August 18
2003

THE eLEARNING DEVELOPERS' JOURNAL™

Strategies and Techniques for Designers,
Developers, and Managers of eLearning

THIS WEEK — DESIGN TECHNIQUES

Learner-interface Design: Recognizing Learner Gestures

Part 5 of 5 parts

BY MICHAEL ALLEN

Much of what we have covered in the four previous articles addressed how we speak to learners and convey information to them. But communication, the primary means of all instruction, requires listening if it is to be more than a presentation. Just as good instructors develop sensitive listening skills through which they constantly assess learner strengths and needs, e-Learning needs to listen to learners and respond to what it hears.

Computers versus instructors

Human instructors and mentors have the ability to observe their students, but computers are unable to see learners. Computers cannot take cues from a furrowed brow, a bright smile, or a blank stare. Computers do not know if learners are looking at the screen, discussing the content with a co-learner, writing in a notebook, or talking on the phone. They don't even know if their learner has left the room!

In many ways, computers are quite insensitive to learners, which severely limits the applicability of many traditional models of instruction to computer delivery. The human gestures computers *can* detect must therefore be put to good use.

Rather than attempting to mimic instructor-led approaches, models of instruction must be adapted to deal with the computer's insensitivity. These instructional models must build on the strengths of computers, not their weaknesses. Good designs compensate for

Continued on next page

True, intelligent communication between the computer and the user is still beyond the reach of most e-Learning software. Unless the learner uses gestures the computer understands, the e-Learning application is limited in its ability to respond. In this final article of the series on interface design, learn the secrets that not only make e-Learning easy to use, but that encourage learners to positively enjoy your applications.

A publication of
THE eLEARNING GUILD

THE eLEARNING DEVELOPERS' JOURNAL™

Publisher David Holcombe

Editorial Director Heidi Fisk
Editor Bill Brandon
Copy Editor Charles Holcombe

Design Director Nancy Marland

The eLearning Developers' Journal is published weekly by ***The eLearning Guild***, 525 College Avenue, Suite 215, Santa Rosa, CA 95404. Phone: 707.566.8990. ***The eLearning Guild*** is an operating unit of Focuszone Media, Inc., 1030 Beatrice Street, Eagan, MN 55121.

The Journal is included as part of ***Guild*** membership. To join the ***Guild*** go to www.eLearningGuild.com.

The eLearning Developers' Journal™ is designed to serve the industry as a catalyst for innovation and as a vehicle for the dissemination of new and practical strategies and techniques for e-Learning designers, developers and managers. The **Journal** is not intended to be the definitive authority. Rather, it is intended to be a medium through which e-Learning practitioners can share their knowledge, expertise and experience with others for the general betterment of the industry.

As in any profession, there are many different perspectives about the best strategies, techniques and tools one can employ to accomplish a specific objective. This **Journal** will share these different perspectives and does not position any one as "the right way," but rather we position each article as "one of the right ways" for accomplishing a goal. We assume that readers will evaluate the merits of each article and use the ideas they contain in a manner appropriate for their specific situation. We encourage discussion and debate about articles and provide an Online Discussion board for each article.

The articles contained in the **Journal** are all written by people who are actively engaged in this profession at one level or another — not by paid journalists or writers. Submissions are always welcome at any time, as are suggestions for articles and future topics. To learn more about how to submit articles and/or ideas, please refer to the directions in the sidebar on page 6 or visit www.eLearningGuild.com.

computer insensitivity in some valuable and powerful ways, and can actually introduce some advantages over traditional instructor-led training.

Precision

Computers are not simply disabled human instructors having abilities equal to instructors except for their sensory capabilities. Computers (and their software, of course) are weaker than human instructors in many other ways as well, such as in a computer's general inability to understand and empathize. But computers do have some capabilities that far exceed human capacities. What computers sense, for example, they tend to sense in great detail. Computers can measure precisely how long it took to make every response, count how many incorrect responses preceded each correct response, and note the order of given responses — for each and every learner.

One-to-one observation

Although instructors must often work with groups of learners, computers can focus their attention on one learner at a time. Whatever capabilities computers have are offered without bias, free from any hampering effects of repetition, a sleepless night, a wandering mind, anxiety over an anticipated event, disappointment in learner progress, a head cold, or any of the countless potential influences on human behavior. Not that computers don't have their own frailties, but, in practice, learners do generally receive the full benefit of the computer's unwavering individual attention.

Unfailing memory

Computers don't forget. Well, mine does. This is the second time I've had to start from scratch to write this section because of computer crashes — but that's another matter. In general, with solid hardware and compatible software applications, computers remember everything in detail and recall it without losing a single bit of information.

Computers can recall without error every response every learner has given, including all timing and sequencing parameters. They can remember forever what topics each learner took up and in what order topics were studied. They can remember every feedback message given and every piece of content accessed. How valuable this is, depends, of course, on instructional design.

> In many ways, computers are quite insensitive to learners, which severely limits the applicability of many traditional models of instruction to computer delivery. The human gestures computers *can* detect must therefore be put to good use.

Analysis

Computers are better at tracking details than at forming impressions or coming to conclusions — primarily, I suppose, because we tend not to write software to do this. Considerable advanced research over the last three decades has focused on applying computer power to higher-order instructional systems, such as intelligent tutoring systems, but the bulk of today's e-Learning is, unfortunately, restricted to working at less intelligent levels until practical means of developing smart teaching programs become available.

Computers can, however, make very objective analyses. Networked computers can use past experience with other learners, for example, to identify frequent problem areas. They can determine if any particular sequence of answers predicts future particular learning barriers, or if a certain level of rehearsal is essential before moving to more advanced challenges. Some of these designs are more practical than is commonly thought.

Helping learners communicate

Interactivity and the ability to individualize learning activities starts with learner responses expressed through gestures the computer can recognize. The more we know about a learner, the more effectively we can adapt instruction and increase its impact. It's therefore important to devise interactions to elicit information-rich responses. Let's look now at the mechanics of using and supporting recognizable learner gestures.

Mouse

Most people are able to handle the hand-eye coordination necessary to use a computer mouse, although it's not easy or natural for everyone. It's too easy to assume your learners will have good mousing skills, when there are still many people without them. A bad experience with the mouse can distress learners and fuel a lasting negative attitude about e-Learning.

Comfort with computers and comfort with the mouse are often closely entwined. Learners who are not comfortable with the mouse, or who are not eager to give it a try, often have apprehension about working with computers in general. Fear of computers is sometimes a result of initial difficulties in using the mouse and nothing more.

Keep mousing simple. Some interaction designs require the skilled behaviors of a frequent computer user. If you require fancy mouse manipulation, you should be sure the interaction provides a learning experience or needed information that you can't get in simpler ways.

Inactive highlights. It is a good and common practice to let users know when things they might expect to be active are not. "Dimmed" buttons are a good example. When a button appears darkened, grayed, or colorless in contrast to its usual state, users are supposed to understand that the button is inactive.

As always, it is particularly important that clickable objects are differentiated from static, never-clickable objects in some consistent fashion. Multiple conventions can be used, such as underlining hypertext and creating a 3-D button appearance for other objects, but it is important that the same designation be used throughout for each type of item.

Active highlights. As an alternative, you can simplify mouse-driven interactions by applying highlights to active objects. Highlights might take the form of a surrounding glow, a drop shadow, a bright color, underlining, and so on. Highlights can be any standard convention you use to let learners know which objects can be clicked.

Highlights should obviously be distinctive, but not distracting. They should be used consistently, appearing always when an object is clickable and never when it's not. Of course, the same style of highlighting should never be applied to static objects.

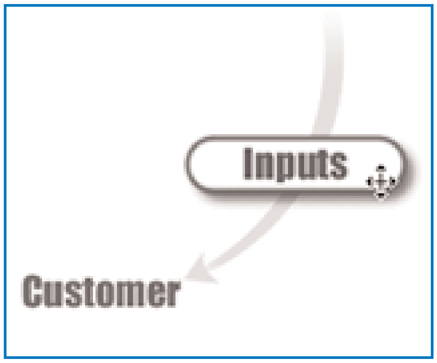


FIGURE 1 *Use drop shadows to indicate an active learning object on-screen.*

Active or inactive? Which highlighting style is best? I have a personal preference. My preference is to use both active and inactive styles, one for navigation items and another for transitory learning objects.

Because they are nearly always on the screen, I prefer to present navigation items in distinct active and inactive dress, always minimizing the need for inactive items as much as possible. For interactive learning objects, on the other hand, I prefer to highlight objects that are active, usually by giving them a drop shadow. (See Figure 1, above.) The drop shadow seems to lift them off the screen, giving them something of a natural "click me" or "move me" look.

Using dual approaches sounds contrary to notions of consistency and minimized conventions. But it actually seems to put less of a burden on learners because they understand it so easily, and it helps them focus their attention on the right things. It also gives more freedom to build attractive and meaningful displays than other approaches. The two schemes also help differentiate navigation components from interactive components.

Hide-and-seek games. A less attractive approach requires learners to actively determine what is and isn't active. In the worst case, learners must click objects to find out if they are active. If something happens, the object was active. If nothing happens, it apparently wasn't.

With this approach, if the computer takes longer than about a tenth of second to respond — that's the optimal response time in many cases — the learner may easily conclude that an active button was inactive or that things are "stuck." The learner may click again, click on a second choice, or conclude the computer has frozen. In any case, the learner's focus and the learning rhythm are disrupted. Even worse, the learner may decide that a correct answer or optimal decision is not even a viable path.

In a somewhat less annoying, but still far from optimal approach, learners must roll their mouse pointer over items to determine if they are active or not. When the cursor is over active items, evidence of their active state is revealed. (See Figure 2 on page 4.) It may be that the cursor changes shape, it may be that a highlight appears on the object, or it may be that the object itself does a little dance.

Revealing activity status on rollover is better than nothing, but it generally makes little sense to require the learner to search for active items. Activity status should usually be recognizable without learner effort. Exceptions may occur for some kinds of problem-solving or gaming contexts, so this rule is not set forth unequivocally. But there should be justification for using the learner's time and energy to ferret out what is and isn't active.

I should point out that it is usually very desirable for the cursor to change shape when it is over an active object. It's especially helpful if the shape indicates what type of action is possible, such as click or move. It's good for objects to come alive when the cursor is over them. And it's fine to highlight objects when the cursor is over them. Used alone, however, these techniques set up the generally unacceptable hide-and-seek affair described above. Using these conventions redundantly with other signs of activity status, reassures learners that things are as they expect.

Rollover tags. It is helpful to label icons and other objects that might not be easily recognized. Icons are used precisely because the object needs to be small. (See Sidebar 1 "Icons and Gestures" on page 5.) Labels are often avoided because they take too much space. The answer can be to use rollover tags, sometimes called "tool tips," which pop-up when the cursor is over the object.

Once again, however, consistency is really important. It can be close to infuriating to have tags pop up for some items and not for others. Invariably, it seems, there will be no tag offered for the item I'm most uncertain of. And, as always, inconsistency can lead learners to wonder if their computer has frozen up when it fails to respond as expected. If you're going to use rollover tags, you must provide them for pretty much everything that isn't labeled, or at least for every similar item.

Rollover events. There is something pleasing about making something happen simply by moving the cursor over an object. It would seem that clicking the mouse button is not a strenuous effort, but there is effort involved. The physical effort is negligible, but the mental effort required to make the decision to select and commit to an option is not.

Clicking an object says, "I want this." It expresses a decision the user is making. Moving the cursor over an object, on the other hand, says, "I'm considering this." If learners understand that information will come up in response to a rollover, it won't leave the current point of interaction, and will require very little effort to clear, the rollover gesture simply says, "tell me more about this." There's no user commitment, little mental effort, and minimal physical effort. Good.

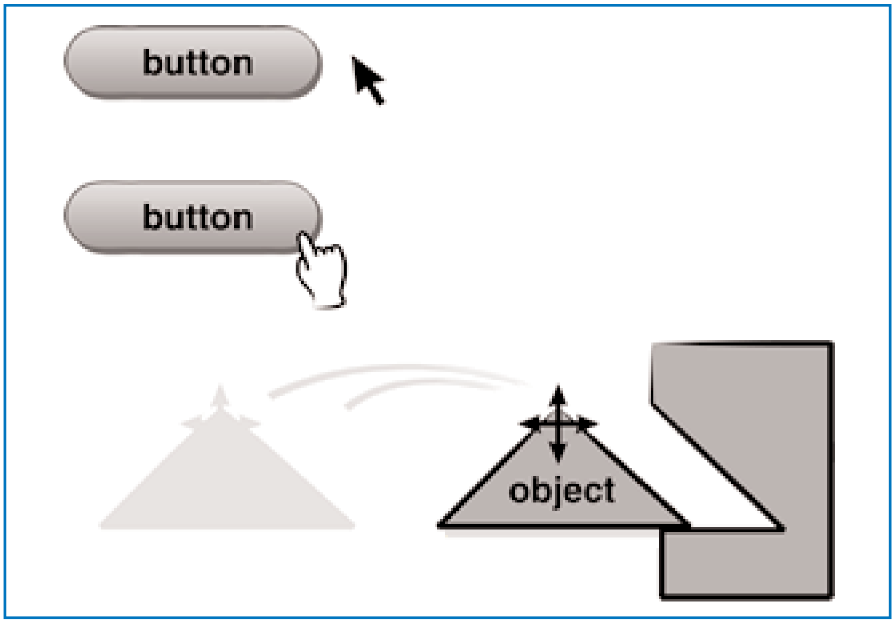


FIGURE 2 *Changing cursor shapes confirm that objects are active or movable.*

Nearly effortless retrieval of helpful information may be an excellent thinking aid and an outstanding capability of e-Learning.

Bad rollover. Not all rollover events are good, however. Bad ones create work for learners by taking them away from their current location. The learner may have been deep in thought and find the relocation quite unexpected and disruptive.

Just as it can be helpful to talk with your hands or use a pencil while reviewing a spreadsheet, learners can find it very helpful to move the cursor on the screen while they look at items and reflect upon them. Taking significant action in response to this movement is an interface design taboo. It prevents learners from using aids that may help them think and, perhaps worse, confuses the difference between mouse movement and mouse clicks.

Bad rollover, bad rollover. Another ineffective rollover design camouflages click options that require a click in the same space. For example, to detail options for the learner, such as a main menu of options, designers sometimes provide large amounts of information — sometimes even scrolling text — that is available by rolling over each item. On a number of occasions, I've seen designs allowing learners to get extensive descriptions of learning objectives by simply rolling over main menu items. While allowing nearly effortless retrieval of information seems good, the problem is that learners may not realize the option to click is also available. They may think they have done everything possible on the page by retrieving the information, or in the latter example, by listing the objectives, and wonder what to do next.

Often, the only other options on the screen are BACK and QUIT, (making the course seem very short, indeed!).

Of course, the designer's intent was that learners would first review each objective in as much detail as needed to select a topic for study. Learners would then click to select the topic which would launch a series of related interactions. However, even with on-screen instructions to click a selected topic, learners often found themselves having to ask for help. The rollover function obscured the option to click and select.

If rollovers are going to be used, the information provided should usually be short, and presented in a way that clearly labels the option of selection rather than appears as the result of a selection.

Clicking. Pointing to an object is one level of skill. Clicking an object isn't much more difficult (although knowing that you need to click an object can be another matter, as previously described). But double-clicking is surprisingly difficult for many learners.

Single and double clicks. Clicking an object to select it becomes very natural for frequent computer users, but for some, clicking takes a bit of mental adjustment. It's extremely important for applications to respond consistently to every click. If one item opens with a single click, but other items open only with a double-click, considerable confusion and anxiety can build. Truly, one little slip and learners feel they can't trust your design.

In response to some learner-interface designs tested in our evaluation lab at Control Data, complete with one-way mirrors and a sound-proof observation room into which we sequestered disbelieving designers, a frequent designer-devastating comment we heard from learners was, "I feel so stupid. I can't figure out what I'm supposed to do." It takes very, very little inconsistency for learners to feel lost and frustrated.

It's surprising to frequent computer users how obscure double-click interfaces can be, let alone right button clicks. The problem is generally that there's no signal to say "single-click for this function and double-click for this other one." When used, double-clicking should be a short-cut for functions that can be requested in other ways. It's also probably best to stick to using it as an "open" command.

Two general suggestions with respect to double-clicking:

1. Avoid it.

2. If it's really necessary, use it often enough that learners both get the hang of it and reap enough benefit from the shortcut.

Drag and drop. If double-clicking is an obstacle for learners, drag-and-drop gestures can be almost impossible for them. It's unfortunate, because there are times when this gesture seems very appropriate for the skill being taught.

Again, two points:

1. Drag-and-drop gestures are often used when it's quite unnecessary. Carefully consider your alternatives. (See alternative forms of essentially the same question, each using a different interac-

SIDEBAR 1 *Icons and gestures*

The advantage of using icons is that they are small and take up little screen real estate. The bad news is that designing instantly recognizable icons can be difficult, especially tiny ones.

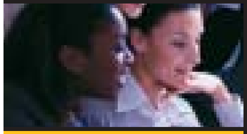
Icons are appropriate when they save time, effort, and space. They should also meet at least one of these criteria (and hopefully not just the third one):

1. Their meaning or function is instantly recognizable.

2. They will be used often and can be learned with little effort.

3. Rollover labels are provided.

If it makes learners concentrate unnecessarily on the interface, don't use icons. Instead, reduce the number of gestures learners must use to accomplish the same results, or free up a little screen space. Ideally, icons contribute imagery that helps learners identify and remember key features or ideas.



tion interface in Michael Allen's *Guide to e-Learning*, pages 281-283. The drag-and-drop interaction is hardest for learners to perform, yet it yields the same information about the learner's knowledge.)

2. To make drag-and-drop as learner-friendly as possible, provide cues before, during, and after the interaction. Consider these guidelines:

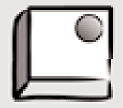
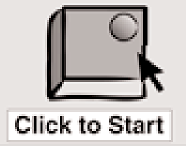
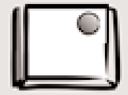
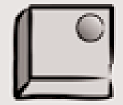
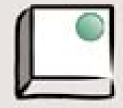
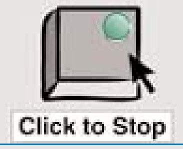
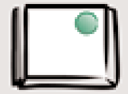
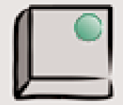
- **Clearly identify movable objects and target areas**. For example, target areas might glow and movable objects (and only movable objects) might have drop shadows.
- **Change the cursor shape when it's over a movable object, and change it again when an object is being held** (i.e., the mouse button is depressed). For example, the arrow cursor could change to a hand shape when it's over a movable object, and the hand could close as if grasping the object while the mouse button was depressed.
- **Confirm valid drop points when the dragged object is close enough to be dropped on them.** This might actually be done by changing the highlight on the nearest target. Not that it's not necessarily the correct target, but simply an indication that dropping the object here is a recognized gesture.

For some instructional applications, it's sometimes necessary to avoid some of these otherwise helpful interface structures. For example, you might not want to confirm acceptability of a drop point until after the learner has made all the adjustments desired, including the positioning of other objects. Or, you might not want to draw specific attention to movable objects, because identification of them is part of the learning challenge.

You can see how carefully instructional drag-and-drop interfaces must be devised. Designers will always find their interactions to be far more intuitive than learners will. Because of the inherent risks, avoid drag-and-drop when it has no genuine advantage over other gestures. If you must use drag-and-drop gestures for instructional operations, take extra care to provide sufficient assistance and feedback.

Click-to-place. A simpler interface that can often substitute for drag-and-drop

TABLE 1 *Button states*

Function Off	**Up**		Simulates buttons found on many physical devices. Function is off as indicated by light. Button is up.
	Over		A state not found with mechanical buttons. Confirms that button can be pressed. A label or tool tip may appear. Cursor must be "over" button.
	Depressed		Button is down, function is off at this point.
	Inactive		Function is off, but button cannot be pressed (clicked) to turn it on.
Function On	**Up**		Simulates buttons found on many physical devices. Function is on as indicated by light. Button is up as indicated by visible edges.
	Over		A state not found with mechanical buttons. Confirms that button can be pressed. A label or tool tip may appear. Cursor must be "over" button.
	Depressed		Mouse button is down, and function is on.
	Inactive		Function is on but button cannot be pressed (clicked) to turn it off.

interactions is click-to-place. The learner first clicks an object to be placed. The object highlights. The learner then clicks a desired location and the object is animated into the clicked location. No dragging or complex highlighting is involved.

If identification of objects to be moved is not a critical part of the interaction, items to be moved can be identified for the learner one at a time. In this case, learners need only identify desired locations. It's faster and simpler yet.

Provide mouse training. As we cover all the nuances of mouse gestures, it's clear there are many conventions to understand and much we expect of users. Development teams are often more interested in dealing with new content and may fail to give mouse training any attention. Especially for older trainees and those not using computers at work, some engaging mouse training can help get things off to a good start and can be very important for the success of e-Learning. In addition to doing all you can to make interaction conventions natural and intuitive, it may be very smart to provide easily accessed explanations and even practice interactions. These provisions can ensure that few learners will be unable to benefit fully from your e-Learning applications.

Buttons. Buttons come in every size and shape these days. Web site designs have led the way in experimenting with what can work (and can't work) as a button. Always looking for something fresh and new, designers have been working to break free from the constraints of traditional on-screen buttons. Screen design has clearly come a long way, especially in terms of aesthetics, but some variations have made it easier than ever for learners to overlook some of the options they have.

For quite some time, interactivity designers have felt buttons needed to look very much like physical buttons, behave like tangible buttons, and even sound like mechanical buttons. Popular operating systems have generally kept the appearance of physical buttons to reduce user anxiety and maximize friendliness. But the convention somewhat constrains display artistry and requires giving buttons a generous amount of often-precious space.

Commercial World Wide Web designs tend to ignore not only the well-established user-interface conventions for the appearance of buttons, but also what is thought of as the proper behavior of buttons (Johnson, 2000). Buttons no longer depress to give users instant feedback. Buttons don't lock down or make a click. In other words, Web site buttons don't act much like buttons at all. You can't tell if they're active, and you can't tell if you've activated them, except possibly by a swirling logo up in the corner telling you something may be happening.

Good learner-interface design does buttons better, but not without effort. Buttons may look like a simple entity, but they are surprisingly complex when done well. Let's dissect them a bit:

The look. Buttons don't have to look like physical buttons, although it helps if they do. All the things we know about using buttons are immediately at our aid when a button looks and acts in familiar ways. Of course, what is familiar changes over time. Frequent Web users now see words floating in white space as probable (though not certain) buttons, for example.

The behavior. Regardless of image, buttons are much more user-friendly when they have behaviors and states that correspond to familiar mechanical buttons. (See Table 1, above.) Movement in response to button clicks is very natural, responsive, and reaffirming button behavior. If up-state buttons appear to be raised, for example, clicking them should make them move down. If a slide switch is illustrated, then of course, it should slide. A rocker switch should rock. A rotating switch should turn. And so on.

Status reporting. Buttons should not only respond to learners but should also report the status of the function the button controls. For example, if a play button has been clicked, the button should lock down to indicate that the unit is playing. *It should not* mutate into a pause or stop button, the presence of which more naturally report pausing or stopping than playing.

When each of these states is implemented, buttons can be very comfortable to use. Just implementing up/off and depressed/on states, in fact, is a considerable improvement over the unresponsive pseudo-buttons so often found on the Web today.

Button placement. We've talked in previous articles about maintaining the learner's comfort by keeping familiar objects in the same place from display to display. We've talked about consistency in this regard as well, such as consistency in use of colors, backgrounds, overall style, and so on. Buttons are obviously components that should be handled in all these consistent ways as well, but you should also go further and consider the natural placement of buttons.

If the language you are using to communicate with learners reads from left to right, then the "forward' function is naturally a left-to-right (arrow points to the right) function and "backward" is naturally a right-to-left function (arrow points left). You should then be sure to put buttons relating to forward movements on the right and buttons relating to backward movements on the left. With other languages and cultural preferences, the natural placement might be the reverse or use up and down.

Important note. Don't put commonly used functions next to "dangerous" or irreversible functions. Don't, for example, put **BACK**, **NEXT**, and **REFORMAT HARD DRIVE** buttons all together in a cluster with even spacing. The grouping may look nice, but it invites problems.

TABLE 2 *Voice recognition*

Pros	Cons
• Alleviates typing skills • Alleviates spelling challenges (which are often irrelevant but can interfere with keyboard interactivity) • Mimics real world behaviors, such as making a sales pitch, talking with subordinates, or handling phone calls • More natural and fluid method of responding; may save time • May encourage more deliberate speech and better enunciation	• More difficult for learners to organize, edit, and review responses • Recognition errors may frustrate learners and lead to incorrect response judgements • May not tolerate accents or speech abnormalities • Requires special equipment, software, and quiet environment

Allow learners to reverse actions. Buttons are obviously very easy to click. This means it's not only easy to click the intended one, but it is also easy to click the wrong one. Unless it is required by a simulation or by the instructional paradigm in use, allow learners to reverse any action they've taken through a too-hasty click. The ability to back up both reduces the problems from clicking unintended buttons, and also lowers overall anxiety about using e-Learning.

Require confirmation. Sometimes it is too difficult to allow learners to reverse their actions. If this is the case, ask learners to confirm their intent to take the requested action. Tell them what is going to happen and ask them if this is what they want. Don't make the confirmation just a second click of the same button, however. Require them to move the mouse to a second location or to use a completely different gesture.

In all of this, make sure the effort and disruption are correlated with the consequences. Don't require extensive confirmation for an action of minor concern.

Keyboard

Close your eyes and imagine a nightmare in which you have walked into a classroom where you were going to teach for the first time. You have prepared your mind and asked that provocative opening question. You search the room to call on an eager student waving a hand high in the air in hopes of making a good impression on you. Referring to a name chart, you select one. A student stands and begins making garbled sounds while trying to communicate through some sort of frantic sign language.

You then realize his mouth is taped shut so he cannot speak. He can only wave his hands and point to things. You scan the room to find someone who can explain. Quite startled, you find all the students are eager answer the question, but they all have their mouths taped. With some considerable anxiety, you wonder just how you're going to teach this class if all they can do is point.

You awake! But the nightmare doesn't quite end. As you return to your work designing e-Learning applications, you realize that if you only allow learners to use the mouse to communicate with you, they're quite like students with their mouths taped shut.

With inventive designs, you can do amazingly well with just the mouse and its buttons, but the keyboard allows you to strip the tape off the mouths of learners so they can speak to you. You might just find letting learners speak with greater flexibility allows you to create more effective learning experiences.

Now, we need to talk about that tape over *your ears*!

The mouse may be a bit of challenge to learn, but typing is much more of a skill. Many people can type at least a little, but we are reluctant to require typing skills. We have been so inventive in finding ways the mouse can be used that the keyboard is almost in the way. But some behaviors require learners to recall information, not just point to or recognize it.

Even so, in many cases we are able to devise mouse-driven activities that depend on the recall of information. Rather than asking for the information directly, which may require learners to type, we ask them to perform a higher level task such as operating simulated machine controls in proper sequence.

Listening to the keyboard. Keyboard input may be essential for some applications, however for others, voice recognition technology may be the better interface solution, making it possible to bypass the issue of typing skills. In either case, however, you have to be prepared to listen.

COMPUTER Who invented the first electronic digital computer?

STUDENT It was either Atanasoff at Iowa State College or Mauchly at Ursinus College — there's a debate about it.

COMPUTER Close, but I need his full name.

STUDENT Whose full name?

COMPUTER That's incorrect.

Designers tend to force everything into mouse gestures probably because it takes much more preparation to analyze open-ended responses. There's a risk that correct answers won't be recognized and even a risk that incorrect answers will be accepted as correct. It's quite possible to structure questions that narrow the range of responses and to structure response analysis so that it achieves desired results. But it isn't easy, and it does take practice. There is a cost issue. And if the narrowing is too severe, the question doesn't require the needed level of cognition.

Authoring tools can be of considerable assistance in this. They typically provide the capability of ignoring irrelevant words, requiring certain words or specific phrases, adjusting to variations in word order and punctuation, accepting synonyms, and controlling the amount of spelling and punctuation variation allowed. More advanced tools provide contextual analyses that help prevent such inane dialogs as the example above.

Full keyboarding support. If learners are being trained to use software applications that use the keyboard, it obviously makes more sense to have learners make keyboard entries than to click the keys on a screen image of a keyboard.

If learners will be using the keyboard, even if somewhat intermittently, it's appropriate to have many of the interface controls activated by keyboard commands. Learners shouldn't have to click a screen SUBMIT button to enter text they have typed on the keyboard, for example. Don't force learners to needlessly bounce back and forth between the keyboard and mouse.

> One little slip and learners feel they can't trust your design. A frequent… comment we heard from learners was, "I can't figure out what I'm supposed to do." It takes very, very little inconsistency for learners to feel lost and frustrated.

Voice

Voice recognition technology is continuing to advance in capabilities and drop in cost. This is excellent news for e-Learning, because many training tasks involve spoken communication. Call center training and sales training, for example, involve learning to transmit and receive verbal information, sometimes while simultaneously editing databases or using software tools.

Continuous speech recognition is much more valuable than discrete word or command recognition for teaching communication skills, but it has been either unreliable or far too expensive until fairly recently. The authoring challenges inherent in processing open-ended responses (see "Listening to the keyboard" above) create extra work and costs, but the value of creating more authentic, real-world interactions is often worth it. (See Table 2 on page 8.) It's important that designers consider this powerful tool if it fits the targeted performance, because it may be uniquely capable of teaching skills otherwise handled effectively only through one-on-one mentoring.

Learner interface design: A final comment

Learning takes time, effort, and involvement. Through good interface design, we can focus the learner's time and effort on interactions that involve them in beneficial learning experiences. With poor interface design, we re-channel learner time and effort into coping with the mechanics of computer operation. In short, poor interface design can preclude the success of otherwise well designed e-Learning.

Unfortunately, it's far too easy to find examples of poor interface designs in products ranging from home-grown web sites to commercial applications, even those provided by the world's largest software vendors. In this series of articles, I've listed a number of terrible things you may be tempted to do in learner interface design, and I've tried to delineate some better alternatives.

Good interface designs are said to be "transparent," allowing learners to focus clearly on the content and interactions being provided. But great interface design does more than stay out of the way. It both sets the stage for meaningful and memorable experiences, helps both teaching software and learners to communicate with each other, and helps learners focus on learning rather than computing. It becomes a part of the experience, itself contributing in meaningful, fun, and helpful ways.

SIDEBAR *References*

Allen, Michael W. (2003). *Michael Allen's Guide to e-Learning: Building interactive, fun, and effective learning programs for any company*. Hoboken, NJ: John Wiley & Sons.

Johnson, Jeff. (2002). *GUI Bloopers: Don'ts and do's for software developers and web designers*. San Francisco: Morgan Kaufmann.

About the Guild

The eLearning Guild™ is a Community of Practice for designers, developers, and managers of e-Learn-ing. Through this member-driven community, we provide high-quality learning opportunities, networking services, resources, and publications. Community members represent a diverse group of instructional designers, content developers, web developers, project managers, contractors, consultants, and managers and directors of training and learning services — all of whom share a common interest in e-Learning design, development, and management.

The eLearning Developers' Journal™

The Guild publishes the only online "e-Journal" in the e-Learning industry that is focused on delivering real world "how to make it happen in your organization" information. The Journal is published weekly and features articles written by both industry experts and members who work every day in environments just like yours. As an active member, you will have unlimited access to the Journal archive.

People Connecting With People

The Guild provides a variety of online member networking tools including online discussion boards, and the Needs & Leads™ bulletin board. These services enable members to discuss topics of importance, to ask others to help them find information they need, and to provide leads to other members.

Resources, Resources, Resources

The Guild hosts the e-Learning industry's most comprehensive resource knowledge database. Currently there are over 3,300 resources available. Members have access to all of these resources and they can also post resources at any time!

Guild Research

The Guild has an ongoing industry research service that conducts surveys on a variety of topics each year. These topics are identified by the Research Advisory Committee. The data collected is available for all Members.

It's About Leadership

The Guild draws leadership from an amazing Advisory Board made up of individuals who provide insight and guidance to help ensure that the Guild serves its constituency well. We are honored to have their active engagement and participation. The Guild has also established two committees made up of active members who help steer its events program and research efforts.

Discounts, Discounts, Discounts

Guild members receive discounts on all Guild conferences and on other selected products and services. Your Guild membership will save you 20% off the list price of Guild events!

Join today at www.eLearningGuild.com!

Become a member today! Join online at www.eLearningGuild.com.

THANK YOU TO THESE GUILD ENTERPRISE SPONSORS

Because the most dramatic learning happens through first-hand experience, everything we do is focused on creating experiences — engaging, challenging and yes, fun, experiences — that people will take back to work and use every day to improve their performances.

www.alleninteractions.com
Contact: Jackie McMillan
jmcmillan@alleni.com
800.204.2635

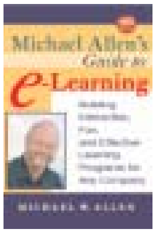
Cyclone Interactive is an interactive media and web development firm creating online, CD and presentation solutions for a wide range of clients and industries.

www.cycloneinteractive.com
Contact: Earl Dimaculangan
earl@cycloneinteractive.com
617.350.8834

To learn about ***Guild*** sponsorship opportunities, please contact David Holcombe at dh@eLearningGuild.com or call 707.566.8990.

AUTHOR CONTACT

Dr. Michael Allen is the primary architect of Authorware, the founder and former chairman of Authorware, Inc.— which merged with Macromind/ Paracomp to form Macromedia, Inc., and the Chairman & CEO of Allen Interactions. Michael is widely respected for his abilities to define, design, and build tools that allow creative individuals to harness the potential of evolving interactive multimedia technologies. In recent years, he has concentrated on creative application design and defining unique methods for developing meaningful and memorable learning applications that fully "engage the mind."



Michael is the author of *Michael Allen's Guide to e-Learning.* He holds a B.A. degree in Psychology from Cornell College, Ph.D. and M.A. degrees in Educational Psychology from The Ohio State University. Contact Michael by email at mallen@alleni.com.

Article © 2003 Michael W. Allen.

ONLINE DISCUSSIONS

Extend your learning beyond the printed page! If you are looking for more information on this topic, if you have questions about an article, or if you disagree with a viewpoint stated in this article, then join the online discussions and extend your learning.

Follow these easy steps to participate:

1. Go to http://www.eLearningGuild.com and log in.
2. Click on the *Online Discussion* link on the left-hand navigation menu.
3. Select this article by title from the *e-Learning Discussions* list, or use the *Search Subjects/Post* box to find it.
4. Click on *Add A New Message*.
5. Enter your message. It will be posted as soon as you hit the *Add Message* button on the form.

Additional information on the topics covered in this article is also listed in the Guild Resource Directory.