

Principles of UI design



Lecture 25
CS 211 Spring 2006
Andrew Myers

Announcements

- Assignment 6 due in about one week
 - Get client working soon, GUI later
 - Try using telnet to talk to server to understand protocol
 - Correctness first, performance later

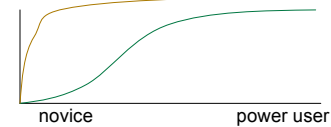
What makes a good (G)UI?

- Good UI lets user accomplish goals
 - Wants success more than knowledge
- Does not aim at:
 - What makes programmer's job easiest
 - What makes marketing guy happiest
 - Feature count is not the goal
 - What users say they want
 - Users usually don't think about end goals!
- A well-designed interface makes users happy!
 - Car designers work very hard on this...

Know your user

- Frequent or occasional use?
- Gets training?

- Match learning curve to user



- Match visual design to user

The on-ramp

- Gentle learning curve essential for infrequent use/quick learning
 - Can user get value out of program immediately?
 - Does user have to remember a lot?
- UI consistent with familiar applications
- Simple, clear displays
- No loaded guns

The power user

- Applications for frequent/long use should focus on maximizing productivity
 - Don't dumb it down or condescend to user
 - Accommodate multiple skill levels: gentle learning curve still useful
 - Optional power interface?
- Optimize for common actions
 - Few input actions to achieve goals
 - Exploit muscle memory and visual memory
 - Users know where controls are (don't move them around!)
 - Iconic representations easy to recognize
- Sophisticated, information-rich displays
 - Don't waste valuable screen real estate

UI as dialogue

- Interface use is a dialogue between user and computer
- Break dialogue into short interactions
 - Intermediate goal satisfaction
 - Undo
 - Avoid side effects
- GUIs are successful because they *restrict the vocabulary*
 - Hard to say wrong things
 - "Is -la" vs. double-click
 - Exploit **context sensitivity** (double-click on directory vs. file)
 - Avoid gratuitous memorization

Talking to the user

- Application should reflect its state to user
 - Show when it's busy doing something (but stay responsive)
 - Show how close it is to done
- Controls should be apparent
 - User should be able to identify possible actions to take
 - Controls can highlight/animate to show location
 - User should know when input occurs
 - Avoid subtle side effects
- Minimize error possibilities
 - Don't set the user up for a fall
 - Gray out controls leading to error messages

Modes

- Modes are states of the application in which only some tasks can be performed
 - UI may change in different modes
- Bad:
 - Must be exited from but is hard to exit from
 - Not associated with any visually apparent object
 - Serves no meaningful role
- Good: (e.g., windows!)
 - Restrict vocabulary so easier to do task (fewer inputs, shorter menus...)
 - Application help is more relevant
 - Clearly indicate current mode (even tie to physical action, e.g. dragging)
 - Easy to exit

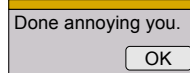


Choices

- Menus, buttons, radio buttons support choosing task
- But user can only keep 7 things in head
 - Avoid long menus, many buttons...

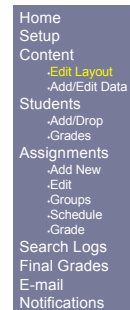
UI as a place

- Most applications have more than one "screen" or "window"
- Ties into built-in notion of "place"
 - Window = room in a house
 - Modes!
- Avoid unnecessary rooms
 - Don't make the user go around the house to accomplish a goal
 - Add a room for tasks that need their own
 - Make it easy to find your way around
 - Pop-up dialog windows are rooms too



Keeping the user situated

- Windows are modes
 - Make it easy to get from one window to others
 - Make it easy to know where you are
- Overview displays help
 - Show whole space and where you are in it
 - Drawing programs
 - Powerpoint slide view
 - CMS
- Keep navigation easy
 - Obvious UI
 - Implicit UI



Visual design

- Visual appearance should be clear and consistent
 - Similar things should look similar
 - Avoid distracting visual clutter
- Use color and intensity carefully
 - Eye needs good contrast (in brightness!) to read
 - High contrast and bright colors for important things
 - Low contrast, dull colors if unimportant/inactive
 - Assign colors consistent meanings (red = bad/danger, green=good, blue=active control,...)



BAD COLOR CHOICES!

Visual coding

- Information can be compactly encoded in visual appearance
 - Shapes: can distinguish <15 rapidly
 - Size, length: <6
 - Color: <10 (but: 7% of males RG colorblind)
 - Orientation: 24
 - Texture
 - Linewidth
- Multiple dimensions can reinforce



Use space well

- Avoid attention-getting separators

8	1	6	8	1	6	8	1	6	8	1	6
3	5	7	3	5	7	3	5	7	3	5	7
4	9	2	4	9	2	4	9	2	4	9	2

- Separate and group information with space, low-contrast elements, color cues
- Information-rich displays save space
 - Use visual coding cues, icons to add info
 - Wasted space = more rooms = more navigation
 - Avoid clutter: maximize information/"ink" ratio

UI as abstraction

- A user interface creates a mental model: an **abstraction** of what application is doing
- Model should serve user's needs and goals
 - Goal is not to make user's mental model match implementation
 - Expose what user needs to succeed!

References

- *Principles of Computer Graphics*, by Foley and Van Dam. Ch 8.2-8.3, 9
- *About Face 2.0*, by Cooper and Reimann.
- *Envisioning Information*, by Tufte