XINTERANIMATE Example

Simple Animation Case
- Show a sequence of a simple square (64x64) image gradually changing from black (0) to white (255).
- Make the brightness transition at 1 digital count (DC) intervals

Animation Sequence without XINTERANIMATE

```idl```
IDL> a=bytarr(64,64,256)
IDL> for i=0,255 do a(*,*,i)=i
IDL> for i=0,255 do tv,a(*,*,i)
```idl```

Setup and Execution of XINTERANIMATE

```idl```
IDL> a=bytarr(64,64,256)
IDL> for i=0,255 do a(*,*,i)=i
IDL> xinteranimate, set=[64,64,256]
IDL> for i=0,255 do xinteranimate, frame=i, image=a(*,*,i)
IDL> xinteranimate,/keep_pixmaps
```idl```

XINTERANIMATE Window

CW_ANIMATE
- Similar functionality can be found in the compound widget CW_ANIMATE
- Supporting functions
  - CW_ANIMATE_LOAD
  - CW_ANIMATE_GETP
  - CW_ANIMATE_RUN
**CW_ANIMATE_DEMO**

**Widget Definition**

```idl
pro cw_animate_demo
  A = BYTARR(64, 64, 256)
  base = WIDGET_BASE(TITLE = 'Animation Widget')
  animate = CW_ANIMATE(base, 64, 64, 16)
  WIDGET_CONTROL, /REALIZE, base
  FOR I=0,15 DO CW_ANIMATE_LOAD, animate, FRAME=I, IMAGE=A[R*,*,I]
  CW_ANIMATE_GETP, animate, pixmap_vect
  CW_ANIMATE_RUN, animate
end
```

**Event Handler**

```idl
pro EHANDLER, EV
  WIDGET_CONTROL, /DESTROY, EV.TOP
end
```

**Supporting Functions**

- **CW_ANIMATE_LOAD**
  - Loads images in the same manner as XINTERANIMATE

- **CW_ANIMATE_GETP**
  - Saves "PIXMAPS" from one invocation of CW_ANIMATE widget to another
  - Analogous to XINTERANIMATE, /keep_pixmaps

- **CW_ANIMATE_RUN**

**Modal Widgets**

- Modal Widgets also known as pop-up widgets are a means to simplify a GUI interface by allowing an input window to be presented to the user at an appropriate time
- Because a new window is created and destroyed, there needs to be a mechanism to pass the values specified to the used back to the calling widget.

**Variables inside a Widget**

- Remember that variables defined in a widget only exist as long as that widget exists
- We have to somehow pass out any values defined to a calling function

**pop_fslider Case Study**

- **Problem:** We want to create a simple slider that a user can call from the IDL prompt and return to the calling routine a floating point value reflecting the position of the floating point slider (CW_FSLIDER)

```idl
IDL> a = pop_fslider()
```

- **Output:**
  ```idl
  0.5
  ```
function pop_fslider
base = Widget_Base( /column )
fslider = CW_Fslider( base, /edit )
quit_button = Widget_Button( base, value="Quit", event_pro='quit_event' )
Widget_Control, base, /realize
pointer = Handle_Create()
global_data={fslider_id:fslider,handle: pointer }
Widget_Control, base, set_uvalue=global_data
Xmanager,'pop_fslider',base
fslider_value = *pointer
print,'In Widget Definition=', fslider_value
return, fslider_value
end

pro pop_fslider_event, event
Widget_Control, event.top, get_uvalue=global_data
Widget_Control, global_data.fslider_id, get_value=fslider_value
pointer = global_data.handle
print,'In Event Handler =', fslider_value
handle_value, pointer, fslider_value, /set
Widget_Control, event.top, set_uvalue=global_data
end

pro quit_event, event
Widget_Control, event.top, /destroy
end