Comp 341/441 - Human-Computer Interface Design

Spring Semester 2017 - Week 12

Dr Nick Hayward

Users and Skills

skills change over time

- familiarity, experience, and comfort with an application often increase a user's skills
- skills tend to improve as follows
  - *improved awareness of the application's options, tools, and capacity*
  - *improved and increased awareness of how to perform tasks, handle special cases successfully*
  - *a much lower rate of errors, issues, and mistakes*
  - *increased rate of productivity and completion, speed, efficiency, and so on...*
  - *a general increase in confidence and greater ease at achieving a sense of flow with the application...*
- might also expect general improvement in quality of work
  - *quality often hard to define, measure, and assess*
  - *easier for procedural tasks and jobs than conceptual*

Users and Skills

practice makes perfect

- improve skills through regular practice
- for our applications and products
  - *ensure users practice and repeatedly perform given tasks*
- some application scenarios naturally make it easier for users to practice
- simple act of repetition of regular tasks often mimics regular practice
  - *practice due to necessity*
- *"people generally become skilled in whatever becomes routine for them."*
  - *Card et al. P.188. 1983.*
- **deliberate practice** is the act of intentionally practicing with focused attention
  - *specific goal of improving skill levels, working and training at increasing levels of difficulty*
  - *often requires careful monitoring and evaluation of work and results*
  - *motivation and self-improvement important*

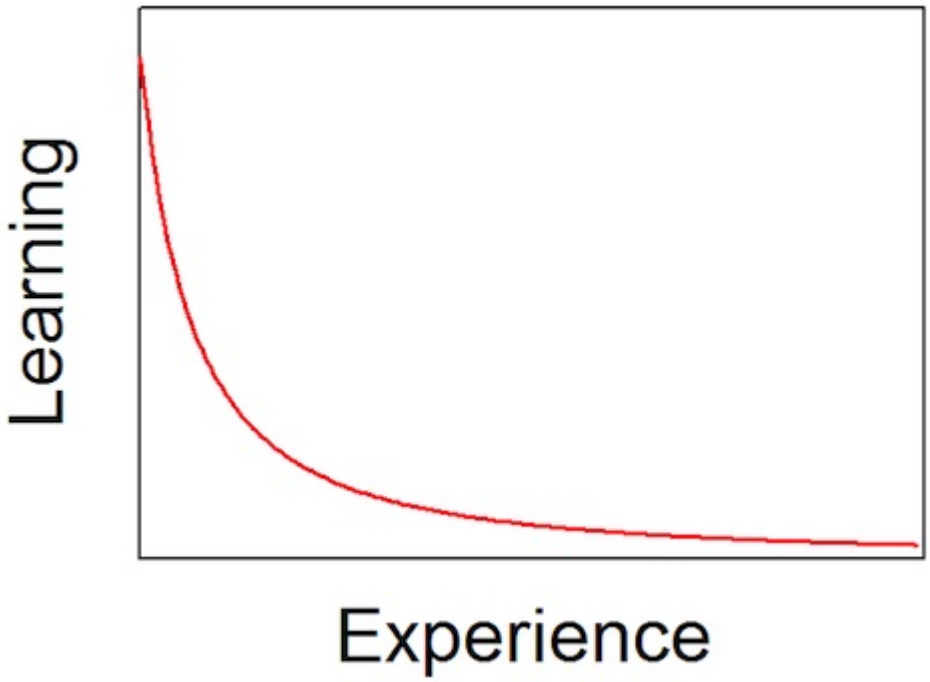
Users and Skills

monitoring practice and skills

- **Power Law of Practice** - Card et al. 1983
  - *applies to most mechanical and cognitive skills, not always relative to knowledge acquisition*
- as users gain in experience relative to increased practice
  - *related application performance tends to increase rapidly, then slow to a steady rate*
  - *steady peak normally reflects attained peak performance for the practiced skill*
- lack of practice naturally leads to loss of performance and skill
  - *drop in frequency and intensity of practice*
  - *motor skills do not normally atrophy as quickly as knowledge based skills*
  - *simple to refresh these skills with a period of further training and practice*
- designers need to be aware of this potential for skills atrophy
  - *complex, detailed applications may consider detailed help systems, options*
  - *allow a user to quickly refresh knowledge using practice exercises, tests, incentives...*

Image - Users and Skills

power law of practice



Power Law of Practice

Source - Wikipedia

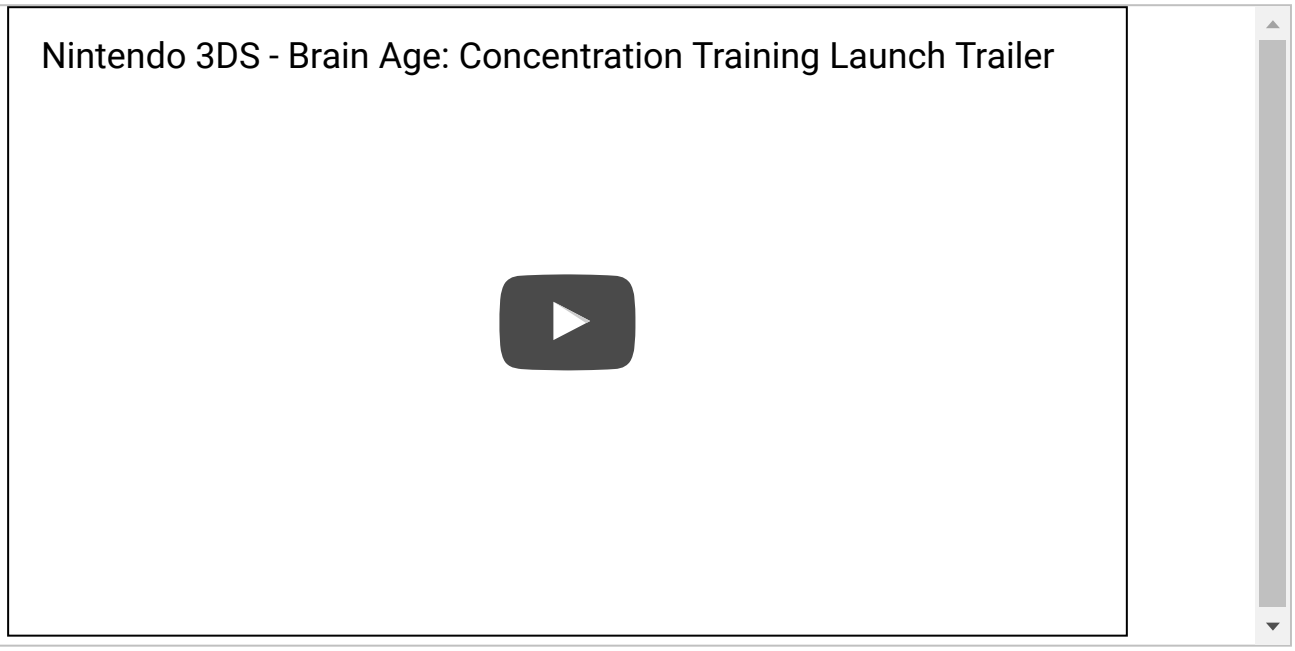
Users and Skills

gaining competence

- practice allows us to determine improvement relative to a given activity
- **four stages of competence** model suggested by Robinson in 1974
- this model suggests the following stages a user may follow to mastering a skill
  - ***unconscious incompetence***
    - user is unaware of how bad he or she may be relative to a particular skill
    - may even by unaware that the skill exists
  - ***conscious incompetence***
    - as user attempts a given skill, they become increasingly aware of a deficiency of skills
    - realise need to improve that skill through further training, learning, practice...
    - may be a daunting and overwhelming realisation for many users
  - ***conscious competence***
    - practice allows a user to engage in training sessions, exercises...
    - effectiveness of such training can vary greatly
    - often dependent upon task itself, suitability of chosen practice and training
  - ***unconscious competence***
    - complete a task without really thinking
    - act of working, completing an exercise has become natural to the user
    - do not really need to think about the given act...
- games are a good example of hands-on training and practice

Video - Users and Skills

Nintendo's Brain Age



Nintendo Brain Age: Concentration Training
Source: YouTube

Principles for Usability

intro

- consider some of the underlying design principles that help guide our designs
- eg: Don Norman's design principles for usability
  - *Norman, D. The Design of Everyday Things. 1988.*
- Norman introduced a set of basic design principles and concepts
  - *consistency*
  - *visibility*
  - *affordance*
  - *mapping*
  - *feedback*
  - *constraints*

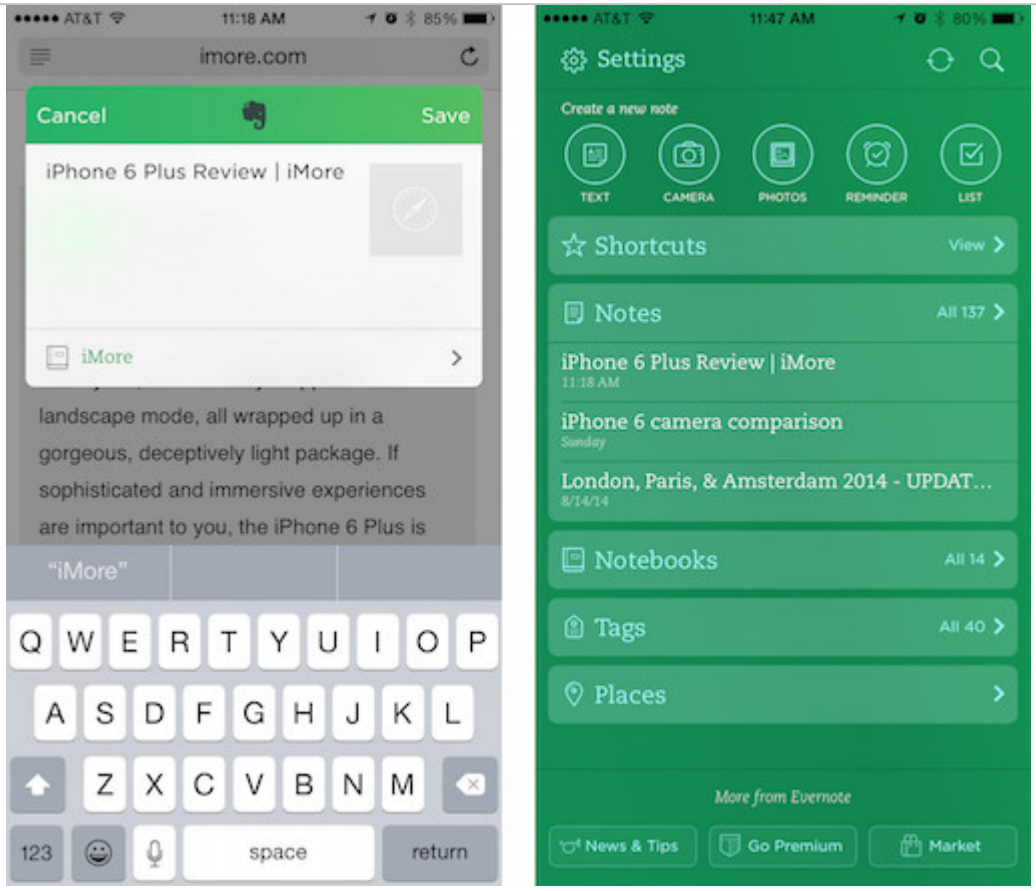
Principles for Usability

consistency

- one of the primary ways our users learn is by discovering *patterns*
  - *new situations easier to learn by reference to existing patterns of knowledge*
- *Consistency* is key in helping our users recognise and apply such patterns
- overall, things that look the same should perform the same general way
  - *same button, same colour normally infers same pattern of interaction and usage*
- behaviour and actions should also follow a similar pattern
  - *sound, animation, vibration etc should follow a similar pattern for users*
- design inconsistency can cause confusion and overload for our users
- memorisation of exceptions may also increase user resentment towards the app
- internal design and interaction consistency crucial for our users
- external consistency equally important and useful
  - *consistency between OS and app design guidelines*

Image - Principles for Usability

Evernote



Evernote for iOS 8

Source - Evernote

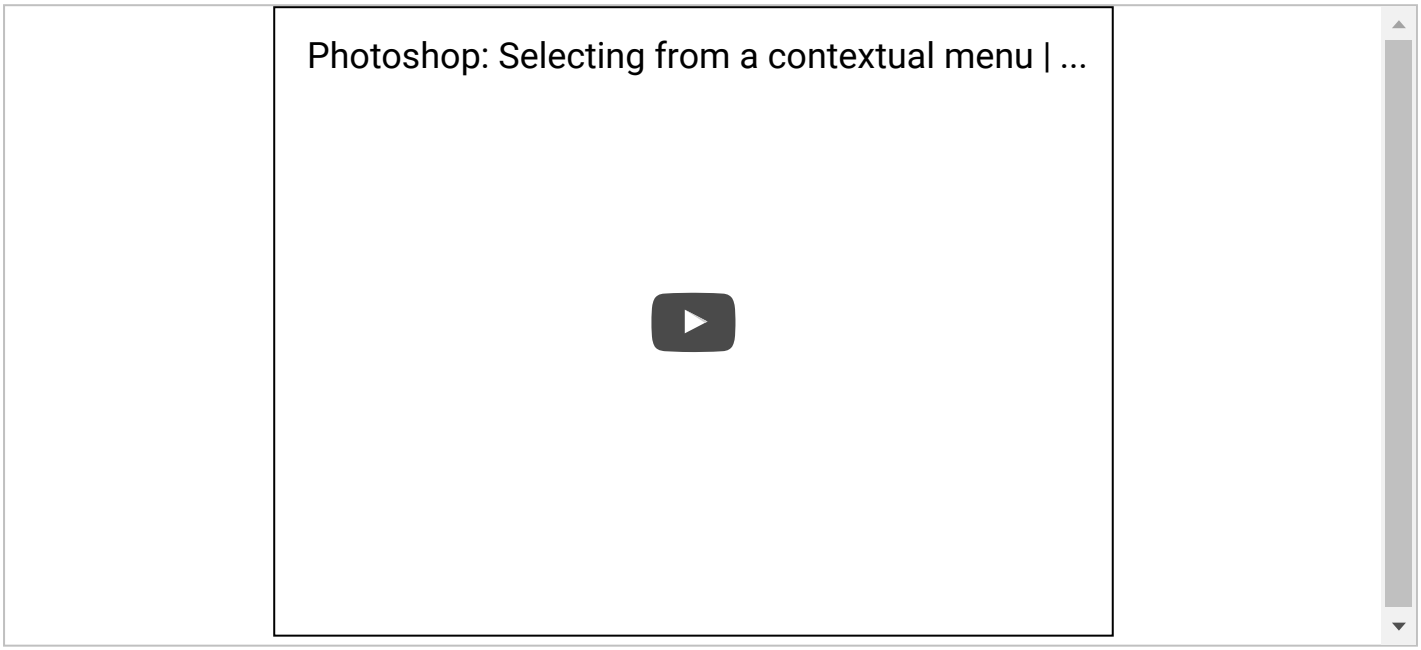
Principles for Usability

visibility

- users normally learn app functionality by visually inspecting the UI
  - *eg: available menus, menu items, icons, buttons, links, tools etc...*
- sequential tasks should be well labelled and navigation obvious
  - ***next** button obvious, and highlighted*
- usability and learnability naturally improved when options and commands clear and visible
  - *controls should be easily visible, contextually appropriate, logically placed*
- functionality within an application that is not visually represented often hard to discover
  - *keyboard shortcuts often a bad choice for sole command option*
  - *shortcut combinations often noted in visual menus*
- visibility does not, necessarily, infer that all options and functions be graphically represented
- impractical for many complex applications
  - *need for careful, considered design choices and contextual awareness*

Video - Photoshop

contextual menus



Photoshop: Selecting from a contextual menu
Source: YouTube

Principles for Usability

affordance

- a visual attribute or physical property of a given object or control
- gives the user clues to the operation or functionality of an object or control
- system parts manipulated to allow a user to interact with the given system
  - *eg: a door handle*
  - *shape of door handles, the nature of the door itself present clues to functionality*
- visual clues can be used to show UI element functionality
- eg: make controls, buttons etc appear clickable and ready for interaction
- add some highlight to show a user that a submit button is ready for a completed form
- design conventions developed for a reason
  - *offer a useful reminder of how patterns can easily be developed relative to a UI*
  - *blue underline for links on a web page*

Video - Principles for Usability

material design

Google's Material Design
Source: YouTube

Principles for Usability

mapping

- expected relationship between a performed action and the expected result
  - *mapping between a given control and its behavioural effect*
- such mappings should be logical, explicit, and straightforward
  - *descriptive labels, icons etc on buttons, menus...*
- controls should be positioned in a logical manner
  - *adhering to conventions where possible*
  - *many UI guidelines, real-world examples to help guide our design choices*
- modifications of expected conventions will cause unnecessary issues for users
  - *where necessary, reinforce with training and help...*

Principles for Usability

feedback

- plays a crucial role in reinforcing users' perception, expectations, general experience...
- principle of feedback states that designers should offer users confirmation or acknowledgement for the result of an action
  - *good or bad, successful or unsuccessful*
- distinguish two types of feedback
  - ***activational feedback***
    - provides evidence that a given control was actioned successfully.
    - eg: a button pressed, menu item selected, slider control moved to a new position
    - feedback may be offered visually, in a tactile manner for physical controls, an audible alert
  - ***behavioural feedback***
    - provides evidence an action etc has had an effect of the application, system...
    - eg: app closes an open, active window, shows a dialog window and status message, audible sound...

Video - Principles for Usability

material design

Google's Material Design
Source: YouTube

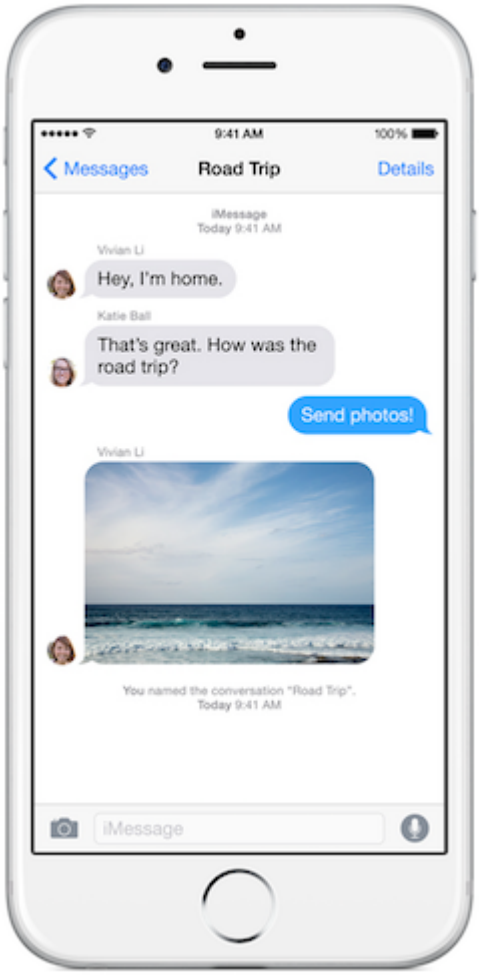
Principles for Usability

constraints

- apps and interfaces need to be designed and tested to prevent invalid states
  - *incorrect, invalid user interaction, invalid actions...*
- constraints may take various forms
  - *check correct relationships between elements and actions*
  - *check elements active only as needed*
  - *actions only performed when default data etc available*
  - *menu items active relative to contextual requirements*
  - *physical products often display such constraints*

Image - Principles for Usability

Message app on iOS



Messages for iOS

Source - Apple

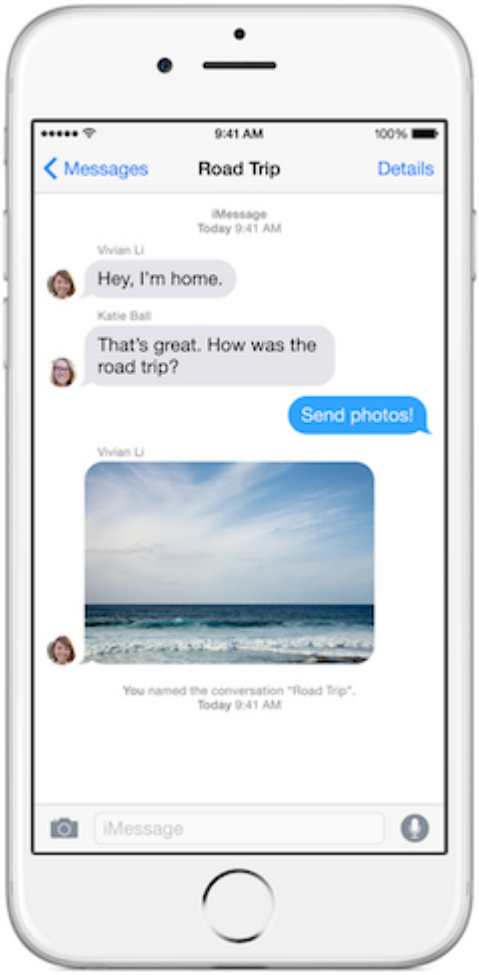
Principles for Usability

naming

- names and labels key aspect of human communication, thought, understanding...
  - *also an important consideration in design*
- naming helps users understand the application
  - *their current location relative to navigation*
  - *the data and information they are viewing*
  - *action they can and cannot perform...*
- good naming helps a user form a correct mental model
- do not confuse naming with the use of technical jargon and terms
- precise, consistent naming helps us form unambiguous instructions, help, feedback...
- naming helps identify as well as differentiate between aspects of the design and functionality
- names should be unique relative to the context and the application
- namespaces are useful relative to application design and development

Image - Principles for Usability

good(ish) naming

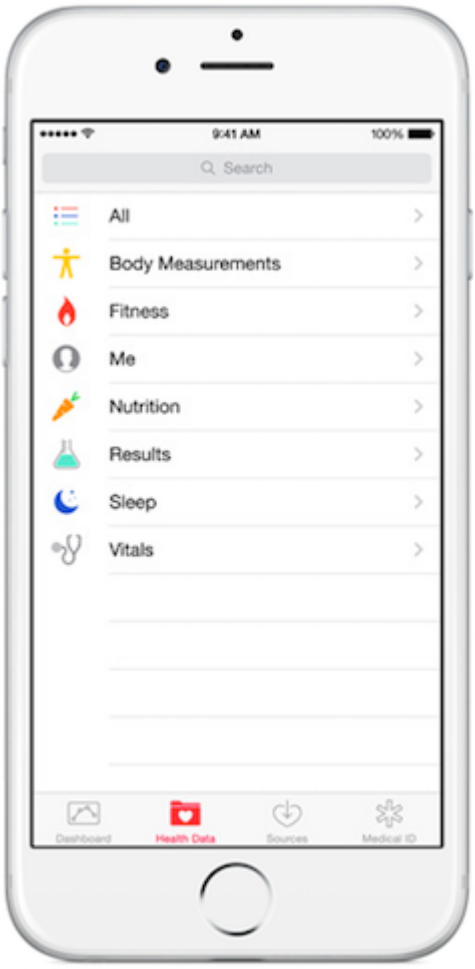


Messages for iOS 8

Source - Apple

Image - Principles for Usability

bad(ish) naming



Health for iOS 8

Source - Apple

Principles for Usability

naming guidelines - a few thoughts

- does the name accurately reflect and describe its intended target?
  - *consider the action of the element relative to the name*
- is the name clear, concise, and free of ambiguity?
- use concise, easy to remember names
  - *better than longer, hard to remember descriptions*
- does the name inherently assume prior knowledge from the user?
  - *consider naming relative to perceived domain knowledge*
- acronyms are useful, but assume prior knowledge of the domain
  - *be careful when using acronyms, and consider cultural bias*
    - eg: VAT well known in Europe
- carefully consider capitalisation, and ensure consistency for chosen pattern
  - *eg: This Is Capitalised...This is Capitalised...This is not Capitalised (fully)...*
- users should be able to pronounce a name...not helpful if they have to check first

Image - Principles for Usability

cultural naming concerns

Calpis Water	Pocari Sweat
	

Source: Calpis | Pocari Sweat

Image - Principles for Usability

bad naming and icon

iOS 6 icon	iOS 8 icon
	

References

- Carstens, A., and Beck, J. *Get ready for the gamer generation.* Tech Trends 49. PP.22-25. 2005.
- Cooper, A. et al. *About Face 3: The essentials of interaction design.* Wiley. 2007.
- Nielsen, J. *Heuristic evaluation.* Usability inspection methods. New York. John Wiley and Sons. P. 30. 1994.
- Tyldesley, D.A. *Employing usability engineering in the development of office products.* Computer Journal, Vol. 31. No. 5, PP. 431-436. 1988.