Subject: CW_ANIMATE: Multiple Animations won't start properly
Posted by Haerer on Fri, 15 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hello,

I'm writing an IDL-Program to filter a set of volume
images repeatedly with interactively changed filter params.
To judge the result of each filtering, the filtered image
set should be displayed in an animation immediately after
filtering and before defining the new filter params:

- Input original image set
- Display image set in animation Ai, i=0
- Filter Loop:
    i = i + 1
    Interactively select filter params, perform filtering
    Display resulting image set in animation Ai, i=1,...
    Query for termination; if yes: Goto End
  End of Filter Loop
- End

I'm using IDL 5.2 on Windows NT 4.x

For multiple animations, the IDL-documentation refers
to use CW_ANIMATE. A program example for a single animation
is given there. From this I created a test programm calling
three subroutines:

1) create and display animation A0, as in docu-example
2) make dummy keyboard input
3) create and display animation A1, as in docu-example

Routines 1) and 3) call XMANAGER with the keyword
/NO_BLOCK or /JUST_REG to be able to make the dummy
keyboard inputs. Otherwise the program would not proceed
to step 2)

==========> MY PROBLEM :

I succeed in creating independent animations A0 and A1
in this manner, but unfortunatly A0 WILL CYCLE ONLY ONCE
AND THEN STAY FROZEN until I give my keyboard input.
This behaviour is of no use for me, since I need to judge
the filtered images with a running animation before
proceeding to input the filter params for the next step.

==========>

If I start the routines sequentially by hand A0 will
cycle properly before and after calling the keyboard input
routine.
This is nice, but also useless in practice, since I need
a main program loop doing preprocessing and filtering of
the images.

Has anybody a hint, how to get the animation running
with the programm waiting for parameter input?

My test programs are listed below.

Many thanks for every response!

Wolfgang Haerer

```
 ------------------------------------------------------------ ----
 ------------------------------------------------------------ ----
; LOAD_ANIM_TEST.COM
;
; COMMAND PROCEDURE COMPILING
; ANIMATION TEST ROUTINES AND
; STARTING THE MAIN PROGRAMM
;
.run ehandler
.run anim_test
.run anim0
.run rd
.run anim1
anim_test
 ------------------------------------------------------------ -
PRO EHANDLER, EV

; EVENTHANDLER FOR MULTIPLE ANIMATIONS

WIDGET_CONTROL, /DESTROY, EV.TOP
end
 ------------------------------------------------------------ -
PRO anim_test

; MAIN PROGRAM FOR MULTIPLE ANIMATION TEST

anim0 ; start animation #0
rd  ; get dummy kbd-input
```

```
anim1 ; start animation #1

end
 ------------------------------------------------------------ --------
PRO anim0

; CREATE TEST ANIMATION NR. 0

OPENR, 1, FILEPATH('abnorm.dat', SUBDIR = ['examples','data'])

H = BYTARR(64, 64, 16)
READU, 1, H
CLOSE, 1
H = REBIN(H, 128, 128, 16)

; Create an instance of the animation widget and load the frames.
; Note that because the animation widget is realized before the
; call to CW_ANIMATE_LOAD, the frames are displayed as they are
; loaded. This provides the user with an indication of how things
; are progressing.

base = WIDGET_BASE(TITLE = 'Animation Widget 0')

animate = CW_ANIMATE(base, 128, 128, 16)

WIDGET_CONTROL, /REALIZE, base

FOR I=0,15 DO CW_ANIMATE_LOAD, animate, FRAME=I, IMAGE=H[*,*,I]

; Start the animation:

CW_ANIMATE_RUN, animate

XMANAGER, 'CW_ANIMATE Demo', base, EVENT_HANDLER = 'EHANDLER',$
/NO_BLOCK

; Pressing the "End Animation" button kills the application.

RETURN
END
 ------------------------------------------------------------ ----
pro rd

; GET DUMMY KBD-INPUT

read,a,prompt='input a numerical test value: '
print,a
end
```

```
 ------------------------------------------------------------ ----
PRO anim1

; CREATE TEST ANIMATION NR. 1

OPENR, 1, FILEPATH('abnorm.dat', SUBDIR = ['examples','data'])

H = BYTARR(64, 64, 16)
READU, 1, H
CLOSE, 1
; H = REBIN(H, 128, 128, 16)  removed to differ from anim. #0

; Create an instance of the animation widget and load the frames.
; Note that because the animation widget is realized before the
; call to CW_ANIMATE_LOAD, the frames are displayed as they are
; loaded. This provides the user with an indication of how things
; are progressing.

base = WIDGET_BASE(TITLE = 'Animation Widget 1')

animate = CW_ANIMATE(base, 64, 64, 16)

WIDGET_CONTROL, /REALIZE, base

FOR I=0,15 DO CW_ANIMATE_LOAD, animate, FRAME=I, IMAGE=H[*,*,I]

; Start the animation:

CW_ANIMATE_RUN, animate

XMANAGER, 'CW_ANIMATE Demo', base, EVENT_HANDLER = 'EHANDLER',$
/NO_BLOCK

; Pressing the "End Animation" button kills the application.

RETURN
END
 ------------------------------------------------------------ -----

--
 ------------------------------------------------------------ ----------
Wolfgang Haerer            Tel.    : +49-9131-845591
Siemens AG, Med GT 4       Fax     : +49-9131-844771
Basic R&D                  E-mail  : Wolfgang.Haerer@med.siemens.de
Henkestrasse 127           WWW     : http://www.siemens.de/med/
D-91052 Erlangen           Intranet: http://www-gt.med.siemens.de/~hae
                           ( Siemens internal use only)
 ------------------------------------------------------------ -----------
```