Comp 341/441 - Human-Computer Interface Design

Spring Semester 2017 - Week 11

Dr Nick Hayward

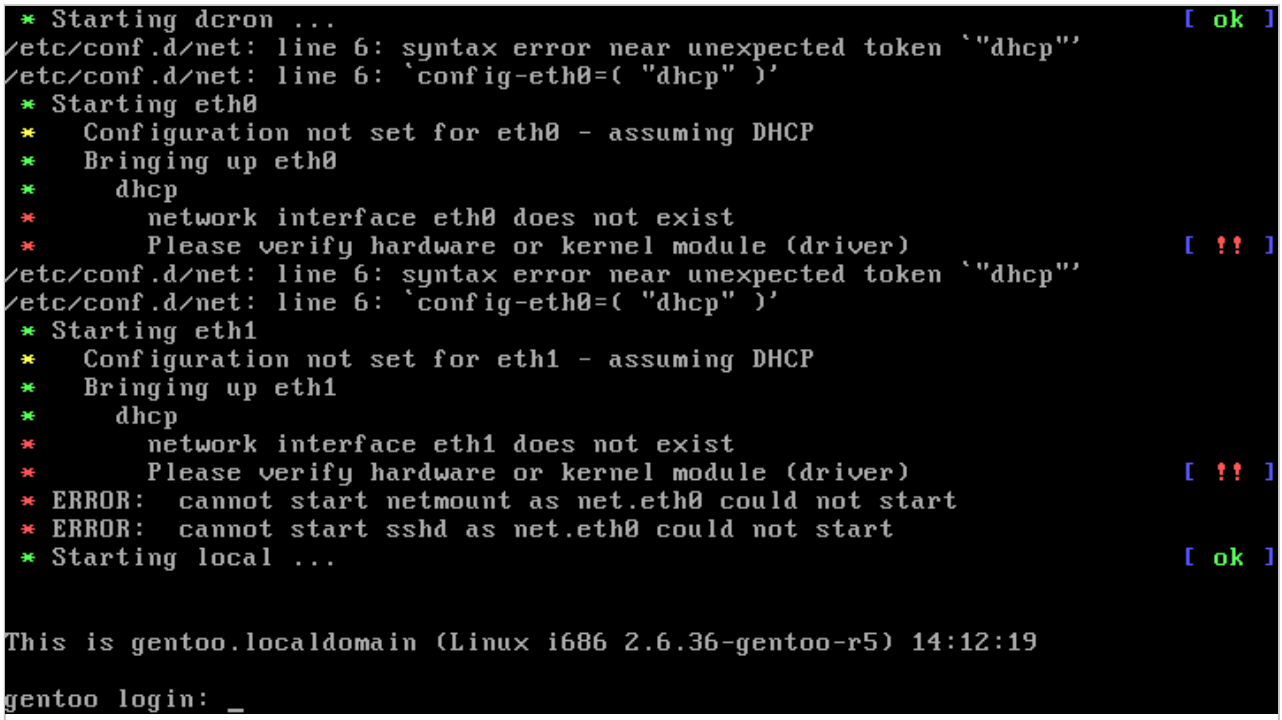
User Experience (UX)

overview - part 1

- broad and over-arching concept
- need to consider many disparate concepts
  - *user's reaction, both positive and negative*
  - *user's general experience with the application including*
    - design and interface
    - potential results and outcomes
  - *general functionality and what an application can do for a user*
  - *does the application, product etc solve a defined problem?*
  - *what can an application help a user to achieve?*
  - *what entertainment value does the application etc provide?*
- software application UX also influenced by acquisition
  - *was it easy to find, download, install, update?*

Image - User Experience (UX)

Linux installs



Gentoo Linux

Source - Gentoo Linux

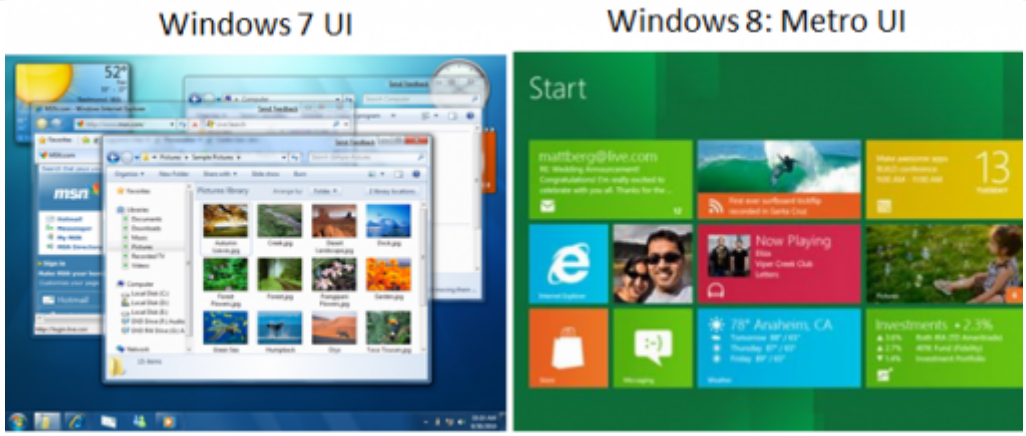
User Experience (UX)

overview - part 2

- user's identification of an **acceptable** product
  - *sense of usability and product preferences*
- Shackel, B. 1991.
  - *product's utility, usability, attraction relative to involved costs...*
- product considered not acceptable vast majority of users seek market alternatives
- UX inherently important aspect of goal to develop and provision successful application...

Image - User Experience (UX)

Windows



Windows

Source - Windows Comparison

Designing our app

considerations - part 1

- tasks and activities a user can and should be able to perform with the product
  - *ie: what is the considered scope of the product's functionality?*
- as we consider each task, how will the interaction develop and be processed?
  - *in effect, what are the expected steps and actions for the user and the product?*
- we need to consider carefully the overall visual style or appearance of the application
  - *eg: visual design and layout for the basic page templates or screen layout - fonts, colours, typography and iconography, any branding...*
- what are the defined **places** in our application?
  - *eg: pages for a website, navigation controllers and panels for mobile apps, levels in games, and so on...*
- how does our user actually navigate between these **places** within our application?
- as we consider further our app's places, what content and layout will be presented to the user in each *place*.
  - *which controls are available, how will they be presented, arranged, and so on?*

Designing our app

considerations - part 2

- how will the user interact with these controls?
  - *ie: just mouse and keyboard, is touch accepted?*
  - *are there behaviours associated with these controls?*
- are there any events within our application that are not triggered by the user?
  - *eg: timer driven events, remote calls and services, backup protocols, automatic updates...*
  - *are any behaviours actioned during such events?*
- does the application store, request, manage any data?
  - *what type of data, where, format, protocols, services...*
  - *how do we present this data on-screen and to the user?*
- is there a naming scheme for interface and interaction elements?
  - *eg: data, elements, places, objects, controls, navigation, and any other pertinent concepts...*

Designing our app

considerations - part 3

- error handling scheme for the app
  - *how will the user be informed? will the user have the option to gracefully recover from errors etc?*
- are there defined user roles in the app?
  - *what actions, privileges are permitted per role?*
- how do our users request or find assistance within the app?
  - *is it an active system or passive? ie: interactive or reference based documentation, tutorials, videos, discussion forums etc...*
- how is the app structured to promote app guidance for users through tasks?
  - *help for the users to work out how the app actually works...*

Designing our app

considerations - part 4

- need to engage in a number of related tasks
  - *eg: gathering requirements and their analysis*
- need to understand our user base, the target audience for our app
  - *includes their characteristics, requirements, how they intend to interact with the app*
- as designers and developers we will need to understand
  - *the type of work users want to complete*
  - *the inherent tasks*
  - *the effective problem domain*
- to a lesser degree, this will also require an understanding of the technology requirements
  - *eg: chosen languages, frameworks, device hardware...*
  - *impacts how and what we are able to design and provision for our users*
- need to consider prototypes, mockups, design documentation and specifications, and testing...

Users and Skills

intro

- continue to consider our application's users
- primary challenge involves consideration of product development relative to both beginner and advanced users
  - *how to make usable and productive app for all concerned*
  - *comprehensible and learnable for beginners*
  - *do not hinder expert users from optimal productivity*
- carefully consider user skill levels
- be aware of changes to skill levels over time
- aware of practical ways to help our users attain and improve skill levels
- understanding user's skill levels helps application of interaction concepts and principles

Users and Skills

user categorisation - part 1

- we can often categorise users by application skill levels and aptitude
- **evaluation user**
  - *testing and evaluating an app and not yet committed to its usage*
  - *trying to determine its suitability for their requirements*
  - *no pressing tasks or action at hand*
- **beginner user**
  - *trying to accomplish some tasks with the application*
  - *little or no prior experience with the app's usage*
  - *general feelings of uncertainty and learning by trial and error, general experimentation*
  - *some, but not all, will use the available tutorials, help documentation etc*

Users and Skills

user categorisation - part 2

- **intermediate user**
  - *more confident and experienced user, able to complete most of their required tasks*
  - *unlikely they will have explored all of the app's features and options*
  - *user comfort and fluency will not have been achieved for the application*
  - ***perpetual intermediates***
  - *Cooper et al. 2007.*
- **expert user**
  - *greater application confidence and certainty*
  - *awareness of product's domain and advanced options*
  - *able to complete tasks with ease, solving problems as they arise...*
- **power user**
  - *considered an extension of an **expert** user with a fascination of the application*
  - *normally enjoys customising the application and testing its limits*

Video - Usability

Users and skills

Your First Script - Apps Script Tutorials
Source: YouTube

Users and Skills

development of skills

- user classification is inherently a simplistic interpretation of skills acquisition and development
- many disparate factors influence development of skills. For example,
  - ***domain knowledge***
  - *assumption of underlying, pre-existing knowledge for a given application's scope*
  - ***general computing skills and knowledge***
  - *many applications assume general computing skills and knowledge*
  - *eg: simple ability to use similar applications*
  - *ability to use their chosen mode and tools of interaction*
  - ***general intelligence and reasoning abilities***
  - *an assumption of general reasoning and extrapolation skills*
  - *ability to read and understand help documentation...*
  - ***persistence, motivation, and dedication***
  - *some users will, of course, give up when faced with problems and challenges*
  - *others are more persistent and will try to solve a problem or issue*
  - *gamification and rewards may help this issue...*

Users and Skills

assumptions - part 1

- consider basic assumptions about users' minimum required skills and knowledge
- often dependent upon goals and functionality of the product, application...
- some inherent assumption of skills for your application
  - *eg: user will be able to use a keyboard, mouse, touchscreen...*
  - *basic level of verbal, reasoning, and mathematical knowledge*
- valid user testing important relative to such assumptions
- testing helps define and highlight unrealistic design choices and assumptions
- modify assumptions and design in response to testing feedback
  - *re-consideration and re-design may be necessary*

Users and Skills

assumptions - part 2

- assumption of Domain knowledge - Documenta Latina
- gaming and applications
  - *eg: Royal Game of Ur*

Royal Game of Ur

Source - Royal Game of Ur British Museum

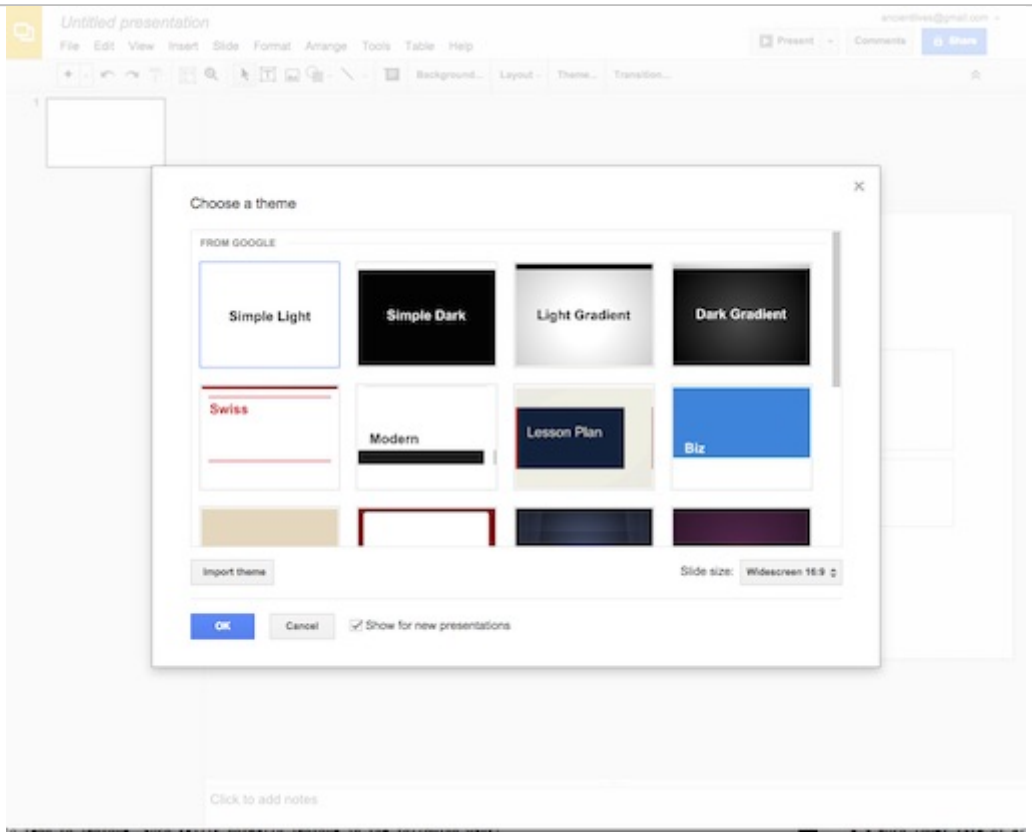
Users and Skills

skill levels and design - part 1

- **evaluators**
  - *design needs to present good first impression, be pleasing overall, and inviting*
  - *should not give the impression of being overly complex*
  - *introductory material, such as demo video or guided tour with step-by-step instructions*
  - *sample files, demo material allows users to test functionality and see what is possible*
- **beginners**
  - *functionally easy for our users to learn and discover an application*
  - *eg: offer wizard style guidance to create an initial project, document*
  - *easy undo/redo errors and mistakes - hopefully promotes experimentation in the app*
  - *in-depth tutorials and intro guides, such as manuals, help videos, online help*

Image - Users and Skills

getting started



Google Slides - Getting Started

Source - Google Slides

Users and Skills

skill levels and design - part 2

- **intermediate**
  - *in addition to the above considerations*
  - *fully indexed and searchable help resources*
  - *allow users to quickly find exactly what they need*
  - *online forums and social options and interaction promote sense of community*
- **expert**
  - *quick completion of tasks with maximum efficiency*
  - *provide shortcut options, keys, and greater customisation options*
  - *bypass and limit beginner tools, wizards, menus etc...*
- **power**
  - *allow greater freedom for users and interaction*
  - *user developed scripts, plugins, add-ons*
  - *developer tools, APIs, discussion forums, manuals...*
  - *carefully consider security implications*

References

- Card, S.K., Moran, T.P. and Newell, A. *The psychology of human-computer interaction.* Lawrence Erlbaum Associates. 1983.
- Robinson, W.L. *Conscious competency - the mark of a competent instructor.* Personnel Journal, 53. PP. 538-9. 1974.
- Shackel, B. *Usability - context, framework, design, and evolution.* Human factors for informatics usability. Cambridge University Press. PP. 21-38. 1991.