Subject: Re: Widget to play and pause image stack display
Posted by Russell Ryan on Thu, 10 May 2012 16:40:11 GMT
View Forum Message <> Reply to Message

On May 10, 12:35 pm, Russell Ryan <rr...@stsci.edu> wrote:
> On May 10, 10:24 am, Helder <hel...@marchetto.de> wrote:
>
>
>
>
>
>
>
>
>
>
>> Hi,
>> I would like to have a base widget with a draw widget where I play a movie (as in displaying a series of images from a stack).
>> I would like to have the typical two buttons that start the play of the movie and the pause button.
>> I'm not that good with widgets, but I can't thick of how I would be able to interrupt a loop that displays the images.
>> I would have had something like:
>
>> PauseButtonPressed = 0
>> ImgNr = 0
>> LoopInterval = 0.2
>> WHILE ~PauseButtonPressed THEN
>> TV, Image[ImgNr,*,*]
>> ImgNr++
>> IF ImgNr GT nImages THEN ImgNr = 0
>> WAIT, LoopInterval
>> ENDWHILE
>
>> Is there a way to check if the user has clicked something in the loop?
>
>> Thanks,
>> Helder
>
> Hi Helder
> Yeah, there is a way to do this....
> Here's a little example:
>
> pro timer_event,event
> widget_control,event.id,get_uval=uval
>
> case uval of
> 'START': begin

```
>       widget_control,event.top,get_uval=state
>       (*state).stop=0b
>       widget_control,(*state).wtime,timer=(*state).time
>     end
>     'RESTART': begin
>       widget_control,event.top,get_uval=state
>       (*state).iter=0L
>       (*state).stop=0b
>       widget_control,(*state).wtime,timer=(*state).time
>     end
>     'STOP': begin
>       widget_control,event.top,get_uval=state
>       (*state).stop=1b
>     end
>     'TIME': begin
>       widget_control,event.top,get_uval=state
>
>       ;your movie stuff here:
>       n=50
>       x=findgen(n)/(n-1)*2*!PI
>       x+=(*state).iter*2*!PI/10.
>       plot,x,sin(x)
>       ;end of movie stuff
>
>       ;more to end the loop
>       if (*state).stop then return
>       if ++(*state).iter gt (*state).maxiter then return
>       widget_control,event.id,timer=(*state).time
>     end
>
>     'DRAW':
>     else:
>   endcase
>   end
>
>   pro timer
>
>   base=widget_base(/col)
>   wtime=widget_base(base,uval='TIME')
>   wdraw=widget_draw(base,xsize=200,ysize=200,uval='DRAW')
>   wstart=widget_button(base,value='Start',uval='START')
>   wrestart=widget_button(base,value='Restart',uval='RESTART')
>   wstop=widget_button(base,value='Stop',uval='STOP')
>
>   state={wdraw:wdraw,$
>          wtime:wtime,$
>          wstart:wstart,$
>          wstop:wstop,$
```

```
>         time:0.1,$
>         stop:0b,$
>         maxiter:100l,$
>         iter:0l}
>  state=ptr_new(state,/no_copy)
>  widget_control,base,/realize,set_uval=state
>  xmanager,'timer',base,/no_block
>
>  end
```

I should have said, the speed of the animation is set by time (which is in seconds). I set it to 0.1 by default (see the state structure). The looping variable is iter and the maximum size (which you don't necessarily need) is set by maxiter. You could change this to be max runtime and keep track of the length of time the animation has run, but that might get you in to trouble (like if the CPU slows down because you're doing something else). But the magic is in the widget_timer. My organization of the event_handler is probably not ideal, but it is just a get-you-started example.

Good Luck,
Russell
