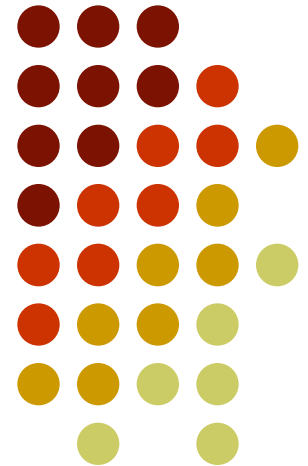


CSC 408 HCI

Basic Usability



Designing usable systems



- You need to consider
 - Users
 - Tasks
 - Environments
- You need to prioritize different usability goals
- You need to evaluate, and eliminate inevitable mistakes

- BUT
 - Some **basic principles of good design**

Principles versus Guidelines



- **Principles:** General, relatively abstract ‘ought’ rules
- **Guidelines:** More concrete, situation-specific advice on what to do to follow principles
- *Contrast*
 - Jakob Nielsen’s *10 Usability Heuristics for User Interface Design*
 1. Basic principles of good design
 2. Categorization of design mistakes covering nearly everything
 - Jakob Nielsen’s *113 Design Guidelines for Homepage Usability*
 1. One of many sets of rules giving concrete advice
 2. Can evaluate websites by looking for violations

Ben Shneiderman's 8 Golden Rules



- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Windows Interface Guidelines



- Set of general principles for interface design in Microsoft's software development document
 - directness
 - user in control
 - consistency
 - forgiveness
 - feedback
 - aesthetics
 - simplicity

Jakob Nielsen's list of 10 UI Usability Heuristics



- According to Nielsen, the ten most general principles for interaction design

From <http://www.nngroup.com/articles/ten-usability-heuristics/>

- **Visibility of system status**

- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

- **Match between system and the real world**

- The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Jakob Nielsen's list of 10 UI Usability Heuristics



- **User control and freedom**
 - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- **Consistency and standards**
 - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- **Error prevention**
 - Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Jakob Nielsen's list of 10 UI Usability Heuristics



- **Recognition rather than recall**
 - Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- **Flexibility and efficiency of use**
 - Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- **Aesthetic and minimalist design**
 - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Jakob Nielsen's list of 10 UI Usability Heuristics



- **Help users recognize, diagnose and recover from errors**
 - Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- **Help and documentation**
 - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

Be consistent

Consistency



- Internal consistency
 - Different parts of the same system work in the same way
- External consistency
 - System works in the same way as other similar systems
 - How do you take a new car for a test drive?

Consistency



- Avoids confusion and minimizes learning effort
 - Enable user to build a reliable mental model of how the interface works
 - Makes the interface familiar and predictable by providing a sense of stability
 - Allows users to transfer existing knowledge to new tasks and focus more on tasks because they need not spend time trying to remember the differences in interaction
 - Allows users to learn and strengthen procedural skills
- Always important: names of commands, layout of information, and operational behaviour.

Redesigning: Continuity



- You *can* redesign and modify, but users' experiences should help them, not confuse them...
- **Make difference visible:** As important as making sure that controls with the same behaviour look the same, is making sure that controls that do different things look different

Stimulus-response compatibility



- How naturally a system's response matches what we expect based on stimulus
 - Bottom UPWARDS = increase
 - Left-to-RIGHT = increase
 - CLOCKWISE = increase
- Use standard stimulus-response relationships
- Never mix positive and negative S-R relationships

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

Feedback to
User

Feedback from the system



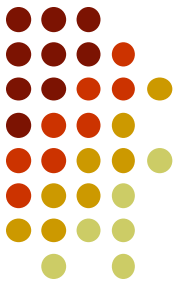
- Every action the user makes should produce a perceptible response.
- The intention is to reduce user uncertainty that the system has:
 - received the last input,
 - is currently doing something about it,
 - or is waiting for the next input.

Feedback from the system



- Commands should result in some visible change to the interface
 - E.g ‘mail has been sent’ in response to a ‘Send’ command
 - Presentation of objects on screen updated to reflect their current state
- Task analysis should enable appropriate information to be identified as feedback for a specific task

Feedback from the system



- System should show its *status*
 - What *mode* it's in (what you can see or do)
 - What *state* it's in (where in process you are)
 - What it expects or *enables the user* to do



Feedback: Response times

- Response time for feedback should be appropriate to the type of user action:
 - e.g. response to keystroke - instantaneous;
response to command input - may take longer
- Provide ‘system busy’ feedback if time will exceed a few seconds or is unpredictable
- Provide indication of how many transactions remain, for example as a bar chart or as a percentage
- This largely disappeared as a problem with fast single user PCs and has re-appeared with distributed web-based applications

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

System should match real world

Matching how the users think



- Interface should let the users work with
 - Users' concepts
 - Users' terminology
 - Operations that fit users' conception of actions
- Dialogues should
 - Use simple familiar language
 - Present information in clear logical order
 - Yield closure: natural beginning, middle, end

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

User in
Control

User in Control



- *User initiates actions*, not the computer or software
 - *User controls pace* of activities
 - Use techniques to automate tasks, but implement them in a way that allows the user to chose or control the automation
 - Users should be able to personalize aspects of the interface, such as colour, fonts, or other options

Minimal User Input



- User input should be as little work as possible
 - BUT balance between number of keystrokes or mouse movements/clicks and memory load
 - Reducing keying errors increases speed of data entry
- *Edit a command* that has produced an error rather than retyping the command
- *Allow selection from a list* rather than typing in a value (recognise rather than recall)
- Do not request input of information which can be derived automatically or which has been entered previously
- Use *default values*

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

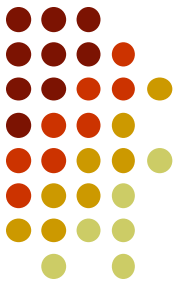
Reduce
Memory Load

Minimize user memory load



- Remembering stuff from moment to moment is **hard** and **unreliable**: *Don't make the users do it!*
- Basic rule: Don't expect the user to remember what has already been done; make this visible at the interface
 - If a command is made up of a number of pieces of data entered by the user in sequence, display these rather than expect the user to remember the data already entered
 - Help the user remember where they are in a transaction sequence - Menu 2/5 Step 1 - 4

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

Error
Prevention

Forgiveness



- Easy UNDO
 - Where possible, allow easy reversal of actions
 - Mistakes have low cost
 - Allow exploration

Error Prevention



- Design Out Errors
 - Make errors less likely
 - Perceptual confusions
 - Similar actions
 - Make users check likely mistakes or irreversible actions
 - If possible, make errors impossible

Murphy's Law



“If there is a wrong way to do something, someone will do it that way”

-- attributed to Edward Murphy (original phrasing debated)

- Design Principle: Design things so that there ISN'T a wrong way to do something

- Design classic:
 - 3.5 inch floppy disk



Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

Appropriate
User Support

Appropriate user support



ERROR messages

- should explain what is wrong and what corrective action is required
- should use ‘jargon’ familiar to the user

HELP messages

- important to recognise different types of help
 - should be available when required and context-specific
 - can the user get help about what responses are possible at a given point in a dialogue?
-
- Often this support is poorly designed in terms of what information is given to the user

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetics
- simplicity

Provide
Flexibility

Flexibility



- Measure of how well a dialogue can cater for *different levels of user skill*
- Provide alternative means of achieving the same goal which match different models of how the interface works
 - e.g. word selection: cursor to start of word and double click, click and drag, click and shift-click
 - e.g. word deletion: word highlighted and Control +X key, select 'Cut' menu option, backspace

Flexibility



- Adapt to the skill level of the user by
 - providing accelerators
 - allow user to answer ahead
 - provide key bindings for menu options
 - providing macro facility
 - accepting abbreviations for command words
 - accepting synonyms (alternative names)
 - allowing user to choose level of instructions or help

Many common elements...



Nielsen

- visible system status
- match between system and world
- user control and freedom
- consistency
- prevent errors
- recognition rather than recall
- flexible & efficient
- aesthetic design
- help users recover from errors
- good help and documentation

Shneiderman

- strive for consistency
- enable frequent users to use shortcuts
- offer informative feedback
- design dialogues to yield closure
- offer simple error handling
- permit easy reversal of actions
- support internal locus of control
- reduce short term memory load

Microsoft

- directness
- user in control
- consistency
- forgiveness
- feedback
- aesthetic
- simplicity

Aesthetic
Design



Aesthetic issues

- Good-looking interfaces are attractive!
- Elegance tends to go with good usability
- Careful use of colour, graphics and formatting can make the design more aesthetically pleasing

- BUT
 - Danger that you can hide or obscure controls or key information
 - Need to get the right balance
 - Nielsen advocates Simplicity - particularly for Website design

The Principle of Affordance



- It should be obvious how a control is used
- Objects *afford* certain actions and not others
 - How their appearance suggests they are used
 - How they can be physically manipulated
 - Affordances are partly learned conventions

Some physical affordances...



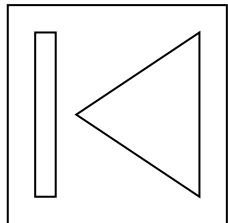
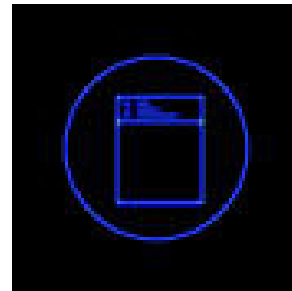
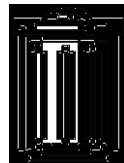
- What actions do these physical objects afford?
- Are they obvious?



Some virtual affordances...



- What do these screen objects afford?
 - What if you were a novice user?
 - Would you know what to do with them?



The Principle of Feedback

- It should be obvious when a control has been used





Illusion of Simplicity

- Beware the *Illusion of Simplicity*
 - Principle: Any attempt to hide complexity will serve to increase it
- Simple-looking screens with hidden controls (e.g. Mac hidden scrollbars)
 - Look great in demos
 - May help to ship units
 - Usability nightmare
- Principle: If the user can't find it, it doesn't exist

Discoverability



- Design interfaces so that users can **discover all the functionality**
 - Q: How do you know if features are discoverable?
 - A: Need to do user testing
- MS Office 2007 redesign: the ribbon was introduced to improve discoverability (as users kept demanding functionality they already had but couldn't find) - largely successful
- **Findability**: How easily users can *locate* features they know exist

Using usability heuristics in design



- Useful
- BUT
 - You'll still make usability mistakes
 - Can you or can't you violate guidelines and conventions?
- You'll still need to **test** and **revise!**

Can you break the rules?



- Depends which ones
- Feedback is vital
- Consistency matters

- BUT
 - Some conventions can be varied
 - Can put menus on right
 - Can vary web link colours
 - You'll need to TEST to see whether your innovative/different design creates problems or not
 - You'll need to TEST ANYWAY

Reading



- Jakob Nielsen **113 guidelines for web homepages**
<http://www.nngroup.com/articles/113-design-guidelines-homepage-usability/>
- **Chapter 9.1-9.3** of Stone, Jarrett Woodroffe, Minocha,
User Interface Design and Evaluation