Answer Judging

The Quizfour Page

Judging User Responses

An integral part of Computer-Based Training (CBT) involves judging a user's response to a question for accuracy. Even in non-CBT applications, it can be useful to compare a user's response to a list of possible responses for branching purposes. The quizfour page displays four country flags and asks the user to name the countries. It judges the responses and displays feedback.

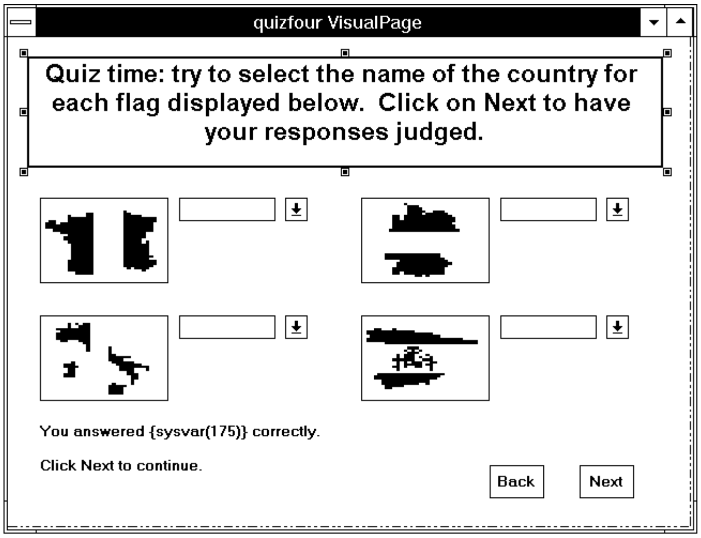
Objectives

After completing this chapter, you will be able to

- use Everest's Load Template feature
- move the Book Editor's pointer
- employ Combo objects
- edit multiple objects in one step
- enable Everest's built-in question scoring features
- specify correct answers
- drop an icon on the Book Editor
- judge user responses with the Judge icon
- provide feedback
- display sprite animation
- embed a variable for display in a Textbox
- use a System variable

Page Appearance

When this page is complete, your VisualPage editor should resemble



Quiz time: try to select the name of the country for each flag displayed below. Click on Next to have your responses judged.

You answered {sysvar(175)} correctly.

Click Next to continue.

Preparing the quizfour Page

Don't Reinvent the Wheel

The quizfour page is quite similar to the showfour page. Rather than build the quizfour page from scratch, you can use showfour as a template.

Loading a Template

Page templates can reduce your development time. Templates are previously built pages that contain commonly used elements (backgrounds, navigation buttons) that you can customize. Some authoring tools come with thousands of templates. However, experience has shown that searching through so many templates negates the time savings.

Experienced authors prefer to use as the template a prior page they have created, and that's exactly what Everest makes quick and easy. Here's how.

In the main Author window

> from the File pull-down menu, choose Load Template.

When the Load Template window appears

> double click on showfour.

When Everest asks for the name of the new page, enter

```
quizfour
```

Notice the Names

Basically, Everest has copied the page, and renamed some of the objects. Look in the Book Editor. Notice that Everest updated some of the object names to include "quizfour" (previously the names began with showfour, the name of the page from which you used as the template).

Also notice that Everest did not update the names of certain objects. Can you think of why it did not? The answer is that it does not change the name of objects you previously renamed manually.

Building quizfour

Editing the Page

Until now, you've added new objects to the page as you needed them. This time you used a prior page as a template. So, now, all you need to do is change the objects to meet the requirements of the current page.

Editing the Textbox

First,

> delete the contents of the Textbox at the top of the VisualPage editor (highlight the text by dragging the mouse across it while holding down the left mouse button then press Delete).

Then type the new contents of the Textbox:

```
Quiz time: try to select the name of the country
for each flag displayed below.  Click on Next to
have your responses judged.
```

This text is longer and probably does not fit in the Textbox. So,

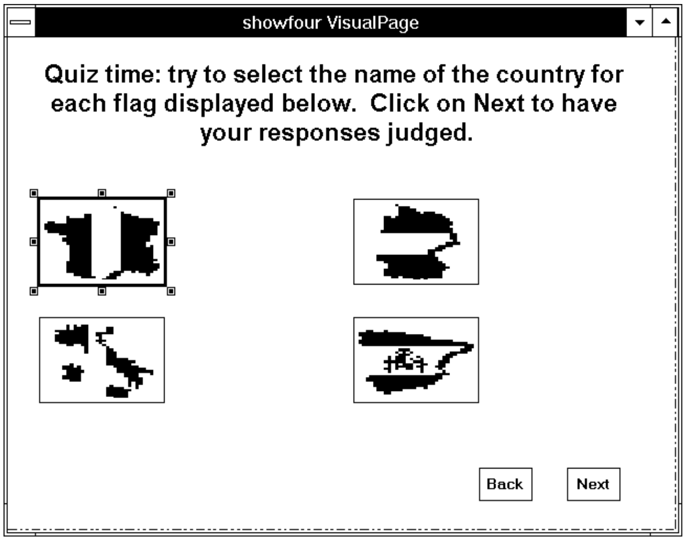
change to a smaller FontSize (try size 14 of the Arial font)

and/or resize the Textbox.

If all or part of your text seems to be missing, it might have scrolled off the top of the Textbox (if so, while the Textbox has the focus, press the Cursor up key a few times to scroll it back down).

Making Room for Responses

The quizfour page needs room for the user to enter a response for each of the four flags.   Move the flags (remember the RIGHT side mouse button drags objects after you have clicked the LEFT side mouse button to focus on them) so that your VisualPage editor resembles



Quiz time: try to select the name of the country for each flag displayed below. Click on Next to have your responses judged.

Adding the Combo Objects

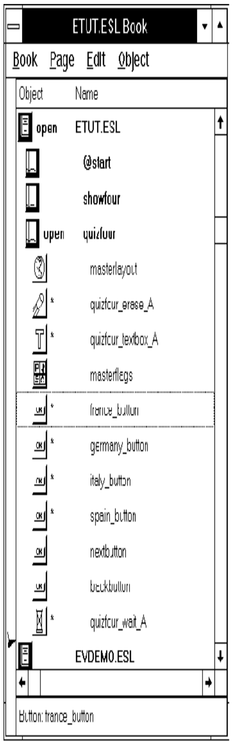
Everest offers several interactive objects that let you obtain a response from the user. The Input object is a "fill-in-the-blank" kind of field. The Check and Option objects let users pick from a group of items. The Listbox displays a list of items and allows the user to highlight one or more.

What if you'd like the user to be able to type a response OR choose one from a list? The Combo object does exactly that.

Wait! There's Wait!

Normally at this point you would drag a Combo object from the ToolSet and drop it on the VisualPage editor. But, there's a tiny complication on the page. Care to guess what that might be?

The answer is that objects dropped on the VisualPage editor are added to the page at the location of the pointer in the Book Editor. When you create a new page, or open an existing one, Everest places the pointer and the end of the page. Your Book Editor should resemble



Object	Name
open	ETUT.ESL
	@start
	showfour
open	quizfour
	masterlayout
*	quizfour_erase_A
*	quizfour_textbox_A
	masterflags
*	france_button
*	germany_button
*	italy_button
*	spain_button
	nextbutton
	backbutton
*	quizfour_wait_A
	EVDEMO.ESL

Button: france_button

Notice the location of the pointer. If you dropped a new object on the VisualPage editor now, Everest would put that icon it after the Wait icon...and if you then Previewed the page, that new object would not be displayed (because the Wait objects interrupts the execution of the page for a user response).

One Solution

One solution is to drop the icon on the Book Editor, instead of the VisualPage editor. That lets you control the position of the new object in the page.

Another Solution

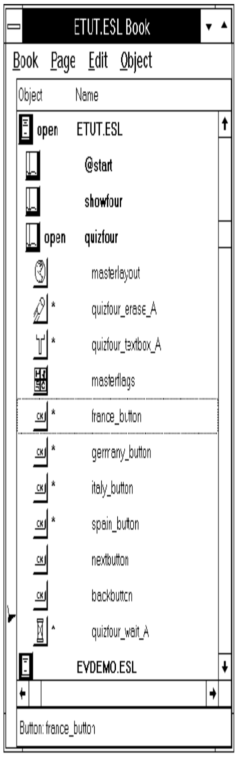
However, since you will be adding several Combo icons, it is better to move the pointer where you want it.   To do so

move the mouse pointer over the Book Editor's pointer

hold down the left mouse button

drag the Book Editor's pointer just above the Wait icon.

Now your Book Editor should resemble



Object	Name
open	ETUT.ESL
	@start
	showfour
open	quizfour
	masterlayout
*	quizfour_erase_A
*	quizfour_textbox_A
	masterflags
*	france_button
*	gerrany_button
*	italy_button
*	spain_button
	nextbutton
	backbuttcn
*	quizfour_wait_A
	EVDEMO.ESL

Button: france_buttor

Now Add the Combo Objects

The Snap-to Grid forces objects to align on certain boundaries...by default, the settings are for a 10x10 grid.

With the Book Editor's pointer properly positioned

drag a Combo icon from the ToolSet

and drop it on the VisualPage editor near the upper-left flag.

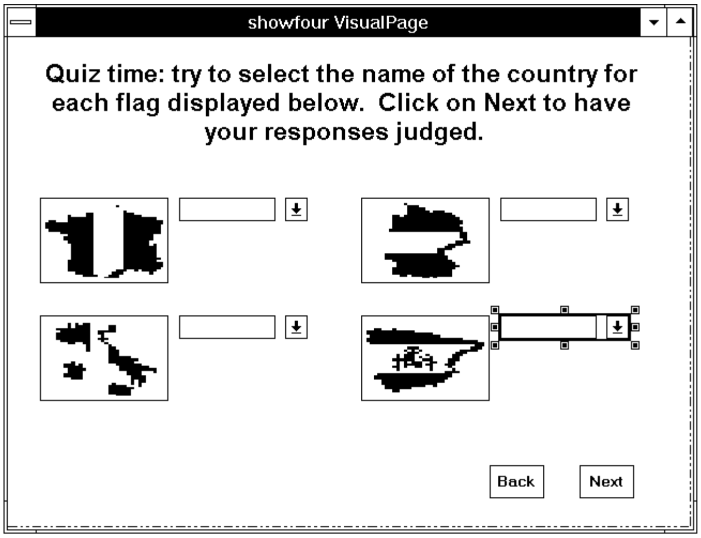
Then, in Germany, Italy, Spain order

repeat this for the three other flags

and try to position them to align approximately with the Buttons.

If you are having trouble aligning them exactly, don't worry. Everest's Snap-to grid (which is enabled by default) is forcing them into certain spots. Just do the best you can for now. Later, if you want to turn off the Snap-to-grid feature, you can do so from Everest's Settings window (from the main Author window's File pull-down menu, choose Settings).

Adjust the width of all four Combo objects so that your VisualPage editor resembles



Quiz time: try to select the name of the country for each flag displayed below. Click on Next to have your responses judged.

Combo ItemList

The user can type a response directly into a Combo object, but in this case let's also offer a list of choices. You can specify this list via the ItemList attribute.

The list of items will be the same for all four Combo objects; you could type them into each object individually, but there's an easier way.

Changing Multiple Objects At Once

Everest lets you change a particular attribute for more than one object at a time. To do so, you need to first highlight the desired objects in the Book Editor. To highlight all four Combo objects, in the Book Editor

click and hold the left mouse button on the first Combo icon

drag downward to extend the highlight to all four Combo icons.

After clicking in the Attributes window to highlight it, pressing Ctrl+Shift+i a few times will help you find ItemList.

Then, in the Attributes window,

find the ItemList attribute

click on it, and change it to

```
;Canada;France;Germany;Italy;Mexico;Spain
```

Notice the semicolon at the start, as well as between country names. Any unique character can be used as a separator between the items. Whichever character you choose as separator must be typed as the first character of ItemList.

Finally, be sure to press

Enter

which tells Everest to apply this change to all four Combo objects.

Set Tries

While the four Combo objects are still highlighted, you might as well set another attribute that is common to all. In the Attributes window

find the Tries attribute

set it to `10`

Setting Tries to anything enables automatic scoring of user responses. Be sure to press

Enter

to verify that you want to copy this setting into all four Combo objects.

The main purpose of Tries is to set the number of extra chances a user receives to answer correctly. The Design Guide has more information about Tries, and about how to use Everest's two levels of built-in scoring.

Enter the Correct Answers

The remaining Combo attribute to set is Answers1 (the correct answer). Since each Combo has a different correct answer, you no longer want to be editing all four simultaneously. So, in the Book Editor

click on the first of the Combo objects to highlight it only.

Then, in the Attributes window

set the Answers1 attribute to `france`

At run time, Everest compares what the user enters with what you specify in the Answers attributes. For Computer-Based Training type projects, this is the basis for answer judging user responses, and scoring them.

Use Lower-Case Letters

Notice the use of lower-case letters. Unless case is significant, you should employ lower-case letters in your answers. This is because (when the AdjustResponse attribute is set to Yes) Everest removes all spaces from a user's response, and converts it to lower case, prior to comparing it with the Answers attributes.

If case is significant (for example, if you really want the user to enter France, and france is unacceptable), toggle AdjustResponse to No, and include upper-case letters in the Answers attributes. To avoid annoying the user, rarely should you distinguish between upper and lower case.

More About the Answers Attributes

Everest has a wide range of answer judging capabilities that are especially useful for CBT. These include: spelling allowance, pattern matching, phonetic sound alike matching, word search, answer ranges, and more. The details can be found in the Design Guide and on-line help.

Enter the Other Answers

Next, enter the correct answer for the upper-right Combo object; this should be the Button with the flag of Germany.

In the VisualPage editor

click on this Combo object to focus on it

set its Answers1 attribute to `germany`

Repeat the process for the other two buttons, except set Answers1 to

`italy`

or

```
spain
```

as appropriate.

Disable the Flag Buttons

Recall that on the previous page, the user could click a flag Button to reveal the country name. On this page, the only purpose of the Button objects is to display the flags. Clicking them will have no effect. However, clicking on a Button still visually depresses it, and that could be confusing for the user.

The solution is to let the Buttons be visible at run time, but to disable them. That way they will ignore clicks.

In the Book Editor

> highlight all four flag Button icons (NOT the Combo icons!),

and do not include the Next or Back buttons. Then, in the Attributes window, change the attribute named Initially to

```
1
```

which makes an object visible but not enabled at run time. Finally, press

> Enter

to apply the change to all four Buttons.

Focus on the First Combo

This page contains multiple interactive objects. At run time, which one will have the focus (also called the highlight or cursor) first?

To eliminate the possibility of the focus going to the wrong object first, it is a good idea to specify which object should initially have the focus. This is quite easy to do in Everest. For this page, the focus should start on the upper-left Combo object. To make this happen, in the VisualPage editor

click on the upper-left Combo object

click on it again if it has not received the focus.

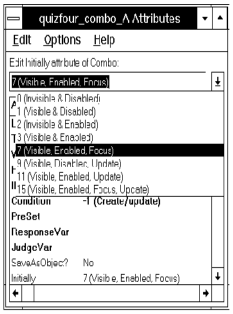
The caption of the Attributes window should read “quizfour_combo_A Attributes.” If it does not, repeat the prior step. Then in the Attributes window, set Initially to

7

You can also assign the focus by using SetFocus in A-pex3 programming.

which does three things: it makes the object 1) visible, 2) enabled, and 3) gives it the focus. A general recommendation: on interactive pages, you should assign a value of 7 to the Initially attribute of exactly one object.

Don't worry about trying to remember these special numeric settings (such as 7 for Initially). You can simply click on the drop down arrow to the right of the edit field in the Attributes window for a list of valid settings, along with descriptions of each, as illustrated below:



Edit Initially attribute of Combo:

7 (Visible, Enabled, Focus)

0 (Invisible & Disabled)
1 (Visible & Disabled)
2 (Invisible & Enabled)
3 (Visible & Enabled)
7 (Visible, Enabled, Focus)
9 (Visible, Disabled, Update)
11 (Visible, Enabled, Update)
15 (Visible, Enabled, Focus, Update)

Condition	-1 (Create/update)
PreSet	
ResponseVar	
JudgeVar	
SaveAsObject?	No
Initially	7 (Visible, Enabled, Focus)

Judging and Feedback

The Wait Object

Now its time to modify the Wait object. In the Book Editor

click on the Wait icon.

When the user clicks the Next button, you want answer judging to take place. Recall that when the user clicks the Next button, event code 34 is generated (because 34 is that button's ClickEvent).

On prior pages, in the Wait object's NextActivator attribute, you entered 34 (the event code for the PgDn key). However, in this case, when the user clicks Next you do not want to move to the next page, you want answer judging to take place. So in the Attributes window,

click on NextActivator

delete the value (clear the 34 from the edit field)

click on JudgeActivator and

set it to `34`

Same event code...different attribute. Since the event code for the Next button is now in the JudgeActivator, when the user clicks that button, Everest will perform answer judging. Or will it?

Where's JudgeAction?

You might have noticed that there is no JudgeAction attribute to accompany the JudgeActivator. That is because JudgeActivator works in a special way:

when a JudgeActivator event is matched, Everest scans downward in the page for the next Judge icon, and when it finds one, performs answer judging, and continues execution of the page with the icon after the Judge.

You'll add a Judge icon to the page in a moment.

While here editing the Wait object's attributes, verify that the BackActivator attribute is set to 33 and BackAction to BRANCH @prev. These should have been copied when you loaded the template; correct them now if needed.

Add a Judge Icon

Where should the Judge icon go? If you were to place it before the Wait icon, Everest would perform answer judging before the user had a chance to respond! Therefore, clearly, the place for the Judge icon is after the Wait icon.

But, where's the Book Editor's pointer? If you've been following the directions precisely, it is positioned above the Wait icon. You could drag the pointer downward, however, we'd like to show you a different way.

A New Drag-and-Drop Technique

NOTE: In the next step, do **not** drop the Judge icon on the VisualPage editor, instead drop it on the Book Editor, just after the Wait icon. Here are the details:

drag a Judge icon (the gavel) from the ToolSet

drag it over to the Book Editor, and before dropping it, position it just below the Wait icon

you may need to slide it up or down a bit until the Book Editor's pointer moves to just below the Wait icon

when the Book Editor's pointer is just below the Wait icon, drop the Judge icon.

This is the technique to use when you want an icon to appear in a certain location in the page. If the icon drops in the wrong spot, simply move it by dragging it within the Book Editor via the right-side mouse button.

Textbox for Feedback

In CBT, feedback typically follows answer judging. Everest lets you do anything you want as feedback...text, animation, sound, branching, whatever...you simply put the desired objects after the Judge object. For now, we'll keep the feedback simple: a Textbox that displays the number of correct responses.

From the ToolSet

drag a Textbox

and drop it near the bottom of the VisualPage editor where some room remains.

Click inside the Textbox, then type

```
You answered {sysvar(175)} correctly.
Click Next to continue.
```

Embedding Variables

The Textbox above contains an example of an embedded variable. The name of the variable is Sysvar(175). Sysvar(175) is one of approximately 200 system variables (a list can be found in the on-line help for Sysvar). Everest automatically stores certain information in system variables, and lets you access them for specific purposes. For example, Sysvar(175) contains the number of user responses that were judged as correct.

To display the value of a system variable (or any variable, for that matter) you surround it with opening and closing brace characters: { }. At run time, Everest substitutes the value of the variable for the expression inside the { }.

Insert vs. Replace

There's one subtle detail to embedding variables. When the character after the { is a lower-case letter (as in the example above), Everest inserts the expression into the line, thereby automatically adjusting the location of the text that follows.

When the character after the { is an upper-case letter, Everest replaces the expression, and does NOT adjust the location of the text that follows.   Authors use this approach when they are aligning text vertically (such as in columns), and do not want Everest to shift subsequent text.

For example, if the value of Sysvar(175) is 4, at run time Everest changes

You answered {sysvar(175)} correctly.

into

You answered 4 correctly.

If, instead, the Textbox contained (note upper-case S after {)

You answered {Sysvar(175)} correctly.

then at run time, Everest would display

You answered 4   correctly.

This is the only place in Everest that case is significant in variable names.   Most of the time, you should use lower-case letters.

After Feedback, Some Fun

Common Approaches

After displaying feedback, most authors use either of two approaches

1. if any questions were missed and some Tries remain, let the user retry
2. proceed ahead (don't give extra chances)

For now we'll use the second approach.

Some Animation Fun

In fact, let's have some fun and include some (admittedly pointless) sprite animation. But, first, you've done a fair amount of work, so

> Save the page.

Then, if it is not there already,

> drag the Book Editor's pointer to just below the Textbox that's below the Judge icon (i.e. put the pointer at the end of the page).

Next, in the ToolSet

> double click on the Picbin icon.

Note that this page already has a Picbin. That's fine. The first Picbin holds the icons for the flag buttons. This second one will hold the cels of the animation. So, in the Attributes window for the new Picbin

> double click on BMPFile

to open the Load File window, and from there, find and

> click once on mothcell.bmp

> click on the Peek button to view it.

The MOTHCELL.BMP file contains 8 cels of a moth (or butterfly, if you prefer) each with its wings in a different position. We'll use these images in Everest's AnimPath animation feature.

> Click OK to select the file.

By default, the Picbin expects more cels (10 x 7, in fact). You need to change this, so in the Attributes window

> set Columns to `4`

> and set Rows to `2`

The Moth's Container

In the @start page, you used the Animate icon to display an animation. The Animate icon can display .FLI and .FLC type animation files. Here, you'll be using a different animation technique...one that requires the Picture object.

So, from the ToolSet

drag a Picture icon (not Spicture)

and drop it anywhere on the VisualPage editor. In the Attributes window for the Picture object

set Width to `80`

and Height to `70`

which is the approximate size of one of the moth cels.

Specify the Starting and Ending Cels

Continuing in the Attributes window for the Picture object, scroll down the list to find and

double click on the AnimCelStart attribute.

This opens the Select Icon window; the 8 moths should be displayed. From here

double click on cel 1 (top left).

Back in the Attributes window

double click on AnimCelEnd

and in the Select Icon window

double click on cel 8 (bottom right).

Specify the Animation Pace

In the Attributes window

click on AnimSpritePace

and set it to `50`

This determines the time interval (in milliseconds) between animation frames; the larger the number, the slower the animation. You can try other speeds later. Start with 50 for now.

Overlaid Animation

You want the moth images to overlay the other objects on the page, so in the Attributes window for the Picture

set CopyBgnd to `Yes`

manually set TpColor to `&HFFFFFFF&` (that’s 6 letter Fs)

which makes the transparent color white (which is the background of the MOTHCELL.BMP image).

Animation Path

Finally, you need to define the path of the moth’s flight. Rather than make a particular path, wouldn’t it be nice if the moth just wandered around the window randomly. Everest makes that very easy. In the Attributes window

set AnimPath to `x,y|z`

which enables Everest’s random animation path feature (the character between the y and z is the “pipe” character, generated via a Shift+\ keypress). AnimPath has many nifty features like this one; be sure to read about them in the on-line help at a later date.

Almost Done

Careful. This is not ready to try yet. You need to tell Everest to wait for the user to view the feedback (the text and the animation) before proceeding. So, you need another Wait object.

Yes, a page can have more than one Wait object. In fact, we know of some complex pages that have 5, 6, or even more of them.

Add Another Wait Icon

So, from the ToolSet

drag a Wait icon

and drop it on the VisualPage editor.

In the Attributes window

set NextActivator to the event code for PgDn, `34`

and set NextAction to `BRANCH mchoice`

set BackActivator to the event code for PgUp, `33`

and BackAction to `BRANCH @prev`

Try It

Save

Before you run your page, in the Book Editor

from the Page menu, choose Save.

Run a Preview

The quizfour page is now complete. You can take it for a test drive. In the Author window from the Run menu, choose Preview.

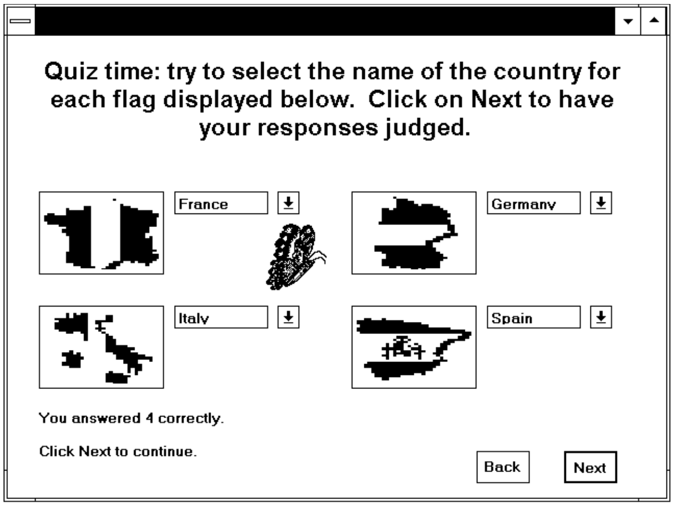
To answer the questions, either simply type your responses, or click on the Combo object's drop down arrow to show a list of choices. To move from one Combo to another, press Tab or click on the desired object with the mouse.

If the decenders on the letters in the Combo objects (like the y in Germany or Italy) are cut off (i.e. invisible), you can fix this at a later time by slightly increasing the Height of the Combo objects.

After entering your responses, click

Next

The feedback Textbox should appear to show how many you answered correctly, and the moth should start flying around the window, as shown below:



Quiz time: try to select the name of the country for each flag displayed below. Click on Next to have your responses judged.

You answered 4 correctly.

Click Next to continue.

Perhaps it would be more appropriate for the moth animation to occur only if the user enters four correct answers. That’s very easy to do...the next chapter will show you how.

But, for now, click

Next

again and Everest should tell you that the page would branch to mchoice.

Click

OK

to end the Preview.

Run From the Start

Preview runs just one page (the current). By now you might be curious about what your book looks like when executed from the beginning. To view it from the start, in the Author window

from the Run menu, choose Start at...

In the Start Running at Page dialog box,

double click on @start.

Click on the Next button to continue. As you proceed ahead, eventually Everest will report "Error -158 Page not found" upon attempting to branch to the page named mchoice.

Click Cancel

to stop the test run. This error is expected because you have not yet created page mchoice. That's covered in the next chapter.

Review

Review

In this chapter you learned how to

- use Everest’s Load Template feature

- move the Book Editor’s pointer
- employ Combo objects
- edit multiple objects in one step
- enable Everest’s built-in question scoring features
- specify correct answers
- drop an icon on the Book Editor
- judge user responses with the Judge icon
- provide feedback
- display sprite animation
- embed a variable for display in a Textbox
- use a System variable