### Lecture 8: Design Principles

### Usability Guidelines ("Heuristics")

- Plenty to choose from
  - Nielsen’s 10 principles
    - One version in his book
    - A more recent version on his website
  - Tognazzini’s 16 principles
  - Norman’s rules from Design of Everyday Things
  - Mac, Windows, Gnome, KDE guidelines
- Help designers choose design alternatives
- Help evaluators find problems in interfaces ("heuristic evaluation")

### Guidelines From Earlier Lectures

- User-centered design
  - Know your users
  - Understand their tasks
- Fitts’s Law
  - Size and proximity of controls should relate to their importance
  - Tiny controls are hard to hit
  - Screen edges are precious
- Memory
  - Use chunking to simplify information presentation
  - Minimize working memory
- Color guidelines
  - Don’t depend solely on color distinctions (color blindness)
  - Avoid red on blue text (chromatic aberration)
  - Avoid small blue details
- Norman’s principles of direct manipulation
  - Affordances
  - Natural mapping
  - Visibility
  - Feedback

### 1. Match the Real World

- 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
- Metaphors are useful but may mislead
2. Consistency and Standards

- Principle of Least Surprise
  - Similar things should look and act similar
  - Different things should look different
- Other properties
  - Size, location, color, wording, ordering, ...
- Command/argument order
  - Prefix vs. postfix
- Follow platform standards

Source: Interface Hall of Shame

Kinds of Consistency

- Internal
- External
- Metaphorical

Case Against Consistency (Grudin)

- Inconsistency is appropriate when context and task demand it
  - Arrow keys
- But if all else is (almost) equal, consistency wins
  - QWERTY vs. Dvorak
  - OK/Cancel button order

Users don’t read manuals
  - Prefer to spend time working toward their task goals, not learning about your system
- But manuals and online help are vital
  - Usually when user is frustrated or in crisis
- Help should be:
  - Searchable
  - Context-sensitive
  - Task-oriented
  - Concrete
  - Short
4. User Control and Freedom

- Provide undo
- Long operations should be cancelable
- All dialogs should have a cancel button
- User-provided data should be editable

5. Visibility of System Status (Feedback)

- Keep user informed of system state
  - Cursor change
  - Selection highlight
  - Status bar
  - Don’t overdo it...

- Response time
  - < 0.1 s: seems instantaneous
  - 0.1-1 s: user notices, but no feedback needed
  - 1-5 s: display busy cursor
  - > 1-5 s: display progress bar

6. Flexibility and Efficiency (Shortcuts)

- Provide easily-learned shortcuts for frequent operations
  - Keyboard accelerators
  - Command abbreviations
  - Styles
  - Bookmarks
  - History
7. Error Prevention

- Selection is less error-prone than typing
  - But don’t go overboard...

- Disable illegal commands
- Keep dangerous commands away from common ones

Description Error

- Intended action is replaced by another action with many features in common
  - Pouring orange juice into your cereal
  - Putting the wrong lid on a bowl
  - Throwing shirt into toilet instead of hamper
  - Going to Kendall Square instead of Kenmore Square

- Avoid actions with very similar descriptions
  - Long rows of identical switches
  - Adjacent menu items that look similar

Capture Error

- A sequence of actions is replaced by another sequence that starts the same way
  - Leave your house and find yourself walking to school instead of where you meant to go
  - Vi :wq command
- Avoid habitual action sequences with common prefixes

Mode Error

- Modes: states in which actions have different meanings
  - Vi’s insert mode vs. command mode
  - Caps Lock
  - Drawing palette

- Avoiding mode errors
  - Eliminate modes
  - Visibility of mode
  - Spring-loaded or temporary modes
  - Disjoint action sets in different modes
More Mode Errors

8. Recognition, Not Recall (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

9. Error Reporting, Diagnosis, Recovery
- 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 nonblaming
  - Not “fatal error”, not “illegal”
- Hide technical details (stack trace) until requested

10. Aesthetic and Minimalist Design
- “Less is More”
  - Omit extraneous info, graphics, features
10. Aesthetic and Minimalist Design

- Good graphic design
  - Few, well-chosen colors and fonts
  - Group with whitespace
  - Align controls sensibly
- Use concise language
  - Choose labels carefully

source: Interface Hall of Shame

Meet expectations
1. Match the real world
2. Consistency & standards
3. Help & documentation

User is the boss
4. User control & freedom
5. Visibility of system status
6. Flexibility & efficiency

Handle errors
7. Error prevention
8. Recognition, not recall
9. Error reporting, diagnosis, and recovery

Keep it simple
10. Aesthetic & minimalist design

Tog’s 16 Principles

- Anticipation
- Autonomy
- Color blindness
- Consistency
- Defaults
- Efficiency
- Explorable interfaces
- Fitts’s Law
- Human interface objects
- Latency reduction
- Learnability
- Metaphors
- Protect users’ work
- Readability
- Track state
- Visible navigation

Shneiderman’s 8 Golden Rules

- Consistency
- Shortcuts
- Feedback
- Dialog closure
- Simple error handling
- Reversible actions
- Put user in control
- Reduce short-term memory load