Computers & the Humanities 281

Stack Design

Instructional Design and Organization

- get away from static page turning
- use hierarchical/branching organization when content allows
- give sense of orientation and control, but not necessarily "do anything from anywhere"
- stacks need beginning, body and close
- provide means to exit the application where appropriate (e.g., a "Quit" button)
- don't let frills overshadow substance
- use alerts for irrecoverable actions
- use alerts for error conditions
- don't use alerts for routine input

Effective Use of Revolution

- navigation/control buttons in consistent location
- mask or disable arrow buttons on first/last cards in sequence
- use grouped objects for constant elements
- use group scripts for efficiency
- extract common tasks as handlers and/or functions (a.k.a. subroutines)
- don't forget quotes around literals
- use meaningful names for variables, handlers, fields, buttons, cards, etc.
- use comments in scripts

Visual Elements

- standard graphics principles: visual balance, simplicity, white space
- convey importance with size, bold, position, color
- convey commonality with same button style, font, color, position, etc.; convey distinction with different style, font, color, etc.
- avoid "TV remote" controls, i.e., lots of tiny buttons that all look alike, have to read label to distinguish
- use meaningful icons
- give visual feedback for button clicks
- use transition visual effects consistently to convey information (don't overuse)
- set cursor to watch or busy in handlers that take any time
- use color to convey information, not just window dressing
- subtle colors for backgrounds, bright colors sparingly, for important items
- consider resolution/colors for all possible environments
- keep contrast high when text on color
- stick to dark on light for body text
- light/white on dark can be effective if used sparingly

Text

- avoid scrolling text fields
- lock text fields
- use display fonts sparingly, never for body text
- stick with 2-3 fonts
- use the same font to convey information consistently
- use curled quotes, em-dashes, etc. as per typographic conventions

Sound

- use sound to convey information, not just window dressing
- don't be too raucus, especially with negative feedback
- avoid joke sounds or long sounds that can get quickly get old and annoying
- use beep only for error messages
- for sound content applications give appropriate controls
- don't use sound as only feedback

Content

- proofread, proofread, proofread, then have someone else proofread
- test, test, test, then have someone else test
- use real content
- avoid ungrammatical distractors
- reinforce correct responses better than punish wrong responses
- avoid making wrong responses "more fun" than right ones

- try to give instructive feedback
- design to test content, not test-taking skill
- randomize intelligently but not obsessively
- don't force students to keep guessing until they guess right

Instructions and Help

- design for intuitiveness, but make help available anyhow
- context sensitive help
- don't overkill help
- avoid over solicitous, condescending help
- avoid over annoying, nagging help
- allow people to turn off active mouseWithin-type helps
- avoid "Click Here" instructions

Meta Issues

- design for all reasonable platforms, state platform requirements clearly in documentation/advertising
- don't ignore copyrights
- acknowlege sources
- hire an artist
- hire a graphics designer
- require some form of documentation
- plan for maintenance and upgrades
- use security to prevent unintended damage
- don't expect to foil knowlegable users
- don't lock out lab managers

Course Schedule
Main Page