I. Usability
II. The UI design process
III. Android style and design patterns
IV. Programming Android interfaces
V. Internationalization


4. **Interface Hall of Shame**
   http://interfacehalloffame.eu
   **QuickTime 4 Player critique**

---

**Index**

1. What’s a good UI ?

2. Learnability

3. Efficiency

4. Safety

5. Principles

6. QuickTime 4
**User interface in industrial design**

System by which people interact with a machine in order to achieve some goal. Includes hardware and software components. Provides a means of:

- input, to manipulate a system
- output, the system indicates the effects of the users’ manipulation

UI of a steam locomotive
User interface in computers

The user interface of a program refers to the information\textsuperscript{a} it presents to the user, and the control sequences\textsuperscript{b} the user employs to control the program to get some result.

\begin{itemize}
  \item[\textsuperscript{a}] graphics, text, sound \ldots
  \item[\textsuperscript{b}] keyboard keystrokes, motion and selections of mouse, touchscreen \ldots
\end{itemize}

Toilet control panel
What's a good UI?

**Definitions**

**User-centred design**
- Gives extensive attention to the needs, wants, and limitations of end users of a product, instead of forcing the users to change their behaviour to accommodate to the product.
- Requires to analyse and foresee how users are likely to use a product, but also to test them in real world tests with actual users.
- UCD + user emotions, feelings = user experience design (UX)

*The question*: A user interface is well designed when . . .

Imagine a badly designed interface, like Campus Virtual: superfast, mail, unpublished news, lots of unused features, students list updating button . . .
The psychology of **helplessness**: the feeling that you cannot control your environment.

- The more you feel that you can control your environment, and that the things you do are actually working, the happier you are
- the UI changes users mood: frustrations and successes, big and small, add up
- Quantity matters more than quality
- The UI has to let users *feel like* they are in control: behaves in the way users *expect* it to behave

---

**Answer**

A user interface is well-designed when the program behaves exactly how the user thought it would.

The best UI lets the user to concentrate in his/her task, doesn’t deviate attention toward the UI.

All other principles are just corollaries.
What's a good UI?

This means, specifically, a good UI is

- fast and easy to use
- easy to learn, or there's even no need to learn at all!
- easy to remember after a time not using it
- safe: few interaction errors and recoverable
- pleasant and even fun

**Usability**

| Efficiency + Learnability + Memorability + Safety + Satisfaction |

So designing a good UI is any difficult? **YES!**

- We know how to communicate to other software engineers, who are like us
- UI design is about how to communicate to users who are NOT like you, “you are not the user”
- The user is always right: usability problems are the designer’s fault
- How to know how the/each user expects the program to behave?
- Users are diverse: different needs, training, attitude...
- There are many possible choices and decisions to make
- Long history of failures

We need to know in detail UI design **principles, patterns** and a building **process**.
Ok, ok, but is this so important?

- “As far as the customer is concerned, the interface is the product”
- UI design, coding and testing has a high cost: \( \approx 50\% \) of development effort
- “You can use an eraser on the drafting table or a sledge hammer in the construction site”
- “A good UI is not noticeable, the memory of a bad one lasts for years, if you are compelled to work with it”

We’ll focus on what makes an UI

<table>
<thead>
<tr>
<th>easy to learn</th>
<th>fast to use</th>
<th>safer</th>
<th>pleasant</th>
</tr>
</thead>
<tbody>
<tr>
<td>mental models</td>
<td>shortcuts</td>
<td>error prevention</td>
<td>color</td>
</tr>
<tr>
<td>metaphors</td>
<td>defaults</td>
<td>undo</td>
<td></td>
</tr>
<tr>
<td>affordances</td>
<td>aggregates</td>
<td>messages</td>
<td></td>
</tr>
<tr>
<td>self-disclosure</td>
<td>user-centered</td>
<td></td>
<td></td>
</tr>
<tr>
<td>consistency</td>
<td>short-term memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>feedback</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recognition vs recall</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Learnability: how to learn

How does a user learn to use an UI?

a) Taking a short course, seminar
b) Reading the manual
c) Reading the on-line help
d) Looking at others to use it

None of them. In general, the user learns by doing

- has a task to fulfil
- explores the interface to achieve his goal
- in case of trouble goes to the on-line help
- or a search engine! “word how to make an index”

Advise: on-line help must answer specific ‘how to’ questions

Learnability: recognition vs recall

- **Recall**: remembering something with no help from the outside world
- **Recognition**: remembering something with the help of a visible cue. Recognition is far, far easier than uncued recall.

List details of entries in a folder in reverse order
Learnability: types of interfaces

Command language: recall

![Terminal screenshot](image1)

---

![Mathematica screenshot](image2)
Learnability: types of interfaces

Command language: recall

Self-disclosure: technique for making a command language visible, helping the user learn the available commands and syntax.
Learnability: types of interfaces

Menus and forms: recognition, far more learnable
Learnability: types of interfaces

Direct manipulation: the user interacts with visual representation of data objects
Learnability: types of interfaces

Direct manipulation:

- **continuous visual representation** (not on demand) of application data objects: icons (files, folders), text (in a word processor), mail messages (webmail client)
- interaction through physical actions: click an object, drag, handle selection (e.g. window resize), gestures (mobile)
- plus **buttons, keystrokes**... which specify commands
- the effect of actions is **immediately visible**, incremental (drag), reversible
- successful because is more natural: exploits perceptual and motor skills (first thing we learn), not linguistic skills like in command, menus/forms

Learnability: models

Remember

*A user interface is well-designed when the program behaves exactly how the user thought it would.*

Every user will build a mental model of how the application works.

*When I was 6 and my dad brought home one of the world’s first pocket calculators, an HP-35, he tried to convince me that it had a computer inside it. I thought that was unlikely. All the computers on Star Trek were the size of a room. I thought that there was just a clever correlation between the keys on the keypad and the individual elements of the LED display that happened to produce mathematically correct results. (Hey, I was 6).*

Learnability: models

System model is a way of describing how the system works: what the parts of the system are, and how they interact to make the system do what it's supposed to do.

Models in software

- **user / mental / conceptual model**: how the user *thinks* the system works
- **system model**: how the system *actually* works

If

- Conceptual model $\subseteq$ System model: easy to learn
- Conceptual model $\neq$ System model: user errors

The UI has to communicate the system model as much as possible.

At a high level, the user model of Twitter is

- that there are other users in the system
- you have a list of people that you follow
- and a list of people that follow you
- each user generates a stream of tweets that are seen by their followers
- tweets are mixed together into a feed
Learnability: models

Alt + Tab in LinuxMint 13: most users assume it simply rotates among all available windows because often an application has a unique window.
Learnability: models

Back button in a web browser

What’s user and system model? Do they match?

- **System model**: go back to the last page the user was viewing according to browser History
- **User model**: “take me to the last screen I’ve seen”

If a website has pages of its own (results of search engine, pages of article) the two histories are different. Previous \( \neq \) Back buttons.

Edit a text in Powerpoint and Paint: text is just pixels in a layer or a graphic primitive?
Learnability: consistency

Why is so easy to use Microsoft PowerPoint, LibreOffice Impress, LibreOffice Write, . . . once you have learned Word (or conversely) ?

Why is so easy to use LibreOffice Calc, Gnome GNumeric once you have learned Microsoft Excel ?

Consistency

Similar things should look the same. Same actions should produce the same result. Different things should look different, different actions produce different results.

Consistency is fundamental to learnability: allows the user to transfer his/her knowledge to a new UI.

“Sometimes you don’t need to reinvent the weel”
## Learnability: Consistency

### Internal: within one same application

- **Gmail**
  - **Mail**
  - **Contacts**

### External: with other applications

<table>
<thead>
<tr>
<th>GIMP</th>
<th>Photoshop</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="GIMP Tools" /></td>
<td><img src="image2" alt="Photoshop Tools" /></td>
</tr>
</tbody>
</table>

- **Rectangle Select tool (R)**
- **Free Select tool (F)**
- **Color Select tool (SHIFT+O)**
- **Foreground Select tool**
- **Color Picker tool (O)**
- **Measure tool (SHIFT+M)**
- **Align tool (Q)**
- **Rotate tool (SHIFT+R)**
- **Paintbrush (Brush) tool (B)**
- **Eraser tool (SHIFT+E)**
- **Pencil tool (N)**
- **Burn (Dodge) tool (SHIFT+D)**
- **Foreground (FG) color**
- **Reset FG & BG to black & White**
- **Ellipse Select tool (E)**
- **Fuzzy Select (Magic) tool (U)**
- **Scissors Select tool (O)**
- **Paths (pen) tool (B)**
- **Zoom tool (Z)**
- **Move tool (M)**
- **Crop tool (SHIFT+C)**
- **Scale tool (SHIFT+T)**
- **Text tool (T)**
- **Blend tool (L)**
- **Airbrush tool (A)**
- **Clone tool (C)**
- **Perspective (Clone) tool**
- **Smudge tool (S)**
- **Swapping FG & BG colors (x)**
- **BackGround (BG) color**

**Additional Tools:**
- **Mont tool (V)**
- **Marquee selection tools (MR)**
- **Layers**
- **Quick selection/Magic wand (W)**
- **Crop tool (C)**
- **Shift tool select tool (O)**
- **Clone Stamp/Pattern Stamp (S)**
- **History (Alt+History brush (Y)**
- **Eraser (Regular/Magic Eraser (E)**
- **Paint Bucket/Gradient (G)**
- **Blur/Sharpen/Smudge (R)**
- **Dodge/Burn/Sponge (X)**
- **Pen tool (P)**
- **Type tools/Type mask tools (T)**
- **Path Selection/Direct Selection (A)**
- **Shape tools (I)**
- **Annotation tools (N)**
- **Eyedropper/Color sampler/Magnify/Count (I)**
- **Hand tool (H)**
- **Zoom tool (Z)**
- **Exchange color (E)**
- **Foreground color**
- **Background color**
- **Standard display/Black/white display mode (Q)**
- **Screen display modes (F)**
Learnability: consistency

Metaphorical: with some real, physical object

![mate-calc linux calculator](image)

Learnability: metaphors

**Metaphor**

User interfaces that look, are manipulated and give feedback like real objects so that the user can transfer his/her knowledge on them and need to learn (almost) nothing.

However, at some point metaphors must deviate from the real thing.
Learnability: metaphors

What’s the most spread metaphor?

Your computer screen is like a **desktop**.

bumptop.com 3d desktop (2010)
http://www.youtube.com/watch?v=MOODskdEPnQ

too far fetched?
Another popular metaphor: document editor like writing, drawing and gluing photos on a sheet of paper.

What do you think of this old Windows calculator?
Learnability: metaphors

Follows closely its metaphor, only $\times$ for $\times$ and $\sqrt{}$ for $\sqrt{}$ but bypasses opportunities for improvement

- just one line of display (there are calculators with paper tape)
- why only one memory slot? why show $M$?
- invisible modes: after entering digits and one operator, clears the display
- cryptic buttons $MC$, $MR$, $MS$, $M+$ (recognition, not recall)
- is it possible to type in keyboard? no clue like blinking underscore cursor
- blue/red, small text in buttons: hard to read

Learnability: affordances

What have in common all these objects?
Learnability: affordances

**Affordance**

Non-verbal property of well designed objects that make it clear how to operate them just by looking at them.

They are learned from experience. Well designed UI exhibit also perceptual (visual) affordances.
What's a good UI? Learnability Efficiency Safety Principles

Learnability: affordances

What can you do with this page?

Create a bookmark with this address:

```javascript
var result = document.evaluate("//text()", document.body, null, XPathResult.UNORDERED_NODE_SNAPSHOT_TYPE, null);
for (var i = 0; i < result.snapshotLength; ++i) {
    var node = result.snapshotItem(i);
    if (node.textContent.match(/\w/) && node.parentNode.nodeName != "STYLE") {
        node.textContent = node.textContent.replace(/[A-Z0-9]/g, "X").replace(/[a-z]/g, "x");
    }
}
void 0
```

visit a web page and then the bookmark.
Learnability: feedback

Feedback

Visual and auditive changes of interface when the user performs some action, as a way to acknowledge or respond to that action.

Every user action must have a feedback:

- push buttons seem to depress and release
- menu options are highlighted
- pressing a key makes a character to be printed or cursor moves
- scrollbars, dragged icons move with cursor
- cursor changes its shape when moving over some controls . . .

But how fast must the feedback be? and what if the system can not respond that fast?

Human perceptual system works in cycles of 50–200 milliseconds: two stimuli closer than that seem to happen simultaneously.

Feedback provided in

- $< 0.1$ seconds seems instantaneous
- $0.1 – 0.5$, 1 second: user notices the delay but maybe ok
- $1 – 5$ seconds: display some “busy” indicator
- $> 5$ seconds: progress bar allowing to estimate completion time approximately (1’, 10’, 1 hour, 10 hours, more does not make sense!)
Efficiency: short-term memory load

Short-term memory

Also called working memory,

- stores what we have just perceived
- is where conscious thinking happens
- opposite to long-term memory, very limited capacity: a few (7 ± 2) ‘chunks’, units of perception
- short lived, ≈ 10 secs.

Demostration:
http://faculty.washington.edu/chudler/puzmatch.html

Chunking: organize items in small groups to improve ability to remember.

Consequence: reduce short-term memory load, the UI should not place excessive demands on short-term memory.

How ? For example

- generic techniques: recognition better than recall, feedback, metaphors
- avoid modal dialogues that block the screen visibility
- don’t force the user to remember things the computer can or knows
- present information in chunks (e.g. file size, structured layout)
- direct manipulation > menus > than shortcuts and commands
- visual cues: tell users where they are, what they are doing, and what they can do next
- avoid long sequences of actions with wizards
- provide shortcuts
What's a good UI? Learnability Efficiency Safety Principles QuickTime 4

**Efficiency: shortcuts**

**Shortcut**

Some way to shorten the number of keystrokes or mouse actions a user needs to perform common actions. Reduces users’ memory load when become automatic.

Can you remember different types of shortcuts?

- key sequences like Ctrl + key, Alt + key, F11 . . .
- menu bars
- up, down arrows to recover previous commands
- web browser bookmarks
- file explorer favourite folders or bookmarks
- File → Open → recent files
- fill in forms with default values like today’s date, username . . .
- auto-completion when typing text, addresses
- aggregation: selection of multiple items for action like drag and drop, open, delete . . . multiple files
Safety concerns preventing and handling “user errors”. Don’t put the blame on the user but on the interface design.

There may be multiples causes for errors. We’ll focus on

- errors caused by modes
- prevention by confirmation
- how to make good error messages
Safety: modes

Mode
States of the interface in which the same user action has different meanings, results. An error occurs when the user misses the right state.

Examples:
- Caps-Lock key, Insert-Overwrite in text editors
- selection of file icons by mouse click or Shift + click

Advice:
- eliminate modes if possible, if not,
- make prominent the feedback of mode state
- use spring-loaded modes: the user has to do something to stay in mode, like press Shift key to write capitals or press first mouse button when dragging

Safety: confirmation dialogs

Use sparingly:
- is disruptive, efficiency loss
- avoid for frequent actions, if not,
- will become part of the action itself, user won’t read it
- better offer undo
What's a good error message?

- describes the problem/causes in a precise way
- the user can understand it, no technical wording
- avoid hostile and out of context terms: aborted, bad, fatal, error, illegal, invalid, violation, default
- constructive: offers a possible solution to the problem
- doesn’t blame the user
- do not associate them with sounds
Golden rules

Also called principles and guides of user interface design, are

- a few properties every good UI should have
- generic: don’t tell you *specifically* what to do or how
- this gap is filled by UI patterns
- most rules already covered by previous usability concepts
Golden rules

Shneiderman's 8 golden rules
http://www.cs.umd.edu/~ben/goldenrules.html:

1. **Strive for consistency** in actions, terminology, aspect.
2. **Enable frequent users to use shortcuts** like abbreviations, function keys, commands.
3. **Offer informative feedback** for every user action.
4. **Design dialog to yield closure**: sequences of actions should be organized into groups with a beginning, middle, and end.
5. **Offer simple error handling** so the user cannot make a serious error and offer mechanisms for handling them.
6. **Permit easy reversal of actions** to encourage exploration of unfamiliar options.
7. **Support internal locus of control**: users feel in charge of the system and that the system responds to their actions.
8. **Reduce short-term memory load**.

Mandel's golden rules, *The elements of user interface design*, T. Mandel, Wiley 1997:

1. **Place users in control**
   - Use modes judiciously
   - Display helpful messages
   - Provide immediate and reversible actions, and feedback
   - Accommodate users with different skill levels
   - Make the user interface transparent
   - Allow users to directly manipulate interface objects
2. **Reduce users' memory load**
   - Rely on recognition, not recall
   - Provide defaults, undo, and redo
   - Provide interface shortcuts
   - Use real-world metaphors
   - Progressive disclosure of functionality
3. **Make the interface consistent**
   - Maintain consistency within and across products
   - Provide aesthetic appeal and integrity
   - Encourage exploration
Golden rules

“Tog’s First Principles”, 16 principles from Bruce Tognazzini

http://www.asktog.com/basics/firstPrinciples.html

Translated to Spanish at
http://galinus.com/es/articulos/principios-diseno-de-interaccion.html

Interface hall of shame: QuickTime 4

open document critiqueQuickTime4Player.pdf