User Interface Design: Principles, Fitts' Law

15 January 2009
CMPT370
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Fluid and C++ program design

- Two ways of structuring your FLTK program:
- BankInterest example: main() in Fluid
  - #include separate file for core logic
- CubeView example: main() in separate C++ file
  - Fluid file: defines CubeViewUI class
    - which contains an Fl_Window
    - which contains a CubeView
      - which is a subclass of Fl_Gl_Window
  - CubeMain.cxx:
    - main() instantiates a CubeViewUI

UI design principles

- Constantine+Lockwood http://www.foruse.com/
- Structure: hierarchy, layout: windows, tabs, etc.
- Simplicity: make common tasks easy
  - Epicentre: design around primary purpose
- Visibility: need-to-know basis
- Feedback: current state, errors, etc.
- Tolerance: be flexible to user mistakes, save user data / user's hard work
- Reuse: consistent naming, behaviour

Fitts' law

- "The time to acquire a target is a function of the distance to and size of the target."
  - MT = a + b log(D/W)
  - MT: movement time
  - D: distance
  - W: width of target (for 1D)
- Tapping test
- Tip: edge of screen is like an infinitely big target

- More reading: Walker, Neff and Smelcer, John (1990). "A Comparison of Selection Time from Walking and Bar Menus." Proceedings of CHI'90, Addison-Wesley, Reading, Mass., pp. 221-225.

Applying Fitts' law: #1

http://www.asktog.com/columns/022DesignedToGiveFitts.html

- MS toolbars can display a label below each tool button. Why are labelled buttons faster to access?
  - Bigger target, easier to hit
  - Spreads the buttons out, less likely to hit wrong button

Applying Fitts' law: #2

http://www.asktog.com/columns/022DesignedToGiveFitts.html

- You have a palette of tools in a graphics application that consists of several 16x16-pixel icons laid out as a 2x8 matrix along the left-hand edge of the screen.
- Without moving the matrix from the left-hand side of the screen or changing the size of the icons, how can you speed up access to the average tool?
  - Lay out as 1x16 column flush against left-hand side: use the screen edge

Applying Fitts' law: #3

http://www.asktog.com/columns/022DesignedToGiveFitts.html

- A right-handed user is known to be within 10 pixels of the exact center of a large, 1600 X 1200 screen. You will place a single-pixel target on the screen that the user must click on.
- List the five pixel locations on the screen that the user can access fastest. For extra credit, list them in order from fastest to slowest access.
  - Right under the mouse
  - Bottom-right, top-left
  - Top-right, bottom-left

Applying Fitts' law: #4

http://www.asktog.com/columns/022DesignedToGiveFitts.html

- Windows' taskbar can be oriented along the edge of the screen and can auto-hide or be constantly displayed. Why is this method of triggering an auto-hidden taskbar a bad idea?
  - Uses up prime real-estate: edge!
  - Too easy to display by accident
- What location would be even easier to access?
  - Screen corner
- Is Apple's Dock better?

Applying Fitts' law: #5

http://www.asktog.com/columns/022DesignedToGiveFitts.html

- Mac application menus go along the top of the desktop; Windows/Linux application menus are attached to the application window. Debate which is better from a usability perspective.
  - Menu on desktop: easier to access, uses screen edge, consistent location
  - Menu on window: with several windows open, clearer which window the menu pertains to

Fitts' law and menus

"The time to acquire a target is a function of the distance to and size of the target."

- How should we optimize menus in light of this?
  - Nesting too deeply is bad, too
  - Minimize distance to oft-used entries
    - MS auto-hide unused menu entries?
    - Must navigate a precise path
  - Centre child menus?
  - Variable-size entries?
  - Pie menus?