Using Visual Dynamic Feedback in UGOpen Programs

David West

Hughes Christensen

david.west@hugheschris.com

281-363-6109

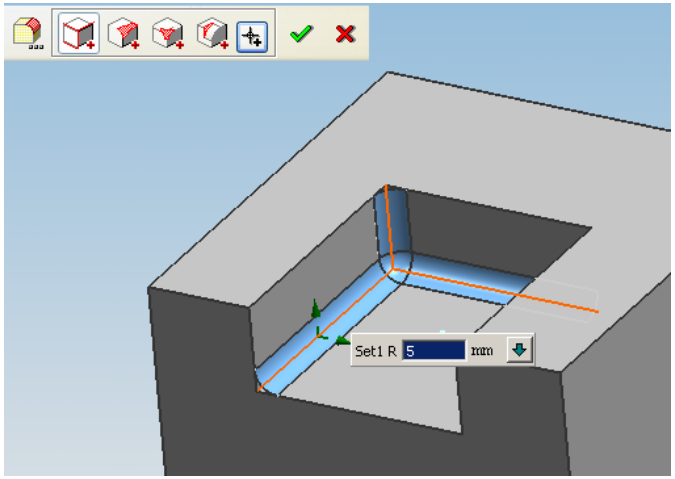
Premium Partners:

- Hughes Christensen is the leading provider of drill bits and services to the oil, gas and geothermal drilling markets.
- A Baker Hughes company
- Headquartered in The Woodlands, Texas, with operations world-wide.
- 2005 gross revenues $700M+

Introduction

- What is visual dynamic feedback?
  - On-screen motion generated by user interaction

- NX does it all the time.
  - Dragging a sketch dimension
  - Placing a projected view
  - Creating an extrusion
  - Creating an edge blend
- Provides a preview so there are no surprises



Different options

UG/Open API provides a limited number of ways to interact with the user

- Motion callbacks – make stuff happen by moving the mouse
- UIStyler dialog callbacks – make stuff happen through the dialog
- Moving solids – code to transform dumb solids
- KF dialogs – Refresh and Update
- New Common API

Motion callbacks

Method

- Prompt the user for a point
- Draw 2D geometry on the screen as the user moves the mouse

Motion callbacks

Code

```
UF_CALL(UF_UI_specify_screen_position(
        "Indicate cone position",
         (UF_UI_motion_fn_t)motion_cb_conehead,
          (void *)&my_data,
           screen_pos,
           &view_tag,
           &pos_resp ));
```

Your motion function

Structure to pass to your function or NULL

Output

Motion callbacks

Code – create the motion callback function

```
static void motion_cb_conehead( double   *screen_pos,  UF_UI_motion_cb_data_p_t
                motion_cb_data,   my_motion_data_t       *my_data )
{
   double radius, …
…
/* Draw a circle and an arc in the view of the cursor.     */
   UF_DISP_display_ogp_circle( motion_cb_data->view_tag,  my_data->matrix,
                               my_data->arc->arc_center,   radius );
   UF_DISP_display_ogp_arc(    motion_cb_data->view_tag, my_data->matrix, 15.0*DEGRA,
                345.0*DEGRA, my_data->arc->arc_center,  1.0 );
   /* Draw a bounding box around the circle.     */
   UF_VEC3_scale( radius, my_data->x_axis, xrad );
   UF_VEC3_add( my_data->abs_ctr, x_y_vec,  pos_array[0] );
     .
   UF_DISP_display_ogp_polyline( motion_cb_data->view_tag,  pos_array, 5 );
```

Motion call backs

Benefits

- Pretty simple
- UG does most of the clean up
- No UIStyler dialog required

Motion callbacks

Limits

- Only works with one UG function – UF_UI_specify_screen_position ☹
- Display 2D only
- UF_DISP_display_ogp_* calls
- All white
- Display only in the view of the cursor
- Sometimes it disappears when the cursor stops

Motion callbacks

Pleasant surprises

- Calls not limited to OGP stuff
  - Try UF_DISP_display_temporary_arc
  - UF_DISP_display_temporary_text
  - UF_DISP_labeled_conehead

You can use different colors and fonts.

But don’t forget to erase with
UF_DISP_regenerate_display

Motion callbacks

Pleasant surprises

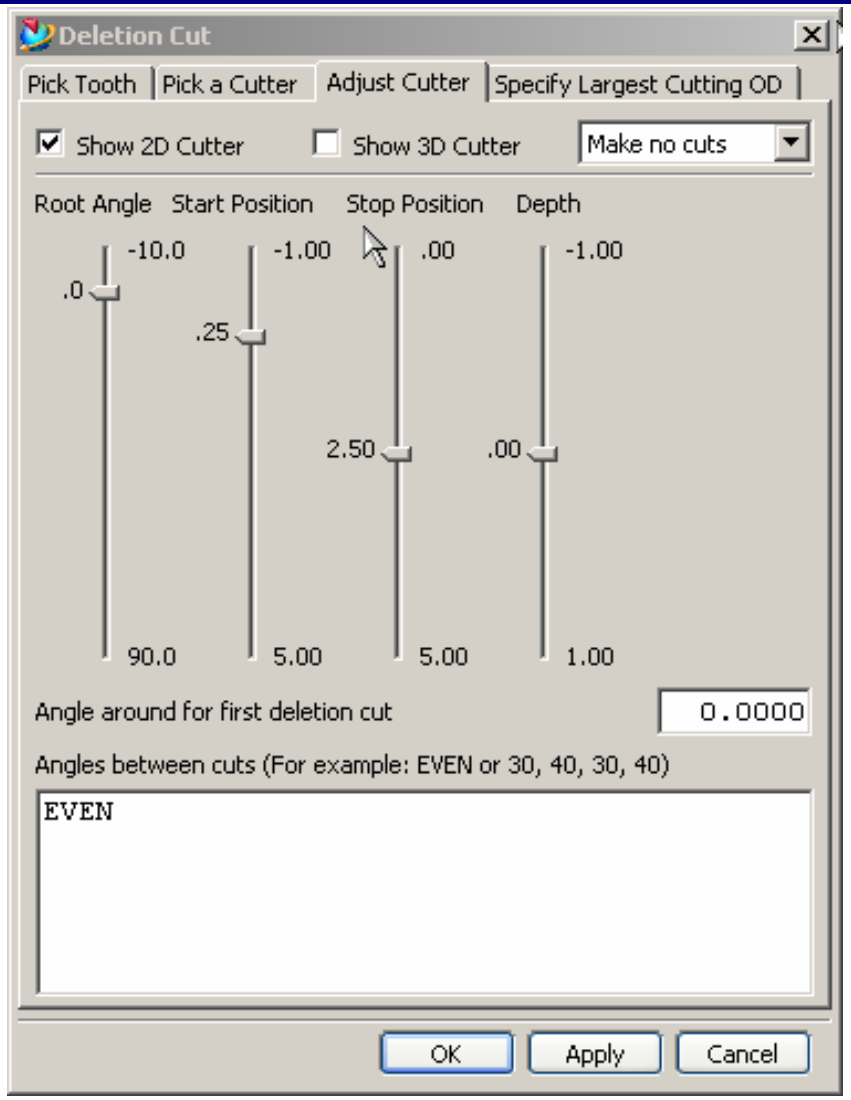
- Can call many UF_MODL functions!!
  - *Just watch your CPU*
  - *You have to clean it up*

More complicated demo

UIStyler Dialogs

Dialogs provide lots of ways to use dynamics

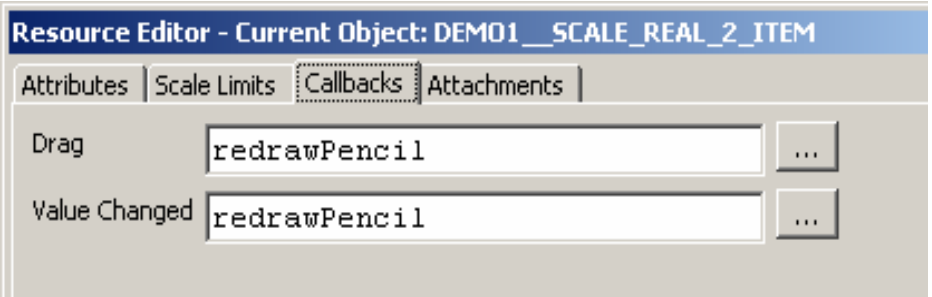
- Drag a slider
- Toggle a check box
- Button to ask for a point – then use motion callbacks



UIStyler Dialogs

Method

1. Create a dialog using UIStyler
2. Decide which sliders, toggles or buttons will support dynamics
3. Define a callback for these items



UIStyler Dialogs

- Benefits
  - Lots more options
  - Much more sophisticated dynamics
  - Can dynamically change the appearance of the dialog itself

  - Demo

UIStyler Dialogs

Drawbacks

- More work
  - Create the dialog
  - Define callbacks for dialog items
  - No automatic draw and erase

- Must own UIStyler??

UIStyler Dialogs

- Pleasant surprises
  - Can change bitmaps
  - Can dynamically turn on or off dialog items

Animate Solids

- Show 3D motion by programmatically transforming solids
- An example is tool motion for CAM verification.
- I use it for CMM motion simulation
- Demo

Animate Solids – Tricks for speed

- Importing a complex solid may be faster than building it in your program
- Try faceting the solid
  - UF_FACET_facet_solid
  - UF_FACET_disassoc_from_solid
- Structures to hold the solid body tags
- Use uf5947 to transform the bodies. It is fast!

Animate Solids - Cautions

- Watch out:
  - Speed is machine dependent. _sleep() time may have to be tweaked.
  - Be careful how you track the current position of the solids.
  - Careful using the Stop button to allow the user to halt the motion.

Animate Solids - Cautions

Code

```
for (int i=0; i<numFrames; i++)
    {
    /* sleep, then transform */
    _sleep(sleepTime);
    /* Check if abort button was pushed. */
    UF_ABORT_ask_flag_status (&flag);
    if (flag)
      {
      resp = 1;
      irc = uc1603("Adjust view or abort program.", resp, ContinueOrQuit, 2);
      if (irc==5)
                UF_ABORT_clear_abort ( );
      }
    /* Transform the solid */
    if (facetBodyArrayCount>0)
      uf5947_ (bigMatrix, facetBodyArray, &facetBodyArrayCount, &one, &zero, &two,
    NULL, NULL, &irc);
    }
```

Animate Solids

- Benefits
  - Show 3D solids moving
  - Unlimited solid complexity
- Limits
  - Speed
  - Difficult to stop
  - No real use interaction – they start it and watch

Knowledge Fusion

- All the benefits of UIStyler dialogs
- Really easy to connect the dialog to the solids.
  - Simply name the dialog item the same name as the KF parameter
  - Define a call back for the item as either Refresh Update or Rule
  - *See the Knowledge Fusion Language Help → Special Topics → User Interface Styler Dialogs*

Knowledge Fusion

- Demo
- What about the undo buffer??

Knowledge Fusion

- Benefits
  - Super easy to build the interface
  - Not limited to just 2D or OGP entities
- Limits
  - Have to own Knowledge Fusion
  - Knowledge Fusion
  - Undo buffer out of control

Other interface items

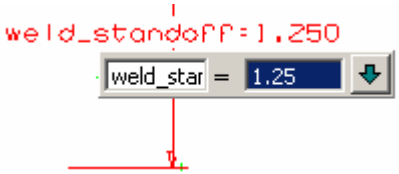
- Here are some other interface items to consider
  - Display a URL
  - On-screen bitmaps

New Common API

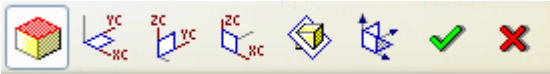
- The dialogs are true windows dialogs!
  - Not limited by UIStyler
  - Supports motion callback for point selection
  - Not sure about other motion abilities in the new API
- Drawbacks
  - You have to purchase NX Open for .NET
  - No training classes available

What I really want

- WIFMs Mini pop-up for data entry



- Mini toolbars



- Create solids without the overhead of feature creation
- Different motion with different mouse buttons
- Ability to program spaceball buttons and use the spaceball for motion in my program
- More places to use motion callbacks
- Ability to know if the call back function is in erase or draw mode.

Questions???