Subject: Re: An Interactively rotating 3D animation ?
Posted by dick on Tue, 10 Feb 2004 06:26:06 GMT
View Forum Message <> Reply to Message

[Apparently my first reply didn't get out to everyone, perhaps because I used an attachment. Rick covered the answer nicely, but I'll re-insert my two cents' worth, for what it's worth!]

Hi,

milan08@hotmail.com (Erica Stanley) wrote in message news:<5755059c.0402090650.390b8d4f@posting.google.com>...
> Currently, I am using XObjView to display a 3D icon (IDLgrPolygon) at
> different positions inside a volume. I take a snapshot of the icon at
> each position via XObjView_Write_Image and then stream the images
> together using XInterAnimate. The problem is I need this in the form
> of a 3D animation because I want the user to be able to interactively
> rotate the model as the icon is animating through the volume. Does
> anyone have some thoughts on how I might accomplish this?

Seeing as how you're using XObjView anyway, this code might be just what you're looking for. Just compile and run it. I set up
a timer widget to trigger the updates, and you can still use the XObjView controls... if you're lucky and don't click as it's doing a redraw. :-(

If anyone has an idea for getting better response from the XObjView widgets while the animation is running, let me know!

===== AnimateXObjView.pro =====

```
;;  AnimateXObjView
;;  ---------------
;;
;;  An example of how to use XObjView to display an animated 3-D
;;  scene, where the view can be manipulated while the animation is
;;  in progress.
;;
;;  Note: The XObjView controls can be somewhat unresponsive, but
;;  they will work if you click them enough times!
;;
;;  Note: Making start/stop buttons would be a nice feature, and
;;  could easily be added to the TimerTLB widget, which should then
;;  be shown on the screen by changing Map=0 to Map=1.
;;
;;  Note: I added the tricky flags to XObjView and XManager so that
;;  this can be used with IDL Runtime or IDL Virtual
```

```
;;   Machine. Numerous object classes need to be loaded before
;;   creating a working .sav file, but a simple way to do this is:
;;   - Compile this file (Alt-F5)
;;   - IDL> Resolve_All
;;   - Run this file (F5)
;;   - Spin the scene a bit
;;   - Close the window
;;   - IDL> Save, /Routines, File='AnimateXObjView.sav'
;;
;;   February, 2004
;;
;;   Dick Jackson                /           dick@d-jackson.com
;;   D-Jackson Software Consulting /       http://www.d-jackson.com
;;   Calgary, Alberta, Canada     / +1-403-242-7398 / Fax: 241-7392

;----------------------------------------

PRO AnimateXObjViewTimer_event, event

;;    Handle one firing of a timer event:
;;    - change the scene in XObjView
;;    - cause XObjView to redraw
;;    - set another timer event to fire

;;    Get UValue of top-level base, it contains a pointer to our info
;;    structure
Widget_Control, event.top, Get_UValue=p

IF (Tag_Names(event, /Structure_Name) EQ 'WIDGET_TIMER') THEN BEGIN

   ;;    Check on XObjView window: if it is gone, destroy this window
   IF NOT Widget_Info((*p).xObjViewTLB, /Valid_ID) THEN BEGIN
      Widget_Control, event.top, /Destroy
      Return
   ENDIF ;; XObjView window is gone

   ;;    Modify objects viewed in XObjView window
   stepDelta = (*p).delta
   FOR i=0, N_Elements((*p).oOrbs)-1 DO BEGIN
      (*p).oOrbs[i] -> GetProperty, Pos=pos
      pos = pos-(Replicate(stepDelta, 3)/2)+ $
           RandomU(seed, 3)*stepDelta
      (*p).oOrbs[i] -> SetProperty, Pos=pos
   ENDFOR ;; each orb

   ;;    Cause update in XObjView
   XObjView, Refresh=(*p).xObjViewTLB
```

```
   ;;    Set another timer event to fire
   Widget_Control, event.top, Timer=(*p).interval

ENDIF ;; Widget_Timer event

END ;; AnimateXObjViewTimer_event

;----------------------------------------

PRO AnimateXObjViewTimer_Cleanup, tlb

;;    Get UValue of top-level base, it contains a pointer to our info
;;    structure
Widget_Control, tlb, Get_UValue=p

;;    Destroy objects and pointers used here
Obj_Destroy, (*p).oOrbs
Ptr_Free, p

END ;; AnimateXObjViewTimer_Cleanup

;----------------------------------------

PRO AnimateXObjView

nOrbs = 10                  ; Number of orbs
interval = 0.1              ; XObjView refresh interval (seconds)
delta = 0.1                 ; Max. amount to wiggle each orb in
                          ; each time interval

;;    Make array of orb objects
oOrbs = ObjArr(nOrbs)
FOR i=0, nOrbs-1 DO $
   oOrbs[i] = Obj_New('Orb', $
                  Pos=2*RandomU(seed, 3)-1, $ ; Range: -1:1
                  Radius=RandomU(seed)*0.1+0.1, $ ; 0.1:0.2
                  Density=RandomU(seed)*0.9+0.1, $ ; 0.1:1.0
                  Color=RandomU(seed, 3)*256) ; all colors

;;    Create XObjView window loaded with orbs
XObjView, oOrbs, TLB=xObjViewTLB, Block=LMgr(/Runtime), $
         Just_Reg=LMgr(/Runtime)

;;    Create widgets to generate timer events
info = {xObjViewTLB:xObjViewTLB, $
        oOrbs:oOrbs, $
        interval:interval, $
        delta:delta}
```

```
timerTLB = Widget_Base(/Column, Title='Timer Base', $
                  UValue=Ptr_New(info), Map=0)
Widget_Control, timerTLB, /Realize

;;    Set a timer event to fire
Widget_Control, timerTLB, Timer=interval

;;    Register timer base widget to handle events
XManager, 'AnimateXObjViewTimer', timerTLB, $
         No_Block=LMgr(/Runtime) EQ 0, $
         Cleanup='AnimateXObjViewTimer_Cleanup'

END ;; AnimateXObjView
```

=====

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392