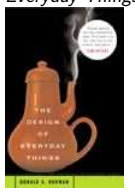
cis20.1
design and implementation of software applications I
fall 2007
lecture # I.5

topics:

- affordances and design
- principles of good design
- usability testing

affordances and design.

- psychologist James J. Gibson created the term "affordance" to describe the relationship between an actor and his world [Gibson, 1977]
- psychologist Donald Norman used the term "affordance" in his book: *The Design of Everyday Things* [Norman, 1990]



- but admits he should have employed "perceived affordance" in the context of design, because the important aspect is what the user perceives
- the term can be applied to both physical and virtual objects
- here, we are interested in virtual objects that appear on a computer screen

- can we design an interface where the user perceives that actions s/he takes will enable the results s/he desires?
  i.e., where the interface affords the actions intended by the designer?
- Norman stresses the importance of understanding the culture of a designer's audience
- Norman says: "In the world of design, what matters is:
  1. If the desired controls can be perceived
  1.a. In an easy to use design, if they can both readily be perceived and interpreted
  2. If the desired actions can be discovered
  2.a. Whether standard conventions are obeyed"
- Four principles for interface design:
  1. Follow conventional usage, both in the choice of images and the allowable interactions.
  2. Use words to describe the desired action (e.g., "click here" or use labels in front of perceived objects).
  3. Use metaphor.
  4. Follow a coherent conceptual model so that once part of the interface is learned, the same principles apply to other parts.

- some devices/interfaces *force* functionality; e.g., you have to put the key in your car to start it
- 7 states of action:
  1. form the goal
  2. form the intention
  3. specify action(s)
  4. execute action(s)
  5. perceive state of the world
  6. interpret state of the world
  7. evaluate outcome
- *gulf of execution* — how well system allows user to perform intended actions
- *gulf of evaluation* — how well system allows user to perceive state after executing actions (i.e., *feedback*)
- *design for error!*
  expect that the user will make errors; assess what errors you as the designer think the user will make, and plan for those errors either by redesigning to eliminate the possibility of them occurring or including methods for handling user error

comparison of interface techniques.

- direct manipulation
  - advantages: visual presentation of tasks, concepts; easy to learn and retain; easy to avoid errors; encourages exploration
  - disadvantages: can be hard to program; more sophisticated hardware/software requirements (e.g., pointing device, graphics)
- menu selection
  - advantages: easy to learn; reduces keystrokes; uses "dialogs"; structures decision-making; supports error handling
  - disadvantages: perhaps too many menus; can slow down expert users (too many menus/keystrokes); screen real estate; requires rapid display rate
- form fill-in
  - advantages: simple data entry; limited training required; convenient reminders of what to do
  - disadvantages: screen real estate
- command language

  - advantages: flexible; great for expert users; allows user-defined macros
  - disadvantages: poor error handling; requires significant training
- natural language
  - advantages: no syntax to learn
  - disadvantages: requires clarification dialogs; more keystrokes (unless voice activated); unpredictable—hard to program!

eight rules of interface design.

from [Johnson, 2006]

1. be consistent
2. aim for universal usability
3. provide helpful feedback for every user action
4. provide closure with dialogs
5. prevent errors where possible and otherwise handle errors elegantly
6. allow reversal of actions
7. make users feel in control ("internal locus of control")
8. limit memory load

memory test.

study this image for 2 minutes

lessons from psychology.

- Fitt's Law (1954):
  measured the amount of time it takes a human to move his/her hand to a target various distances away, in 1-dimension
  other researchers later experimented with movement in 2D and 3D
- lesson: *when you design your interface, think about screen real estate in terms of where your user will have the mouse and where they will want to move it to*
- Power Law of Practice:
  the more you do something, the better you get at it—in the beginning; but over time, progress slows down and learner gets tired
- lesson: *short learning curve is okay, but keep it short!*

things to avoid.

- bad designs: `http://www.baddesigns.com/`
- things that don't work the way you expect them to
- different things that are too similar
- things that are hard to see
- things that don't work well together
- things that get in the way
- things that are hard to handle
- things that are hard to remember
- things that don't fit
- displays that look like controls
- incompatible/unexpected/unnatural mapping of controls to devices
- and on and on and on...

visual structure.

- use of color:
  - color can help but be aware of color blindness
  - limit the number of colors
  - use color to group things together (that you want to group together)
  - use color to support a task (e.g., brightness makes things easy to find)
  - remember color conventions (e.g., red, yellow, green)
- grouping of objects
  - group related objects together
  - visual hierarchy can reflect object hierarchy
  - visual relationships: position, size, "weight"
- balance (use it!)
- clutter (avoid it!)
- ask yourself: where does the eye naturally go?
- use symmetry to ensure balance

- use alignment to establish visual relationships between objects
- consider human optical adjustment
  - use the "squint" test to see if things you want to stand out, do stand out
  - don't make the eye wander all over back and forth across the screen)
- use "negative space"
  - also referred to as white space—i.e., space that is not what you want the user to look at, but space that helps separate items and clarify the visual elements in a design
  -

usability testing.

- heuristic evaluation (from [Johnson, 2006])
  - visibility of system status
  - match between system and reality
  - user control and freedom
  - consistency and standards
  - error prevention
  - recognition rather than recall
  - flexibility and efficiency of use
  - aesthetic and minimalist design
  - help users recognize, diagnose, recover rom errors
  - help and documentation
- assessment
  - formative assessment (during system development)
  - summative assessment (after system completion)

- location, audience
  - laboratory testing
  - pilot testing
  - field testing
- controlled experiments
  - control group
  - intervention group
- factors
  - human subjects: how many? demographics? those tested should match target audience for system
  - variables: dependent (variables that are measured); independent (variables that are manipulated; set a number of different values for testing)
  - hypothesis: what are you trying to prove? what outcome(s) do you predict?
  - experimental methods: between groups (randomized, one condition per subject); within groups (all subjects perform under all conditions)
  - *save all your data!*

references.

- Gibson, J. J. (1977). *The theory of affordances.* In R. E. Shaw & J. Bransford (Eds.), Perceiving, Acting, and Knowing. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Norman, D. A. (1990). *The design of everyday things.* New York: Doubleday.
- Johnson, Andrew (2006). Online course notes for *CS 422: User Interface Design & Programming*, Dept of Computer Science, University of Illinois at Chicago, `http://www.evl.uic.edu/aej/422/`, Spring 2006.