PauseForUser Control Panel Example

This example illustrates using a control panel as a modal dialog via PauseForUser. This technique is useful when you need a more sophisticated modal user interface than provided by the simple input dialog.

We started by manually creating a control panel. When the panel design was finished, we closed it to create a recreation macro. We then used code copied from the recreation macro in the DoMyInputPanel function and deleted the recreation macro.

```
Function UserGetInputPanel_ContButton(ctrlName) : ButtonControl
   String ctrlName

   KillWindow/Z tmp_GetInputPanel      // Kill self
End

// Call with these variables already created and initialized:
//    root:tmp_PauseForUserDemo:numvar
//    root:tmp_PauseForUserDemo:strvar
Function DoMyInputPanel()
   NewPanel /W=(150,50,358,239)
   DoWindow/C tmp_GetInputPanel        // Set to an unlikely name
   DrawText 33,23,"Enter some data"
   SetVariable setvar0,pos={27,49},size={126,17},limits={-Inf,Inf,1}
   SetVariable setvar0,value= root:tmp_PauseForUserDemo:numvar
   SetVariable setvar1,pos={24,77},size={131,17},limits={-Inf,Inf,1}
   SetVariable setvar1,value= root:tmp_PauseForUserDemo:strvar
   Button button0,pos={52,120},size={92,20}
   Button button0,proc=UserGetInputPanel_ContButton,title="Continue"

   PauseForUser tmp_GetInputPanel
End

Function Demo1()
   NewDataFolder/O root:tmp_PauseForUserDemo
   Variable/G root:tmp_PauseForUserDemo:numvar= 12
   String/G root:tmp_PauseForUserDemo:strvar= "hello"

   DoMyInputPanel()

   NVAR numvar= root:tmp_PauseForUserDemo:numvar
   SVAR strvar= root:tmp_PauseForUserDemo:strvar

   printf "You entered %g and %s\r",numvar,strvar

   KillDataFolder root:tmp_PauseForUserDemo
End
```

Progress Windows

Sometimes when performing a long calculation, you may want to display an indication that the calculation is in progress, perhaps showing how far along it is, and perhaps providing an abort button. As of Igor Pro 6.1, you can use a control panel window for this task using the **DoUpdate** /E and /W flags and the mode=4 setting for **ValDisplay**.

DoUpdate /W=win /E=1 marks the specified window as a progress window that can accept mouse events while user code is executing. The /E flag need be used only once to mark the panel but it does not hurt to use it in every call. This special state of the control panel is automatically cleared when procedure execution finishes and Igor's outer loop again runs.

For a window marked as a progress window, DoUpdate sets V_Flag to 2 if a mouse up happened in a button since the last call. When this occurs, the full path to the subwindow containing the button is stored in S_path and the name of the control is stored in S_name.

Here is a simple example that puts up a progress window with a progress bar and a Stop button. Try each of the four input flag combinations.

```
//    ProgressDemo1(0,0)
//    ProgressDemo1(1,0)
//    ProgressDemo1(0,1)
//    ProgressDemo1(1,1)
Function ProgressDemo1(indefinite, useIgorDraw)
   Variable indefinite
   Variable useIgorDraw// True to use Igor's own draw method rather than native

   NewPanel /N=ProgressPanel /W=(285,111,739,193)
   ValDisplay valdisp0,pos={18,32},size={342,18}
   ValDisplay valdisp0,limits={0,100,0},barmisc={0,0}
   ValDisplay valdisp0,value= _NUM:0
   if( indefinite )
      ValDisplay valdisp0,mode= 4// candy stripe
   else
      ValDisplay valdisp0,mode= 3// bar with no fractional part
   endif
   if( useIgorDraw )
      ValDisplay valdisp0,highColor=(0,65535,0)
   endif
   Button bStop,pos={375,32},size={50,20},title="Stop"
   DoUpdate /W=ProgressPanel /E=1// mark this as our progress window

   Variable i,imax= indefinite ? 10000 : 100
   for(i=0;i<imax;i+=1)
      Variable t0= ticks
      do
      while( ticks < (t0+3) )
      if( indefinite )
         ValDisplay valdisp0,value= _NUM:1,win=ProgressPanel
      else
         ValDisplay valdisp0,value= _NUM:i+1,win=ProgressPanel
      endif
      DoUpdate /W=ProgressPanel
      if( V_Flag == 2 )// we only have one button and that means stop
         break
      endif
   endfor
   KillWindow ProgressPanel
End
```

When performing complex calculations, it is often difficult to insert DoUpdate calls in the code. In this case, you can use a window hook that responds to event #23, spinUpdate. This is called at the same time that the beachball icon in the status bar spins. The hook can then update the window's control state and then call DoUpdate/W on the window. If the window hook returns non-zero, then an abort is performed. If you desire a more controlled quit, you might set a global variable that your calculation code can test

The following example provides an indefinite indicator and an abort button. Note that if the abort button is pressed, the window hook kills the progress window since otherwise the abort would cause the window to remain.

```
// Example: ProgressDemo2(100)
Function ProgressDemo2(nloops)
   Variable nloops

   Variable useIgorDraw=0  // set true for Igor draw method rather than native

   NewPanel/FLT /N=myProgress/W=(285,111,739,193) as "Calculating..."
   ValDisplay valdisp0,pos={18,32},size={342,18}
   ValDisplay valdisp0,limits={0,100,0},barmisc={0,0}
   ValDisplay valdisp0,value= _NUM:0
```