A Look at Design …

• Possible User Interface Goals:
  – Intuitive
  – Easy to Use
  – Easy to Learn
  – Pleasant to look at
  – Agrees with what people already know
  – Others?
Bad Design Examples

Complex Shower

Cell phone power button

Automobile title number

Bad Design Examples

No camping sign difficult to see at dusk

Heavy glass door with center handle

Error Message

Error

Uncought Exception:
array_index_out_of_bounds.
Execution terminating.
All data not yet saved will be lost.

OK
Leveraging Affordances

An affordance is an aspect of an object that gives clues about what it is for, how it should be used

- Common in both real world and user interface
  - door knob, steering wheel, stairs, seat of chair
  - scrollbar, window handles, cursors
- Involve tradeoffs
  - can be distracting and take up space
  - if you hide them users may not discover (e.g. disappearing scroll bars)

Affordance Examples

Labels look like buttons

Pointer does not afford click and drag
Internal Consistency

- *Internal* consistency within a system
  - on the same screen: button shape & labels, fonts, etc.
  - from screen to screen: UI controls, layout, font family
  - applies to *text vocabulary* too
    - Up/Down, Previous/Next *not* Up/Next, Previous/Down
- Why?
  - more rapid/accurate pattern induction (e.g., what buttons look like)
  - more confident interaction behavior

Visual Program Design

- Visual features used consistently, design “signature”
  - e.g., title bar, tool palette, window border, title line, standardized set of components and layout
  - not necessarily a functional feature, e.g. special border
Visual Program Design

• Promotes a sense of unity and coherence
  – easier to make connections from screen to screen
• Caution: repeated gratuitous decoration or animation creates a design program that detracts
  – bright saturated colors, black backgrounds
  – animated links, performing titles

External Consistency

• *External* consistency across different systems or between the real world and a system
• Examples
  – the Mac family of apps, Windows apps, the Web
• Why?
  – enables *transfer of learning* from one system to another
  – while mismatches lead to *interference* (wrong actions)
• Caution: consistency can be in the eye of beholder
• 2nd caution: special needs of the user’s task
External Consistency Example

Zoom control is easily mistaken for a scrollbar

Visual Metaphors

- Serve both designer and user
  - for designer: generate ideas, assure external consistency, reduce number of decisions
  - for user: leverage prior knowledge, evoke sense making
- One metaphor may influence multiple design issues
  - library
    - activities: browsing, borrowing
    - information: shelves, card catalog, alphabetical order
    - interaction: sequential browsing, cover first, check-out
  - other examples: shopping cart, messy desktop
- Caution: beware of over-literal application
  - do not want to recreate real world problems
Metaphor Example

- Macintosh trash can (video clip)
  - Place floppy disk in trash to eject
  - OS X revised metaphor

Information Models

- Helps users make sense of a large amount of information and complex functionality
- An “information space” that users navigates
- Want a structure that is simple and coherent, but at the same time comprehensive and flexible
- Many techniques for designing information models
  - hierarchy: menu systems, files and folders
  - directed graph: hypertext links
  - spatial structures: tables, maps, 3D structures
Dynamic Information Models

- **Focus+Context Displays:** detail within overview

![Bifocal Display – London Underground Map](image1)

!["Bubble" Display – DC Metro Map](image2)

Dynamic Information Models

- **Overview+Detail Displays:** detail next to overview

![Radar View – Blacksburg Map](image3)

![Visualization of Human Brain](image4)
Dynamic Information Models

• Animated movement can promote 3D perceptions
• Experience “moving around in” a structure

Perspective Wall Visualization

Cone Trees Visualization

Dynamic Information Models

• Multiple coordinated views (tiled windows, frames)
  – one view may “index” others, control the updates
  – more complex case has multi-way dependencies

Snap-Together Visualization