

I Are What You Eat

UI is an abbreviation for User Interface — a computer's UI is everything you see when using a computer. The Mac has a Graphical User Interface (GUI, oftentimes pronounced "gooey") which utilizes windows, icons, menus, and a pointing device. Text-input systems, like MS-DOS, in which the user types all instructions (in stark contrast to the much easier GUI approach), are called command-line interfaces.

A computer's User Interface is a virtual environment, just as an office or home is a real-world environment. We spend a great deal of time in this virtual environment: doing work, playing games, communicating, and otherwise accessing information. Because so much time is spent in this environment, the smallest details take on greater significance, and even tiny mistakes in interface design become magnified into glaring annoyances.

The Mac OS UI is often held up as a paragon of good human interface design. From its very beginning, the Mac has benefitted from a clearly defined set of Human Interface Guidelines, ensuring that all Macintosh applications behave in a way that is consistent, predictable, and sensible. Software developers ignore the Human Interface Guidelines at their peril!

Speaking of which, Microsoft in particular has a long history of cavalierly ignoring the Human Interface Guidelines in their Macintosh applications, and Microsoft's operating systems show a similar disregard for creating a pleasant and consistent user experience. Although Windows '95 was touted as being as easy-to-use as the Mac OS (and it was a distinct improvement over Windows 3.1), the Windows user interface is hampered by small mistakes and illogical choices which combine to create a distinctly unfriendly (but semi-usable) environment.

By comparing and contrasting the Mac OS and Windows, we can learn a

great deal about how the User Interface can either enhance or detract from the process of using a computer.

Keys to the Kingdom

The three key elements of good interface design are predictability, information, and simplicity. Remember these.

Predictability comes from the consistent behavior of interface elements like windows and menus. For instance, menus always drop down with a click (or click-and-hold) in every application, even though the commands they contain are different. Many keyboard shortcuts are the same in every app — Undo, Cut, Copy, Paste, Save, and Quit are nearly universal. By ensuring that programs behave in a way that is consistent and predictable, the Mac lets users feel confident that they can perform basic functions in any application.

Information must be displayed as efficiently as possible, balancing the need to have all relevant data readily available with the need to conserve screen real estate and avoid a cluttered or overly-complex appearance.

Simplicity should be the guiding principle in all UI design. Never make the user perform two actions when one will suffice. Never ask the user things they aren't likely to care about. Avoid adding so many features that it becomes difficult to learn how to do even the simplest tasks.

Screen Layout and Prioritization

Apple decided that information should be displayed and positioned on the screen according to its importance. Immediate actions always appear in the center of the screen: dialog boxes, alerts, status indicators, etc. The Mac's screen is organized in the way our Western culture reads and writes: we begin at the top on the left-hand side, move to the right and then downward. The Apple, File, and Edit menus are given prominent status, as they contain the most important and often-used commands. Moving to the right, less frequently-used menus like Help and the application menu remain out of the way until they're needed. Microsoft, on the other hand, decided to do just the opposite, probably just to be different from Apple (and avoid further look-and-feel lawsuits). The Win95 Start menu is located at the bottom left of the screen, and moving upward, the first menu item is

for shutting down the computer!

Apple puts their menubar at the top of the screen, so every menu can be accessed from the one centralized location. Microsoft puts menus at the top of each and every window; when many windows are open, it becomes more difficult to find the right menu, and some windows are likely to cover the menus of other windows.

olor

An interesting difference between Mac and Windows is the choice of default cursor color, as seen in this example:

ecause most pages are white with black text, Apple made the arrow cursor black so that it had high contrast against the white background. The black arrow has a white border so it can be still be seen against dark backdrops. When Microsoft replicated Apple's cursor bit-for-bit they inverted the colors — the resulting cursor has less contrast in most situations, and is consequently much harder to track.

'll Tell You Where To Stick Your Buttons

Microsoft appropriated some concepts from MacWrite (embedding a ruler in a document) and MacPaint (tools/palettes) and seemingly went berserk, so that Microsoft applications are littered with "chicklet" buttons, tiny buttons with no clear visual clue as to their function. Commands easily found and understood in menus are instead embedded in a visual cacophony of confusing little buttons, each wasting precious screen real-estate. This can

confuse new users (as well as make life more difficult for experienced users) and increases their chance of making serious errors.

Having some "chicklets" is not necessarily a bad thing; AppleWorks (originally ClarisWorks) handles buttons reasonably well. But not every command or function should be accessible through a button. A toolbar with a few buttons for the most frequently used functions is clearer and easier to use than one filled with dozens of buttons that perform every conceivable task. Buttons should only be used if their effects are visually obvious — if no result is immediately apparent, the user might click again and again, and wonder if they clicked at all. Buttons should never be used for "destructive" actions (eg. deleting files), since accidental mouse-clicks are not uncommon.

À La Carte

Since menus are used so frequently, it could be argued that they are the most important single element in the User Interface. Using menus is the most common way of issuing "verbs" (commands) to a program.

A menu is a group of commands organized by topic. The user selects a command (menu item) from the list, and an action is performed. For instance, the File menu contains a list of actions that can be performed on files (Open, Close, Print, and so on), while actions relating to text editing (Cut, Copy, Paste) reside under the Edit menu. Such a simple concept. Things become more complicated because there are so many details and variables to take into consideration.

Our preferences for menu behavior are formed by what we become used to. One can get used to any sort of inconvenience (even to the point of thinking of it as "natural"), but that doesn't mean it is a necessary fact of life, or that there might not be a better way of doing things. In fact, it was just this type of reasoning that led to all the DOS/Unix people ridiculing the GUI for the first few years of its life.

Some menu items have "sub" menus, so that related menu items can be more logically grouped together, and to prevent menus from becoming too long to be displayed efficiently on the screen. Hierarchical menus are usually denoted with an arrow (showing that more information is available).

It is generally not recommended to have more than one level of hierarchy because it becomes too hard to find things, and becomes painful for the user to navigate all those levels. In Windows 95, for instance, the Start menu contains so many sub-menus that trying to choose the right sub-menu can be like playing Whack-A-Mole!

Peeking in Your Windows

Windows are such an important element of the User Interface that Microsoft named their whole Operating System (shell) after them, forever confusing the terminology and users in a blatant attempt to take credit for concepts developed decades earlier. Microsoft got their windows working properly only 10 years after the rest of the industry had already adopted them, but you have to admire their chutzpah — come late to the game, implement poorly, and take credit for everyone else's work. But rather than continuing to bash Microsoft (as much fun as that is), this section looks at the similarities and differences in window behaviors between platforms, and what is good and bad about each. Sadly, so much is bad (interface-wise), about Microsoft's implementation of windows that it might seem to be Windows-bashing anyway.

A window is a container for displaying information on the screen. It allows the system to overlap different information containers as if they were separate sheets of paper. Screen real estate (the amount of visual space available) is a very important issue, so the idea is to allow users to quickly resize, relocate, reorder, close, and otherwise manipulate these windows.

Apple's Windows

Work at Xerox's Palo Alto Research Center (and earlier work by Doug Englebart, the inventor of the mouse) defined the first steps for windowing technology, such as the basic concept of overlapping windows. But Apple helped define almost everything else we use in this metaphor. Apple originated the titlebar, the grow box, the close box, and most of the direct manipulation (dragging windows around to move them, resizing behaviors, etc.), first for the Lisa and then the Macintosh. Apple's implementation of windowing is clear, efficient, and easy to use.

On the Mac, each window is a separate entity from other windows, and each window can be moved independently.

Palettes are bound to applications (and to individual document windows within the application). When you activate an application, the palettes of

other applications hide themselves, and the active application's palettes appear. This reduces palette clutter.

Windows are grouped by application. When you pull an application to the front by selecting one of its windows, all of its windows come to the front.

Microsoft's Windows

Microsoft tends to borrow from others, taking whatever seems good and adding it to elements assimilated from other systems (resistance is futile). Microsoft's windowing comes from older text-based windowing, X-Windows (Unix's windowing), and from the Mac OS. All this cross-platform borrowing results in an inconsistent windowing system.

The first versions of MS-Windows used tiled windows instead of overlapping windows. In fact, Bill Gates himself is reported to have said with regard to tiled windows, "That's not what a Mac does. I want Mac on the PC, I want Mac on the PC."

In MS-Windows, there are parent windows, and child windows. Child windows are bound to their parent windows, and embedded inside them. When you move a parent window, the contained child windows move as well. You can move a child window anywhere inside of the parent window you want, but it will be cut off by the edges of the parent window. Fortunately, in Windows 95 and later revisions, Microsoft improved parent-child windowing, and made their windows more Mac-like. Mac on the PC, indeed.

n Conclusion

One should not infer from this article that the Mac's User Interface is perfect. Consistency problems, or interface "bugs," exist in every system, and there's always room for improvement. But there are differences in degrees of usability as well. Whereas Windows achieves some level of usability by being "good enough," the Mac's simplicity and elegance inspire loyalty and devotion to the platform, because good UI design is the very essence of the Macintosh experience.

rateful Acknowledgment

This article was inspired by and adapted from David K. Every's essays on

User Interface. For further readings about human-computer interaction issues, the interested reader would do well to seek out the "Interface" section of David's MacKiDo Temple at <http://www.mackido.com/Interface/> . Many thanks to David for his permission to adapt his work for this article.

Brian Kelley

brian@apple wizards.net

<http://apple wizards.net/>