GUI Design Tips

This chapter covers the following topics:

- Introduction
- Do Your Research
- Screen Design
- Menu Design
- Color Usage
- Consistency Check

Introduction

Designing the screens for a GUI application requires different knowledge than designing the 3270 screens for a mainframe. Why is it different?

It is different, because GUI applications put the users in control; these applications are non-modal and unstructured. The users choose the order in which they access windows, and fields within the windows. Traditional database applications often require the users to perform operations in a specific order; these applications are form-oriented and structured.

Designing a GUI screen is also different, because the GUI interface has different capabilities than a traditional mainframe interface. You can design windows that incorporate dialog elements, such as push button controls and list box controls. As you design your GUI windows, which are called dialogs in event-driven Natural, you define the font type and size of the text, the background and foreground colors, and the size of each window.

The following sections provide some tips for effective GUI design.

Do Your Research

- Spend a few hours with your users before prototyping.

  A couple of sessions with your users to iron out their needs, likes, and dislikes is enough to give you to a good basis for beginning your design.

- Take some ideas from existing GUI designs.

  Save time by not re-inventing the GUI. Try out other GUIs with an eye for what works and what does not. Consistency within GUIs helps users learn to use new applications, improves efficiency, and reduces training costs. Get user feedback on existing GUI applications - listen to their likes and dislikes rather than develop a prototype that replicates the weaknesses of poor GUI design.

- Develop your ideas on paper before spending time developing the application online.

  It is faster for you to run through a number of screen design options for your main windows on paper before spending time to create multiple prototypes online. It is quicker than coding and you do not become attached to poor designs.

  If you include your users in the development process, they can quickly comment about their needs and likes before the application is installed in the system. Try to use a paper prototype before reaching for the online development tool.

Screen Design

- Design multiple windows for related subject matter.

  Unlike designing for 3270 monitors, where you try to maximize the number of fields per screen, GUI screens are better designed using subwindows. You can, for example, have the essential fields in the main window, and all optional or supplemental information stored in one or more subwindows. Subwindows can include choices, such as drop-down lists, for the user to browse through if they do not know the information to input into the main window. Messages and field-dependent information are more effectively presented in supplemental windows than in the main window.

- Design clear, uncluttered windows.

  Avoid cluttering your windows with more than three colors, multiple graphics, and a variety of shapes. Balance your objects on the screen with lots of white space so users are not overwhelmed by variety and distracted by the presentation. Try to keep shapes and objects to a minimum and the number of colors low.

- Design accessible, not overwhelming, windows.

  Multiple fonts, font sizes, font types or families, and color schemes can overwhelm your users, making your application seem inaccessible to them. Use a maximum of three fonts, font sizes, and font types per window. Avoid using italics and serif fonts because they often break up on the screen. Use color sparingly. Neutral colors are kindest to your users eyes. Though vibrant reds and greens are very eye-catching, remember that your users spend a lot of their day working in the windows you design.

- Design for both keyboard and mouse use.

  Some users prefer using the keyboard and memorize the short cut commands, while other users are more comfortable using the mouse. Each action should be accessible by both the mouse and the keyboard.

- Design the windows according to your users' needs.

  Though it is tempting to create fabulous-looking screens with lots of functionality, if your users do not use it, it is of no value. Remember that you are designing the application for your users to get a job done, not for you to experiment with all the functionality you have available. First find out what your users need, then tailor your design to meet their needs. You design screens with different purposes in different ways. If you want to prompt the user, you use a conversational style; if you want the user to enter values from a form, you use a data-entry style.

Conversational Screens

- Design conversational screens with field prompts.

  In a conversational-based style, users enter data from a conversation (travel reservations, for example). Conversational-based styles, in which the user relies on the screen for prompting, can be rich with labels, hints, instructions, and even questions for the users to ask their clients.

Data-Entry Screens

- Design data-entry screens with terse labels.

  In a form-based style, users enter data from a form. Each line on the input screen must match a line on the form - and the lines must be in the same order. To maintain a line-for-line correspondence, you can abbreviate labels. Headings and instructions are kept to a minimum. The only purpose of labels is to help users find their places again after interruptions.

Menu Design

The following three criteria are recommended for designing menus.

- Organize menus using the conventions defined by the operating system on which your users run the application. Microsoft Windows, for example, recommends certain menus (**File**, **Edit**, and **View**, for example), options on menus (**Cut**, **Copy**, and **Paste** on the **Edit** menu, for example), and a particular order of the menus on the menu bar (**Help** always appears at the right margin, for example).

- Arrange menus by frequency of use and decide this information through observation or usability testing.

  Anticipate whether usage changes as users become more expert. Watch that this does not violate conventions established for the operating system.

- List menu items alphabetically. Remember to follow the operating-system conventions and user recommendations for frequency of using menu items.

Color Usage

- Be as conservative as possible with color.

  Humans can remember the meaning of no more than five colors at a time, plus or minus two.

- Use color as an additional signal, not as the primary signal.

  Using bright red text to warn a user is not enough; add a warning tone. Eight percent of all males are red-green color-blind and may not notice the red text.

- On charts, do not use colors without adding a secondary key (for example, a broken or solid underline).

Users with black and white monitors must be able to understand the key without the benefit of color. Also, most users do not have color printers.

Consistency Check

- Be consistent throughout the application.

  Do not change fonts, colors, or shapes for related subjects. For example, design all the **OK** buttons in an application with the same shape, size, color, and font. If related objects are presented in different ways, users cannot use the visual clues, taking them longer to become comfortable with the application. Present similar actions in a similar way, using the same font, color, and size for related buttons.

- Adopt a naming convention (and stick with it throughout the application).

  While traditional programs tend to have one large program you modify for a name change, object-oriented programs have numerous pieces of event code that you must edit individually every time you make a name change. When you design GUI applications, you must be much more rigorous about sticking to naming conventions. This avoids a lot of cleaning up time later.