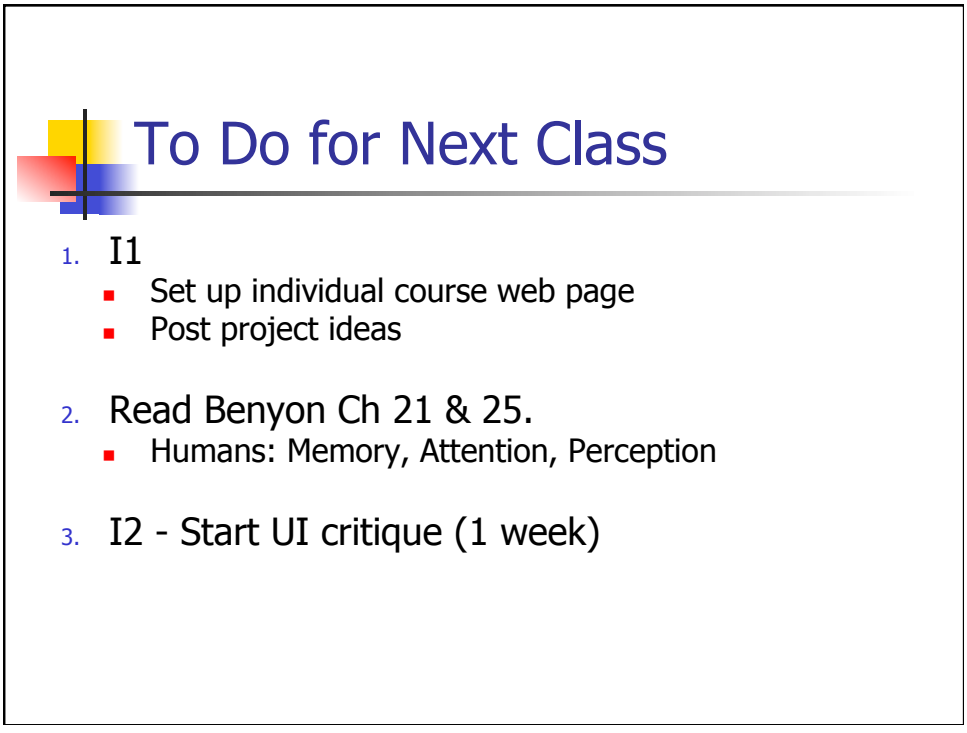
Individual Homework #2 UI Critique

- Find 2 good & 2 bad examples of UI design
- Use Nielsen's Heuristics and/or the list in Benyan 4.5.
  - Make explicit reference to them!
- Include visuals
- Make suggestions for improvement

Project Review

- Must have a substantial UI
- UI must be interactive
- Creative, original, non-obvious is better
- Ideas: research papers & past CHI, UIST, IUI
- Each project should have 2-4 members
- Ideally complementary skills


Projects
Next class
You each present your top idea to the class (30 seconds)
By 9/21
Email me a brief description and list of team members.
I'll reply with OK, or suggestions for change.
9/26 – Project proposal due


To Do for Next Class
1. I1
Set up individual course web page
Post project ideas
2. Read Benyon Ch 21 & 25.
Humans: Memory, Attention, Perception
3. I2 - Start UI critique (1 week)

Project Brainstorming

- Pick your favorite idea(s)
- Show your sketch
- Describe it in one minute