Principles of good GUI Design

Graphical user interfaces (GUIs) have become the user interface of choice. Yet despite the GUI's popularity, surprisingly few programs exhibit good interface design. Moreover, finding information explaining what constitutes a good and intuitive interface is exceedingly difficult. In this article, I describe the basic rules for all good interfaces -the cardinal dos and don'ts.

However, before starting in on what constitutes good design, I need to explain the causes of bad design. This way, if you are tempted to deviate from the tried and true, you'll know where the wrong path leads -and, I hope, get back to good design.

Forgetting the User

Developers often design for what they know, not what the users know. This age-old problem occurs in many other areas of software development, such as testing, documentation, and the like. It is even more pernicious in the interface because it immediately makes the user feel incapable of using the product. Avoid this error diligently.

Give Users Control

GUI designers' predilection for control is evident in applications that continually attempt to control user navigation by graying and blackening menu items or controls within an application. Controlling the user is completely contradictory to event-driven design in which the user rather than the software dictates what events will occur. As a developer, if you are spending a lot of time dynamically graying and blackening controls, you need to re-examine your design approach and realize that you may be controlling the user, who may not want to be controlled. As business changes at a faster pace, flexibility in user interfaces will become a key enabler for change. Allowing the user to access the application in ways you never dreamed can be scary, but satisfying for you as a developer and empowering for the user.

Too Many Features at the Top Level

Examine a VCR built in 1985 and then examine one built in 1995. You will see a startling difference in the interface of the two models. The model built in 1985 will have an abundance of buttons readily available on the faceplate of the unit, many of which will remain a mystery since the manual was lost years ago. The 1995 model will have only a few buttons for the key features people use: play, fast-forward, reverse, stop, and eject. This model will probably have even more features than the model built a decade before, yet the features will be cleverly tucked away behind a drop-down panel or sliding door, accessible when needed but not staring you in the face.

Likewise, you should ensure that features used frequently are readily available. Avoid the temptation to put everything on the first screen or load the toolbar with rarely used buttons. Do the extra analysis to find out which features can go behind the panel instead of on the faceplate.

GUI Successes

Now, let's discuss some GUI successes. Successful GUIs share many common characteristics. Most importantly, good GUIs are more intuitive than their character-based counterparts. One way to achieve this is to use real-world metaphors whenever possible. For example, an application I recently examined used bitmaps of Visa and MasterCard logos on buttons that identified how a customer was going to pay. This graphical representation was immediately intuitive to users and helped them learn the application faster.

Another important characteristic of good GUIs is speed, or more specifically, responsiveness. Many speed issues are handled via the design of the GUI, not the hardware. Depending on the type of application, speed can be the make-or-break factor in determining an application's acceptability in the user community. For example, if your application is oriented toward online transaction processing (OLTP), slow performance will quickly result in users wanting to abandon the system.

You can give a GUI the appearance of speed in several ways. Avoid repainting the screen unless it is absolutely necessary. Another method is to have all field validations occur on a whole-screen basis instead of on a field-by-field basis. Also, depending upon the skills of the user, it may be possible to design features into a GUI that give the power user the capability to enter each field of each data record rapidly. Such features include mnemonics, accelerator keys, and toolbar buttons with meaningful icons, all of which would allow the speed user to control the GUI and rate of data entry.

Dos And Don'ts

Every good developer should have the goal of designing the best GUIs possible. But how does a developer make this goal a reality? By following sound, proven GUI design principles such as those listed in the following sections.

Like any good professional, YOU need some rules for repeatable successful designs. We have used the principles offered here for work with our own customers and have taught more than 20,000 GUI-design students nationally and internationally. These principles should help you as well.

Understand People

Applications must reflect the perspectives and behaviors of their users. To understand users fully, developers must first understand people because we all share common characteristics. People learn more easily by recognition than by recall. Always attempt to provide a list of data values to select from rather than have the users key in values from memory. The average person can recall about 2,000 to 3,000 words, yet can recognize more than 50,000 words.

Be Careful Of Different Perspectives

Many designers unwittingly fall into the perspective trap when it comes to icon design or the overall behavior of the application. I recently saw an icon designed to signify "Rolled Up" totals for an accounting system. To signify this function, the designer put much artistic effort into creating an icon resembling a cinnamon roll. Unfortunately, the users of the system had no idea what metaphor the icon was supposed to represent even though it was perfectly intuitive from the designer's perspective. A reserved-icons table containing standard approved icons, such as the one shown in Figure 1, will help eliminate these problems.

Reserved Icons
Figure 1

Picture	**Meaning and Behaviour**	**Use to Identify an Application**	**Used to Identify a Function**	**Reserved Word Text Label**
ℹ	Information Message	No	Yes (identifies Information message box)	None
⚠	Warning Message	No	Yes (identifies Warning message box)	None
?	Question Message	No	Yes (identifies question message box)	None
✖	Error Message	No	Yes (identifies error message box)	None

Design for Clarity

GUI applications often are not clear to end users. One effective way to increase the clarity of applications is to develop and use a list of reserved words. A common complaint among users is that certain terms are not clear or consistent. I often see developers engaging in spirited debates over the appropriate term for a button or menu item, only to see this same debate occurring in an adjacent building with a different set of developers. When the application is released, one screen may say "Item," while the next screen says "Product," and a third says "Merchandise" when all three terms denote the same thing. This lack of consistency ultimately leads to confusion and frustration for users.

Figure 2 gives an example of a list of reserved words. An application-development group might complete and expand the table with additional reserved words.

List of Reserved Words
Figure 2

Text	Meaning And Behavior	Appears On Button	Appears On Menu	Mnemonic Keystrokes	Shortcut Keystrokes
OK	Accept data entered or acknowledge information presented and remove the window	Yes	No	None	<Return> or <Enter>
Cancel	Do not accept data entered and remove the window	Yes	No	None	Esc
Close	Close the current task and continue working with the application; close view of data	Yes	Yes	Alt+C	None
Exit	Quit the application	No	Yes	Alt+X	Alt+F4
Help	Invoke the application's Help facility	Yes	Yes	Alt+H	Fl
Save	Save data entered and stay in current window	Yes	Yes	Alt+S	Shift+Fl2
Save As	Save the data with a new name	No	Yes	Alt+A	F12
Undo	Undo the latest action	No	Yes	Alt+U	Ctrl+Z
Cut	Cut the highlighted characters	No	Yes	Alt+T	Ctrl+X
Copy	Copy highlighted text	No	Yes	Alt+C	Ctrl+C
Paste	Paste the copied or cut text at the insertion point	No	Yes	Alt+P	Ctrl+V