

# The Art of the GUI

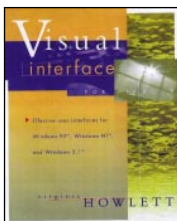
BY VIRGINIA HOWLETT

**Users know what they like in GUI design: a subtle aesthetic. Here's how to get it.**

Visual Basic programmers have faced developmental phases in mastering the principles of graphical user interface (GUI) design in the five years since introduction of VB. First there was the task of discovering how Visual Basic worked: dragging controls, placing them on forms, setting properties, and painting an interface in a matter of hours instead of painstakingly hard-coding it in weeks or months.

The next rung on the ladder of becoming a "VB interface artist" was learning when to use the right control for the right task—do I need a list box or text control? Once the paradigm was mastered and

Virginia Howlett's book, *Visual Interface Design for Windows*, will be published by Wiley Computer Books (ISBN# 0-471-13419-8). Portions of this article are reprinted from the book with the publisher's permission. During her 11 years with Microsoft, Virginia pioneered visual interface design on Windows 3.0, 3.1, 95, and various applications. Currently she is a senior product designer, researching next-generation online document technology in the Microsoft typography group.



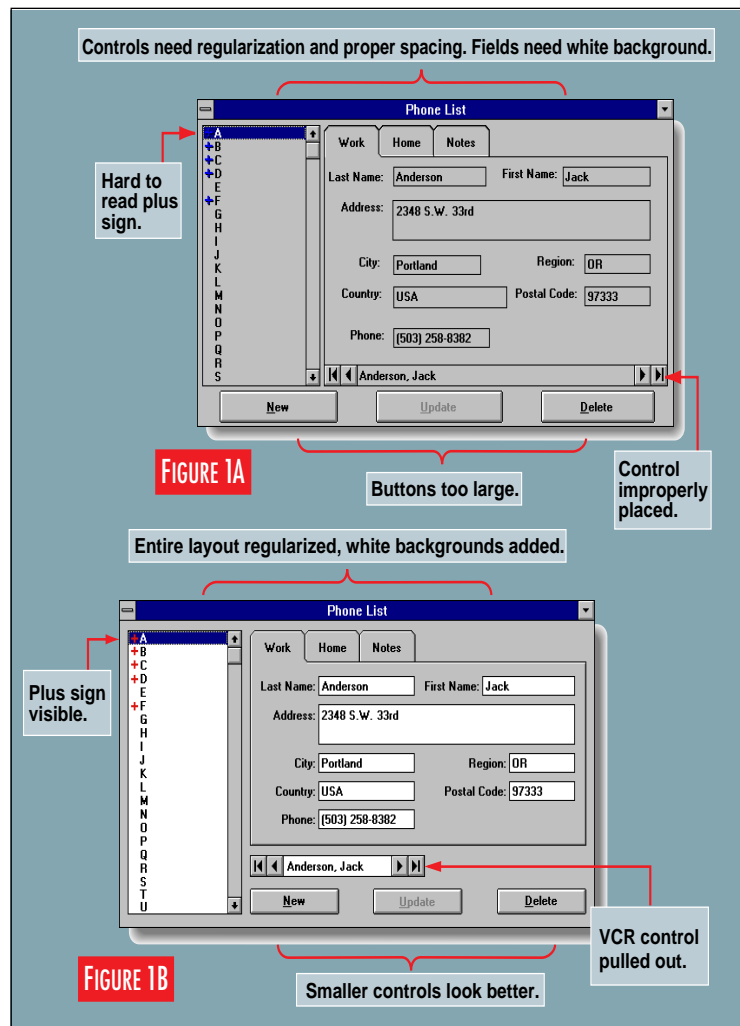
programmers learned how to use controls, they ran the risk of entering the "ransom note interface" phase, where programmers used every color, type of font, font size, and free-form layout. A little constructive feedback from users, colleagues, or even a graphic designer nudged programmers toward the next

phase—restraint and simplicity with adherence to some basic graphic design rules. Each programmer goes through these phases.

Now it's time to consider taking a step up to the next level of Windows custom application development: the GUI as art. Studying the makeovers and guidelines

**FIGURE 1** *A Makeover Step By Step.*

The original Phone List GUI (Figure 1A) lacks contrast between text and labels, the 3-D isn't carried through consistently, and shadows are incorrect. Also, shadows on the plus signs are blurred. In the first revision (Figure 1B) simple changes create vast improvements. By filling the text entry areas and list box with white, making highlights and shadows on the tabs consistent, and changing the plus to red, the function of the window is immediately clearer. In the next revision (Figure 1C), the application is updated to Windows 95 standards. It uses the more subtle border styles 3-D presentation, and it eliminates the second control. In the next revision (Figure 1D), I created a 256-color palette for textured backgrounds and 3-D elements. I've improved the fonts, simplified the controls, and added an image of the person.



I'll provide won't necessarily make your interface worthy of display at the Louvre, but these examples and tips will definitely make you a better interface designer. And keep in mind that I'm describing the artistic elements of GUI design strictly from a practical, hands-on perspective of what works to make interfaces better for your users.

All enhancements to user interfaces must be implemented with an emphasis on usability rather than pure aesthetics. In fact, usability problems are often aesthetic problems. The trend in GUI design is toward greater usability through aesthetic appeal. Not only is this true for multimedia CD titles or World Wide Web pages (for which VB is positioned as a key development tool), but for productivity applications as well. Even if you're developing front-ends to corporate databases, open your mind to new research and possibilities in GUI design. To demonstrate advanced GUI design, I'll critique interfaces from shipping software to show both good and bad GUI implementation.

I've broken this article into two sec-

tions: makeovers and common errors to avoid. Additionally, I've included a discussion of guidelines for nonvisual aspects of application development (see the sidebar, "Plan For Success"). In the makeover section, I'll walk you through a step-by-step redo of a VB app that shipped as a sample with VB 3.0. Then I'll critique a makeover of Microsoft Money. To explain common errors in the second half of this article, I'll use shipping examples of good and bad user interfaces.

You've probably seen the application used in the first makeover: it's Phone List, a sample Microsoft application that shipped with VB 3.0 (see Figure 1A). The app has a list box with expandable entries, and items in a list box being connected to the data that appears on a set of tabs. Obviously this user interface hasn't been polished.

Some big design problems are immediately obvious with Phone List. First, because everything is gray, it's hard to distinguish the text entry areas from labels. The slight 3-D shadows on the top of the tabs aren't carried through to the rest of the tab, and the shadows are inconsis-

tent with a top-left light source (highlights should appear on the top of the tabs, rather than shadows). The bright blue, drop-shadowed plus signs in the list box stand out, but they look fuzzy because there isn't enough contrast to read the shadow as a shadow. Consequently the plus sign looks blurred. Moreover, the plus sign disappears in the default highlight selection color, so it's difficult to see the plus sign on the selected line.

So much for the obvious aesthetic problems. The subtle problems are more difficult to notice but just as important. First, the text entry areas have slightly different sizes and they're not lined up exactly, which creates vague discomfort—a sense of disorganization. Slight misalignments of objects are subtly disturbing because you see objects in the

world by their edges. The controls on this tab need regularization and placement in an even array or grid layout.

Additionally, the scrollbar displaying "Anderson, Jack," at the bottom of the tab is hard to notice. It gets lost in the sea of gray background color. It's crammed close to the buttons below it. The eye easily misses the scrollbar entry when passing from the text entry areas to the buttons at the bottom. Because of its placement at the bottom, the scrollbar looks like a document scrollbar rather than a name-field scrollbar, hence it's easy to ignore. Also, the buttons at the bottom are too large and crammed into too little vertical space.

## SMALL CHANGES, BIG IMPROVEMENTS

Minor enhancements make big improvements in the application's usability and appeal. Simply filling the text entry areas and list box with white, making highlights and shadows on the tabs consistent, and changing the plus sign to a simple red character, makes the window's function immediately clearer (see Figure 1B). The red plus sign, without a drop shadow, is visible on white and on the selection highlight color. The 3-D effect on the tabs mimics the 3-D buttons, making the tabs appear selectable.

Now I'll explain the next level of improvement. Note how in Figure 1B all the text-entry areas have been regularized: they are the same height, and where possible, their left and right edges have been aligned. Also, the vertical spaces between text fields are the same. A modified grid-like appearance has been established, even though the labels are right aligned. The scrolling control has been pulled off the bottom and filled with white to show equivalence between the phone list entry and the scrolling entry. Now that the scrolling control is isolated, with space between it, the tab above, and the buttons below, it is much more visible, and its function is more obvious.

Also, placing the scrolling control text on white helps it stand out because of the increased contrast. The dialog buttons have been made a standard size, and the list box has been made longer to accommodate more entries because the list box function is more important than space for three command buttons.

This revision (see Figure 1B) would be passable as a Windows 3.1 application. It's simple and straightforward, cleanly designed, and its functions are clear. But the functionality of the interface needs improvement. Why do two controls do exactly the same thing? The expanding list control and the scrollbar control both take you to a phone list entry. Each list entry has three tabs associated with it.

Layout now uses "Win 95", highlights, shadows, and non-bold fonts.

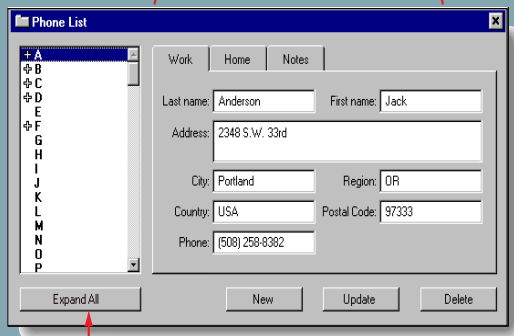


FIGURE 1C

Win95 Expand All control added.

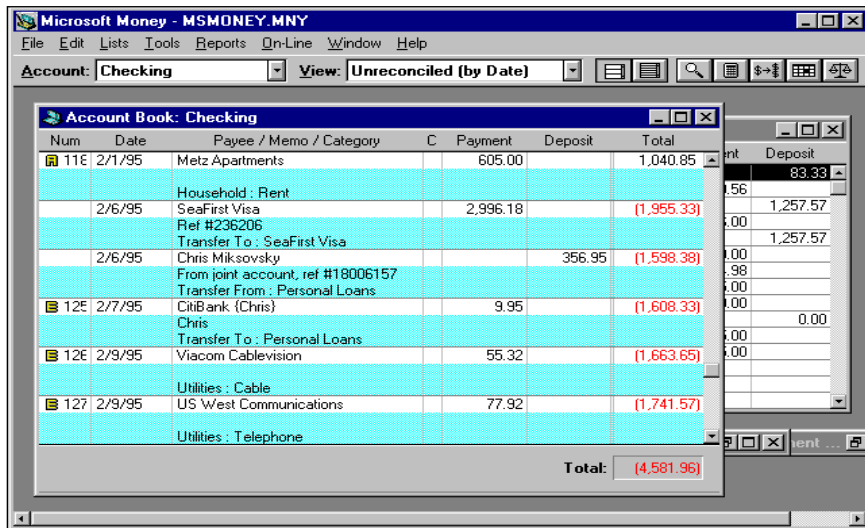
Advanced Win95 GUI uses visually textured 3D with a warm wood-grain background. Verdana—a new font—is used for better legibility at 8 to 10 pts.



FIGURE 1D

Names displayed.

One name field improves layout.



**FIGURE 2** *Old Money.* Microsoft Money 3.0 was designed to look and work like most productivity applications. Its spreadsheet-like appearance was functional, but the design left room for improvement.

## CREATING THE WINDOWS 95 LOOK

In the next revision (see Figure 1C), the phone list application is updated to Windows 95 standards. It uses the more subtle border-styles 3-D presentation, and it eliminates the second control. I've added an Expand All button to quickly give users

a full list from which to choose an entry. This change makes it easy to use the list for finding entries rather than relying on the scrolling control.

Notice how the more subtle 3-D and the non-bold font together make the interface softer and less glaring. Because a

non-bold font is used, one more regularization was added to the entry fields: the words "Postal Code" now fit in the same area as "First Name." This change creates a two-column grid of entry fields. Also, the plus signs in the expanding list have been changed to a black outline to conform to the appearance of expanding lists in Windows 95. The black outline design has the advantage of being visible on any background, and it doesn't interfere with the colors of small icons. In Windows 95, lists of items often have small 16-by-16 pixel icons next to the item name to facilitate visual identification and drag-and-drop, so it's important not to interfere with these icons.

The revisions in Figure 1C soften and clarify the functions of this application even further—making it a good Windows 95 product. Yet it could be improved with subtle icons on each tab, or a small photo of each person on the tab. Adding graphics would mean making the tabs larger and adjusting the layout. It's now a good productivity application. However, if you weren't developing this product for the office professional, and consistency with Windows 95 was unimportant, you could make it even more visually appealing.

I've enhanced this application about as much as it can be improved while still remaining true to the original design (see Figure 1D). For this version I've created a 256-color palette and used it to generate textured backgrounds and 3-D elements. I've improved the fonts, simplified the controls, and added an image of the person. Note that the controls in this makeover essentially remained the same from start to finish.

Slight visual changes can create a so-

phisticated feeling in a very simple product. Notice how the softer colors of the wood texture, the clearer font, and the graphics make the application appear more attractive. Also notice that the warm tones of the wood texture are enlivened by the spots of cooler blues. The softer colors create a more relaxed feeling, one that invites you in and seems less intimidating. Still this makeover changes only the visual qualities of the window.

Now I've reached the point of question-

ing the entire metaphor for this application. What I have works nicely, but I can consider the more possibilities. The next step is to question whether the expanding list is the best way to navigate through the phone list's set of names, and design a different interaction method. Also, a set of tabs is not the most engaging way to present phone numbers and addresses. Depending on the audience, the address/phone entries could be merged and presented as handwritten information on sheets of paper, etched into stone tablets, or chalked on a blackboard, or they could simply zoom out when the name or photo is clicked. This is the aspect of GUI design where it's useful to use your imagination or turn to a graphic designer.

## MONEY MAKEOVER

Shrinkwrapped retail applications benefit from makeovers as well. Money 3.0 was designed to look and work like most productivity applications (see Figure 2). It uses a spreadsheet-like appearance with standard windows and dialogs. However, the Windows 95 version—Money 4.0—is a completely different animal. Most of the things that make Money 4.0 an innovative product aren't evident in the visual interface design. The improvements are the result of a strong user interface design strategy, and they lie deep within the product. The makeover of Money serves as a good case study for improving any type of application because the techniques and design approaches are widely applicable.

Before analyzing the Money makeover, some background information is useful. It's well known Money competes with Quicken, and that Quicken has dominant market share. Rather than competing on features, Money designers decided to leapfrog Quicken in usability. First, the Money team conducted extensive contextual inquiry studies. They surveyed users in their homes to learn what their most common tasks were, and logged details of the methods people used to do their finances, either with software or with traditional paperwork. They analyzed market research, and decided the only possibility for beating the competition was to change the rules of the game.

Instead of mimicking office productivity applications, the designers decided Money should look and function more like a home CD title. Then they did a task analysis and built the design around making the most common tasks the simplest. They reorganized existing features to make them prominent and more usable. For example, charting was buried several dialogs deep in an obscure corner of the product, but now it's available on the startup screen.

The designers built elaborate, detailed prototypes to communicate the vision of

this innovative approach, to get the team to agree on the vision, and to work out details of the design and interaction. They included a graphic designer, and stressed the importance of the visual presentation from a very early stage. Making Money appear more like a home product meant making it more attractive and more visually engaging. Achieving this meant following through on all of the picky details of the visual and interaction designs.

## BANKING ON AESTHETICS

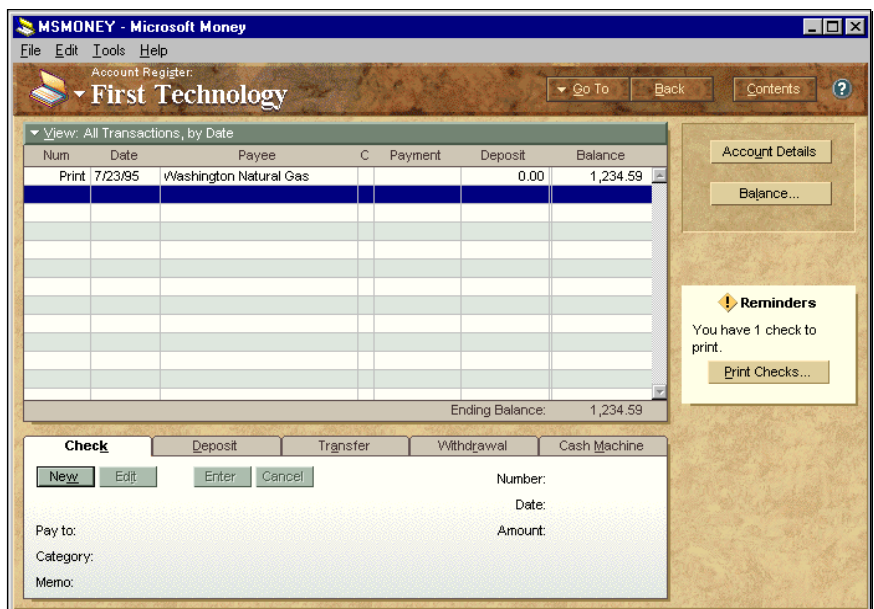
Many details contribute to the success of Money's visual presentation (see Figure 3). I'll outline a few here. Notice that remnants of the Windows 95 interface remain. Money 4.0 has a multimedia/Internet-style design, with a large headline on the upper left, and GoTo, Back, and Contents buttons to navigate.

The Money team chose two neutral-colored wood textures for different elements of the product. These textures make a good background for the functional parts of the interface, subtly unifying them, and the textures communicate a home-like, comfortable, ambiance. The darker wood unifies the navigation area of the product, while the lighter wood serves as a base for the work area. The color of these textures, and the soft grays and greens of the title/menu combinations work well visually with the color of the few Windows 95 standard menus.

This choice of subtle color combinations both tones down the product, making it appear warm and inviting, and enlivens it with gentle contrasts. Note the small, soft, drop shadows under the entry areas, as if they were paper, and the cut-in area on the upper right for functional command buttons. But also note how soft and quiet these touches are. Most users may never notice these details. All they'll see is that the product appears sophisticated, inviting, and clear.

The arrangement of elements and organization of features also works well. The most common task is positioned in the upper left, with less important tasks on the right. Functions are integrated on the same screen instead of on different windows, and everything is exactly aligned within a clear and consistent layout. In short, the product's elements are closely integrated, both visually and functionally. Because it is task based and designed with both visual sophistication and strong internal consistency, Money 4 is simple to figure out and pleasant to use. Although I'm biased, Money 4 meets all the requirements of good GUI design. It's a good interface to emulate in your designs.

In addition to improving the transaction screen for Money, a new home screen takes advantage of a CD-type look, with



**FIGURE 3** *New Money.* Microsoft Money 4.0 was designed to move beyond the look of Windows 95 to a CD or Internet style, with a large headline on the upper left, and GoTo, Back, and Contents buttons for accessing parts of the product. Usability research resulted in grouping functions by priority: first, transactions (grid), then account register information (in the new tabbed dialogs at bottom), and last, account details, balances, and notices (at right).

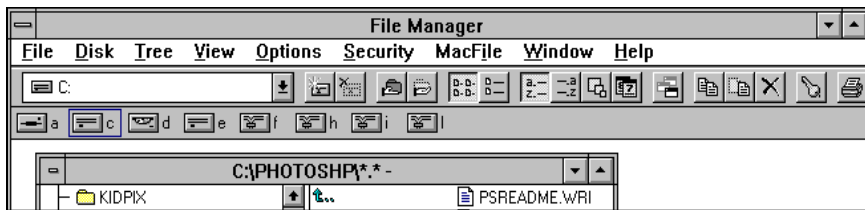


**FIGURE 4** *The New Look Of Money.* Based on usability research, designers added this new startup screen to Money 4.0 with a CD style, and functionality grouped by significance and easy access. The graphics and soft colors emulate the look of an Internet Web page or CD screen. Most important functions are grouped at the upper left, beginning with Account Register. Charting ability is displayed in the middle. Less commonly used functions (Account Manager and Investment Portfolio) are smaller and grouped at the bottom, while infrequently used features are smallest and placed at right.

careful application of the design guidelines I've discussed (see Figure 4). This startup screen makes a good GUI to emulate. The style of the Money home screen is quiet and inviting. Functionality is grouped by significance for easy access, and the graphics and soft colors emulate the look of an Internet Web page or CD

screen. The arrangement of elements and organization of features is clear.

Most important functions are grouped at the upper left, beginning with Account Register. Charting ability is displayed in the middle. Less commonly used functions (Account Manager and Investment Portfolio) are smaller and grouped at the



**FIGURE 5** *Operating Systems Need Consistency, Too.* The Windows NT toolbar uses the Windows 95 style of 3-D buttons. But the rest of NT has the older Windows 3.1 3-D appearance, with a pronounced black border and different highlights and shadows. This Win95 toolbar looks like an outsider within the NT environment.

bottom, while least-often used features are smallest and placed at right. These tasks are grouped in subtle incised frames. The sizes of the illustrations create a visual hierarchy, with more important areas larger, and the least common tasks smallest. Layering functions under different parts of the home screen makes them clear without seeming hidden.

Designers make many errors during user interface development. But if you focus on doing just seven things right, your product will be significantly better. Here are my solutions to the seven deadly design sins.

### INCONSISTENCY AND LACK OF RESTRAINT

The most common problem is lack of attention to details. This problem is even found in operating systems such as Windows NT (see Figure 5). The Windows NT toolbar uses the Windows 95 style of 3-D buttons. But the rest of NT has the older Windows 3.1 3-D appearance, with a pro-

nounced black border and different highlights and shadows. This Win95 toolbar looks like an outsider within the NT environment.

For visual consistency, establish a design strategy and style well in advance of development. Prototype extensively to work out design details. Monitor the small details within the product during development, making sure every pixel is in the right place.

Getting all the details right creates harmony—everything fits together. Sometimes you may have to give up adding something impressive because it doesn't go with rest of product. But it's better to have smooth, consistent, harmonious visuals than one exceptional element that stands out like a sore thumb.

The Microsoft Encarta Encyclopedia is a good example of visual consistency (see Figure 6). This interface, composed of flat controls, shows a non-standard user interface that is nonetheless com-

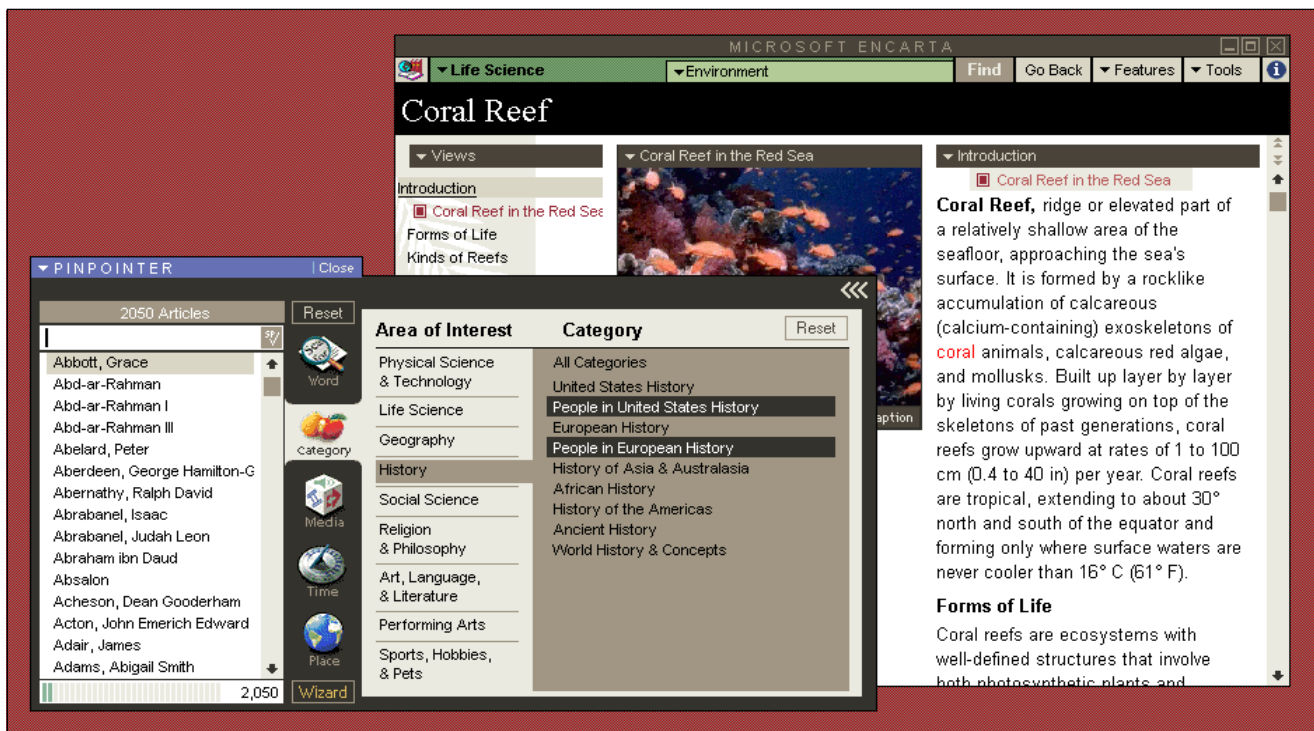
pletely consistent, tightly integrated, and visually sophisticated.

Great design means refinement. When a design lacks restraint the interface looks chaotic, disorganized, and cheap. Everyone's definition of what looks good is different. It's better to create interfaces that are simple and serviceable rather than overbearing and unappealing. Avoid using too many background colors, inconsistent layout, arbitrary uses of typography (such as hard-to-read italic headers), confusing, inconsistent and overdone 3-D elements, and brightly colored icons.

Your interfaces will look more polished and professional if you hold back the urge to make cool stuff. Think of the audience: what visual style do they expect? A simple presentation is always better than a complex one. But, don't be so restrained that your interface is boring. Add interest in the form of subtle, well-chosen illustrations, elegant icons, and well-designed typography.

### OVERBEARING METAPHORS AND BAD 3-D

The overdone metaphor is a sure way to bad design. I've seen a lot of cute pseudorealism that doesn't work consistently. Unnecessary realism, such as showing rings in a notebook, just takes up space without adding to usability. A metaphor can be helpful in providing affordances, but when taken too literally metaphors restrict the design and get in the way of the content or the task at



**FIGURE 6** *Be Consistent.* The Microsoft Encarta Encyclopedia is a good example of visual consistency. This nonstandard interface is composed of flat controls, but they are nonetheless completely consistent, tightly integrated, and visually sophisticated.

hand. You're displaying metaphors on a computer, so you'll be more successful by being true to the medium.

You don't have to be literal, especially at the expense of screen space. Only be literal when it's absolutely necessary to communicate the concept. Some pseudorealism can be good, but not if it gets in the way of content.

Windows 3.1 applications were rife with too much 3-D, inconsistent 3-D, and 3-D for the sake of having 3-D. Overuse of 3-D results in a distracting, overbearing and often just plain ugly design. 3-D effects aren't like a recipe, where if a pinch

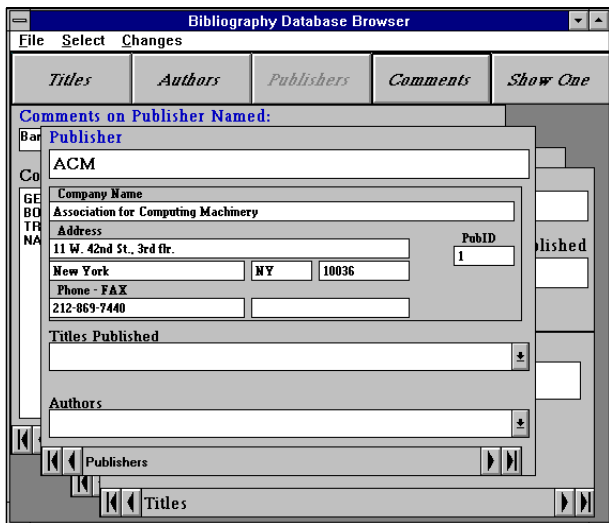
is good, more is better.

Instead of using too much 3-D, stick to consistency with the operating system. If consistency with the 3-D of Windows 95 doesn't fit your audience profile, then carefully define a design style that does and stick to that.

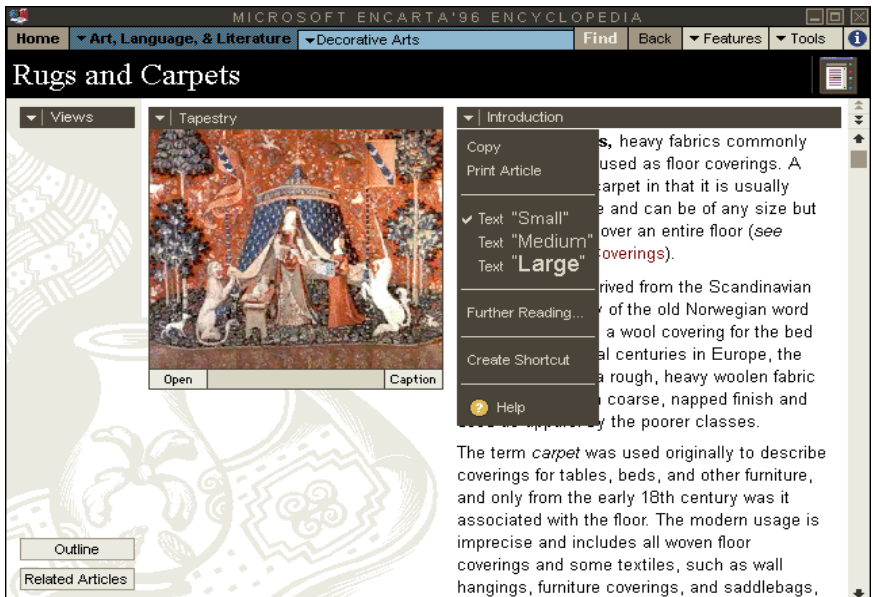
## BAD ICONS AND TYPOGRAPHY

Poorly designed icons don't convey information—they're both confusing and ugly. Using icons that don't fit the product's paradigm is another common mistake.

So many designers today have experienced designing icons, there's no longer



**FIGURE 7** *Avoid Jagged And Inconsistent Fonts.* The serif italic font in this VB screen is jagged and adds nothing to the content. The small, bold, hard-to-read serif font is used for both label text and entry field. Also, the fields are squished together. More space should be allotted between the fields.



**FIGURE 8** *Fonts Make A Statement.* Microsoft Encarta employs a clear, consistent, sophisticated use of typography. High-quality type design sets a sophisticated tone for an interface. Also, always allow the user to change the size and color of fonts from a menu, a dialog, or through the Windows 95 Control Panel.

**User Tip****VB4****SET MARGINS WHEN PRINTING WITH RICHTEXTBOX CONTROL**

*OK, I can't take credit for this one, but it's cool enough that it deserves a mention. If you've used the SelPrint method to print the contents of a RichTextBox control, you may have noticed that the output features a generous 1-inch margin on all sides of the page. The RichTextBox control offers no way to change that margin, but fortunately the Win32 API does: you can use the SendMessage API to send EM\_SETTARGETDEVICE and EM\_FORMATRANGE messages to the control. For details, see this article in the Microsoft Knowledge Base: <http://www.microsoft.com/kb/>:*

Q146022 - How to Set Up the

RichTextBox Control for WYSIWYG Printing

—Phil Weber, VBPI Technical Review Board

**SEND YOUR TIP**

*If it's cool and we publish it, we'll pay you \$25. If it includes code, limit code length to 10 lines if possible. Be sure to include a clear explanation of what it does and why it is useful. Send to 74774.305@compuserve.com or Fawcette Technical Publications, 209 Hamilton Ave., Palo Alto, CA, USA, 94301-2500.*

because there aren't many good fonts available. But it's common to see a poor choice of fonts, resulting in fonts that don't communicate the feeling or style intended by the designer. Another mistake is hard-coding fonts in a particular color and size, possibly one that may be hard to read for large segments of the population. It's also common to see fonts that are too small to read easily or too big and overdone.

The fonts in an old VB sample application I found were chosen poorly (see Figure 7). First, the buttons use a serif italic font, which is jagged, and this font adds nothing to the content. Then, a small, bold, hard-to-read serif font is used for both label text and entry fields. The fields are squished together, and it's almost impossible to read the text. Just using the standard system font, and spacing the entry fields better, would make a big difference in the usability of this form.

For effective use of fonts, have a type designer create the typography for your GUI. High-quality type design sets the tone of an interface. And always allow the user to change the size and color of fonts using a menu, dialog, or through the Windows 95 Control Panel. Microsoft Encarta uses clear, consistent, and sophisticated typography (see Figure 8). And, the interface allows the user to make the text size larger from a readily accessible menu.

**TOO MANY BRIGHT COLORS**

Everybody likes color, but it's easy to overuse it. Windows 3.1 didn't allow many choices for solid colors in the 16-color palette, and most were bright. You need only a small amount of bright color to add interest to a design. Softer, quieter, muted shades are usually better. Also, with color it's easy to assume that the user's taste is the same as yours, but preference for colors varies widely.

Use fewer colors, and use softer, more neutral colors. Use small amounts of bright color for emphasis and punch. Windows 95 has a 256-color palette that offers softer colors when running on 256-color machines. A more popular option is to design your own color palette and load it with your application the way Encarta does.

Following these straightforward guidelines can take you a long way toward improving your application. Define your product's visual style based on its audience. Set up a team of designers and developers early in the process and build bridges between them. Refine, simplify, add interest, and pay attention to the details. Talk to users, build prototypes, conduct usability tests, and improve the design. But above all, have fun! ☒