

## Notes - Design - Designing our app

- Dr Nick Hayward

A brief intro to general considerations for application and interface design.

### Contents

- Intro
- Heuristics
- Requirements and analysis

**Intro** Designing our application's interface is not quite as simple as choosing screen controls, arranging them, and then working out how a user will interact with them. The process is, naturally, slightly more nuanced and involved.

**Heuristics** However, we can perceive some heuristics to help guide our thinking and decision making for this design process.

So, let us now consider some of these key points.

As we design a product's interface, we normally have to think our way through the following salient points and considerations:

- tasks and activities a user can and should be able to perform with the product
  - i.e. 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
  - e.g. 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?
  - e.g. 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?
- how will the user interact with these controls?
  - i.e. 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?
  - e.g. 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?
  - e.g. data, elements, places, objects, controls, navigation, and any other pertinent concepts...
- error handling scheme for the app
  - how will the user be informed? will the user have the option to gracefully recover from errors &c.?
- 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 &c.
- 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...

**Requirements and analysis** To be able to fully consider and plan for each of these outlined points, we will need to engage in a number of related tasks. For example, gathering requirements and their analysis.

We will also need to understand our user base, effectively the target audience for our application. This will include their characteristics, their requirements, and how they intend to interact with the application. Therefore, as designers and developers we will need to understand the type of work they want to complete, the inherent tasks, and the effective problem domain.

To a lesser degree, this will also require an understanding of the technology requirements. Such technology choices, for example chosen languages &c., will have an impact upon how and what we are able to design and provision for our users.

We will need to consider prototypes, mockups, design documentation and specifications and, of course, testing.