

Simple slide show

This lesson demonstrates how to create a slideshow that displays a series of cards.

The slide show

The openCard handler sends a message called **showNextSlide** that goes to the next card after a two-second delay. Going to the next card, of course, sends another openCard message, so the handler is executed again and proceeds to the next card. The cycle continues from card to card. When the user clicks, a mouseUp message is sent. If the slideshow is already going, the mouseUp handler cancels the move to the next card. Otherwise, it starts up the slideshow again, so the user can click to pause and then click again to continue.

The openCard handler

The local declaration at the top of the script creates a variable called sMessageID. Because it is declared outside any handler, sMessageID is a script local variable, meaning it's available to all three of the handlers in this script. This is important, since more than one handler needs to access the variable. We'll see in a moment how this variable is used to store the information needed to pause the slideshow.

```
local sMessageID ## stores the message ID so we can cancel
on openCard
    send "showNextSlide" to me in 2 seconds

    ## store the ID of the delayed message you just sent, so
    ## you can cancel it if the user clicks
    put the result into sMessageID
end openCard
```

When the openCard (https://livecode.com/resources/api/#livecode_script/opencard) handler is executed, it sends the **showNextSlide** message after a two-second delay. The delay is so the user has time to see the card before the next card appears. You can change the delay time, depending on the needs of your slideshow. This message is sent to me. The keyword me (https://livecode.com/resources/api/#livecode_script/me) refers to the current object: the object whose script contains the handler that is executing. In this example, then, the **showNextSlide** message is sent to the stack, because the openCard (https://livecode.com/resources/api/#livecode_script/opencard) handler is in the stack script.

The send command

The send...in time form of the send (https://livecode.com/resources/api/#livecode_script/send) command that's used here puts information about the message into a queue called the pendingMessages (https://livecode.com/resources/api/#livecode_script/pendingmessages) and

assigns a unique number to the pending message. This number is returned in the result function, so our handler can retrieve it. A message waiting in the queue can be canceled later with the cancel (https://livecode.com/resources/api/#livecode_script/cancel) command. Since we want to cancel the pending message if the user clicks the mouse, we need to store the message's number for later use. The next line of the handler stores the message's number in the script local variable `sMessageID`, which we declared at the top of the script.

Every time LiveCode opens a card, the `openCard` (https://livecode.com/resources/api/#livecode_script/opencard) handler places a **showNextSlide** message in the `pendingMessages` (https://livecode.com/resources/api/#livecode_script/pendingmessages) queue. Two seconds later, that pending message executes, instructing LiveCode to go to the next card. This triggers a new `openCard` (https://livecode.com/resources/api/#livecode_script/opencard) message, which queues up a new **showNextSlide** message in the `pendingMessages` (https://livecode.com/resources/api/#livecode_script/pendingmessages) and the process starts over.

Note: After the pending message is sent, it is removed from the `pendingMessages` queue.

The mouseUp handler

The `mouseUp` (https://livecode.com/resources/api/#livecode_script/mouseup) handler checks the `pendingMessages` (https://livecode.com/resources/api/#livecode_script/pendingmessages) to see if the word "showNextSlide" is in the queue. If it is, the slide show is currently in action and should be stopped. All we need to do to stop the slide show is to cancel whatever pending "showNextSlide" message is waiting to be sent. Once that is canceled, the next card is not shown and the whole series of messages stops. To cancel a pending message, the `mouseUp` (https://livecode.com/resources/api/#livecode_script/mouseup) handler uses the `cancel` (<http://livecode.com/developers/api/6.0.2/command/cancel/>) command.

```
on mouseUp
    ## when the user clicks
    if "showNextSlide" is among the items of the pendingMessages then
        ## stop the slideshow
        cancel sMessageID
    else
        ## restart the slideshow
        showNextSlide
    end if
end mouseUp
```

Here is where we use the script local variable `sMessageID`. This variable was declared in the first line, and in the `openCard` (https://livecode.com/resources/api/#livecode_script/opencard) handler, we stored the message number of the pending message in the variable. The `cancel` (https://livecode.com/resources/api/#livecode_script/cancel) command needs this number in order to figure out which pending message to cancel.

If there is no pending "showNextSlide" message, the `mouseUp` handler issues a "showNextSlide" message to start the slide show going again.

The showNextSlideHandler

The **showNextSlide** handler changes the visible card. The visual effect (https://livecode.com/resources/api/#livecode_script/visual_effect) command specifies an effect to be used in the card transition.

```
on showNextSlide
    visual effect "dissolve fast"
    go next card
end showNextSlide
```

The full slide show code

These 3 handlers belong on the stack script.

```
local sMessageID ## stores the message ID so we can cancel
on openCard
    ## executed every time we go to a new card
    ## the slideshow begins when openCard is executed

    ## delay 2 seconds, then go to the next card
    send "showNextSlide" to me in 2 seconds

    ## store the ID of the delayed message you just sent, so
    ## you can cancel it if the user clicks
    put the result into sMessageID
end openCard
on mouseUp
    ## when the user clicks
    if "showNextSlide" is among the items of the pendingMessages then
        ## stop the slideshow
        cancel sMessageID
    else
        ## restart the slideshow
        showNextSlide
    end if
end mouseUp
on showNextSlide
    visual effect "dissolve fast"
    go next card
    ## which will trigger the openCard handler
end showNextSlide
```

0 Comments