CS 160: User Interface Design

Human Information Processing 02/13/12

http://www.youtube.com/watch?v=WHxQU4RhyLk
Most heavily used features directly mapped (volume, play/pause)
Circular movements mapped to linear operations

**Due Today**

**Individual Prog. Assignment 2**
(Source code, executable and video on wiki)
Assigned: IPA 3 (due Feb 27)

Control your media browser using the Kinect
(Combine IPA 1 and IPA 2)

Assigned: Ind. Heuristic Eval. (due Feb 22)

Apply Nielsen's notes on Heuristic Evaluation to application of your choosing

Example: BART Trip Planning

Heuristic: Consistency and Standards

Explanation: The interface offers inconsistent ways to change different trip options. While a dropdown box to choose departure time and a button to reverse stations are available on the main screen, the origin and destination stations cannot be changed on this screen. To change these options, the user must click on the "i" icon in the top bar (which only becomes visible on mouse rollover).

Severity: 3 = Major usability problem: important to fix, so should be given high priority. I rank this problem as major because it occurs frequently - every time the user wants to change stations; and because it is persistent - there is no way for the user to change application behavior to put all controls on the same page.
Group Brainstorm (20 pts)

Great job on producing ideas and illustrating them
Need more targeted user groups!

Contextual Inquiry and Task Analysis

Due Feb 22
Find and interview 3 target users (not from class)
Analyze their tasks
Explain how your application addresses their needs
Compare to five closest existing applications
See wiki for details

Start now!
Finding participants will take time
We will not accept late group project assignments
Heuristic Evaluation

Usability Heuristics

“Rules of thumb” describing features of usable systems
Can be used as design principles
Can be used to evaluate a design

Example: Minimize users’ memory load
Heuristic Evaluation

Developed by Jakob Nielsen (1994)

Can be performed on working UI or on sketches

Small set (3-5) of evaluators (experts) examine UI
Evaluators check compliance with usability heuristics
Different evaluators will find different problems
Evaluators only communicate afterwards to aggregate findings
Designers use violations to redesign/fix problems

Nielsen’s Ten Heuristics

H2-1: Visibility of system status
H2-2: Match system and real world
H2-3: User control and freedom
H2-4: Consistency and standards
H2-5: Error prevention
H2-6: Recognition rather than recall
H2-7: Flexibility and efficiency of use
H2-8: Aesthetic and minimalist design
H2-9: Help users recognize, diagnose, recover from errors
H2-10: Help and documentation
H2-1: Visibility of System Status

Keep users informed about what is going on. Example: response time
0.1 sec: no special indicators needed
1.0 sec: user tends to lose track of data
10 sec: max duration if user to stay focused on action
Short delays: Hourglass
Long delays: Use percent-done progress bars
Overestimate usually better

H2-1: Visibility of System Status

Users should always be aware of what is going on
So that they can make informed decision
Provide redundant information

**Do you want to save the changes you made in the document "statements.txt"?**
Your changes will be lost if you don’t save them.

- Don’t Save
- Cancel
- Save

**Would you like to apply your changes before closing the Network preferences pane?**

- Don’t Apply
- Cancel
- Apply
H2-2: Match System & World

Speak the users' language
Follow real world conventions
Pay attention to metaphors

Bad example: Mac desktop
H2-3: User Control & Freedom

Users don’t like to be trapped!

Strategies
Cancel button (or Esc key) for dialog
Make the cancel button responsive!
Universal undo

Offer “Exits” for mistaken choices, undo, redo
Don’t force the user down fixed paths

Wizards
Must respond to Q before going to next step
Good for infrequent tasks (e.g., network setup) & beginners
Not good for common tasks (zip/unzip)
H2-4: Consistency and Standards

http://www.useit.com/alertbox/application_mistakes.html

H2-5: Error Prevention

Eliminate error-prone conditions or check for them and ask for confirmation
H2-5: Error Prevention

Aid users with specifying correct input

Lego Mindstorms

Don’t allow incorrect input

MIT Scratch
Preventing Errors

**Error types**

**Slips**
User commits error during the execution of a correct plan.

**Typos**
Habitually answer "no" to a dialog box
Forget the mode the application is in

**Mistakes**
User correctly executes flawed mental plan
Usually the result of a flawed mental model – harder to guard against

H2-6: Recognition over Recall
H2-6: Recognition over Recall

Minimize the user's memory load by making objects, actions, and options visible.

H2-7: Flexibility and Efficiency of Use

http://www.iphoneuxreviews.com/?p=114
H2-8: Aesthetic and Minimalist Design

No irrelevant information in dialogues
H2-8: Aesthetic and Minimalist Design

Present information in natural order

Occam’s razor
Remove or hide irrelevant or rarely needed information – They compete with important information on screen
Pro: Palm Pilot, iPhone
Against: Dynamic menus
Use windows frugally
Avoid complex window management
H2-9: Help Users Recognize, Diagnose, & Recover from Errors
Good Error Messages

H2-9: Help Users Recognize, Diagnose, & Recover from Errors
H2-10: Help and Documentation

Help should be:
• Easy to search
• Focused on the user’s task
• List concrete steps to carry out
• Not too long

Types of Help

Tutorial and/or getting started manuals
Presents the system conceptual model
Basis for successful explorations
Provides on-line tours and demos
Demonstrates basic features

Reference manuals
Designed with experts in mind

Reminders
Short reference cards, keyboard templates, tooltips…
Types of Help

Context sensitive help
Search

The Process of Heuristic Evaluation
### Phases of Heuristic Eval. (1-2)

1) **Pre-evaluation training**  
   Provide the evaluator with domain knowledge if needed

2) **Evaluation**  
   Individuals evaluate interface then aggregate results  
   Compare interface elements with heuristics

   **Work in 2 passes**  
   First pass: get a feel for flow and scope  
   Second pass: focus on specific elements

   **Each evaluator produces list of problems**  
   Explain why with reference to heuristic or other information  
   Be specific and list each problem separately

### Phases of Heuristic Eval. (3-4)

3) **Severity rating**  
   Establishes a ranking between problems  
   Cosmetic, minor, major and catastrophic  
   First rate individually, then as a group

4) **Debriefing**  
   Discuss outcome with design team  
   Suggest potential solutions  
   Assess how hard things are to fix
Examples

Typography uses mix of upper/lower case formats and fonts
Violates: Consistency and Standards (H2-4)
Problem: Slows users down

Fix: pick a single format for entire interface

Probably wouldn’t be found by user testing

Severity Rating

Used to allocate resources to fix problems

Estimates of need for more usability efforts

Combination of Frequency, Impact and Persistence

Should be calculated after all evaluations are in

Should be done independently by all judges
Levels of Severity

0 - don't agree that this is a usability problem
1 - cosmetic problem
2 - minor usability problem
3 - major usability problem; important to fix
4 - usability catastrophe; imperative to fix

Severity Ratings Example

1. [H2-4 Consistency] [Severity 3]

The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.
### Debriefing

- Conduct with evaluators, observers, and development team members
- Discuss general characteristics of UI
- Suggest improvements to address major usability problems
- Development team rates how hard things are to fix

#### Make it a brainstorming session

Little criticism until end of session

### Pros and Cons of Heuristic Evaluation
HE vs. User Testing

HE is much faster
1-2 hours each evaluator vs. days-weeks

HE doesn’t require interpreting user’s actions

User testing is far more accurate
Takes into account actual users and tasks
HE may miss problems & find “false positives”

Good to alternate between HE & user-based testing
Find different problems
Don’t waste participants

Why Multiple Evaluators?

Every evaluator doesn’t find every problem
Good evaluators find both easy & hard ones
Decreasing Returns

Problems Found

Benefits / Cost

Caveat: graphs are for one specific example!

Number of Evaluators

**Single evaluator achieves poor results**

Only finds 35% of usability problems
5 evaluators find ~ 75% of usability problems
Why not more evaluators??? 10? 20?
Adding evaluators costs more
Many evaluators won’t find many more problems

But always depends on market for product:

popular products → high support cost for small bugs
Summary

Heuristic evaluation is a discount usability method

Have evaluators go through the UI twice
   Ask them to see if it complies with heuristics
      Note where it doesn’t and say why

Have evaluators independently rate severity

Combine the findings from 3 to 5 evaluators
Discuss problems with design team

Cheaper alternative to user testing
   Finds different problems, so good to alternate

The Model Human Processor
Why Model Human Performance?

To predict impact of new technology/interface
- Apply model to predict effectiveness
- Could build a simulator to evaluate user interface designs
**Human Info. Processor**

**Processors:**
- Perceptual
- Cognitive
- Motor

**Memory:**
- Working memory
- Long-term memory

**Unified model**
- Probably inaccurate
- Predicts perf. well
- Very influential
Perceptual Processor

Physical store from our senses: sight, sound, touch, …
Code directly based on sense used
Visual, audio, haptic, … features
Selective
Spatial
Pre-attentive: color, direction…
Capacity of visual store
Example: 17 letters
Decay time for working memory: 200ms
Recoded for transfer to working memory
Progressive: 10ms/letter

How many 3’s

1281768756138976546984506985604982826762
9809858458224509856458945098450980943585
9091030209905959595772564675050678904567
8845789809821677654876364908560912949686
How many 3’s

1281768756138976546984506985604982826762
9809858458224509856458945098450980943585
9091030209905959595772564675050678904567
8845789809821677654876364908560912949686

Visual Pop-Out: Color

http://www.csc.ncsu.edu/faculty/healey/PP/index.html
Visual Pop-Out: Shape

Feature Conjunctions

http://www.csc.ncsu.edu/faculty/healey/PP/index.html
Preattentive Features

Perceptual Processor

**Cycle time**

Quantum experience: 100ms

Percept fusion

[Information Visualization. Figure 5. 5 Ware 04]
Perceptual Processor

**Cycle time**
- Quantum experience: 100ms
- Percept fusion
- **Frame rate necessary for movies to look continuous?**
  - time for 1 frame $< T_p (100 \text{ msec})$ -> 10 frame/sec.
- Max. morse code rate can be similarly calculated

**Perceptual causality**
- Two distinct stimuli can fuse if the first event appears to cause the other
- Events must occur in the same cycle

Perception of Causality [Michotte 46]

**Michotte demonstration 1.** What do you see? Most observers report that the red ball hit the blue ball. The blue ball moved "because" the red ball hit it." Thus, the red ball is perceived to "cause" the red ball to move, even though the balls are nothing more than color disks on your screen that move according to a program.

![Red Ball](https://cogweb.ucla.edu/Discourse/Narrative/Heider_45.html)
Perception of Causality [Michotte 46]

**Michotte demonstration 1**. What do you see? Most observers report that the red ball hit the blue ball. The blue ball moved “because the red ball hit it.” Thus, the red ball is perceived to “cause” the red ball to move, even though the balls are nothing more than color disks on your screen that move according to a program.

http://cogweb.ucla.edu/Discourse/Narrative/Heider_45.html

Perceptual Processor

**Cycle time**

Quantum experience: 100ms

Causality

![Graph showing causality percentages over time]

- Immediate Causality
- Delayed Causality
- Independent Events

Percentage of Judgments

<table>
<thead>
<tr>
<th>Time Before Second Object Moves (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
</tr>
<tr>
<td>50%</td>
</tr>
<tr>
<td>100%</td>
</tr>
</tbody>
</table>

14 98 182
Working Memory

Access in chunks
Task dependent construct
7 +/- 2 (Miller)

Decay
Content dependant
  1 chunk 73 sec
  3 chunks 7 sec

Attention span
  Interruptions > decay time

Long Term Memory

Very large capacity
Semantic encoding
Associative access
Fast read: 70ms
Expensive write: 10s
Can also move from WM to LTM via rehearsal

Context at the time of acquisition key for retrieval