Dynamic UI Resizing

A technique for flexible front panel scaling
Agenda

- Introduction
- A scaling VI
- A scaling class hierarchy
- Adding interaction
- Decoupling
- Afterthoughts

- Questions
Introduction (About)

- Wiebe Walstra
- Wiebe@CARYA
Introduction

- Dynamic scaling \ dynamic panes

- The code is not:
  - A product
  - Finished
  - Polished

- The code is:
  - Working for me
  - Flexible
  - Freely available

- Omitting 90% of the details
Introduction
A Scaling VI
A Scaling VI

- Scaling anything\everything is a problem
  - GObject Position is read\write
  - GObject Bounds is read only
  - Set Size is a pain
Library name or path: LabVIEW

Function name: ResizeObjFromRef

Thread:
- Run in UI thread
- Run in any thread

Specify path on diagram:

Function prototype:

```c
int32_t ResizeObjFromRef(int32_t Reference, int32_t *AddX, int32_t *AddY);
```

Consider using a wizard instead...
A Scaling VI

```
<table>
<thead>
<tr>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
</tr>
<tr>
<td>AddX</td>
</tr>
<tr>
<td>AddY</td>
</tr>
</tbody>
</table>
```
A Scaling VI
A Scaling VI

- Object
  - + Init Object
  - + Scale
A Scaling VI

Bounds

\[ x' = Ax + B \]
\[ y' = Ay + B \]
A Scaling VI

DEMO 1
A scaling class hierarchy
A scaling class hierarchy
A scaling class hierarchy
A Scaling VI

Object
+ Init Object
+ Resize

Viewport
+ Init View
+ Resize
A Scaling VI

Object
- Init Object
- Resize

Viewport
- Init Viewport
- Resize

Divider
- Init Divider
- Resize
A Scaling VI

Common parent. (Why?)

<table>
<thead>
<tr>
<th>Object</th>
<th>Viewport</th>
<th>Divider</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Init Object</td>
<td>+ Init Viewport</td>
<td>+ Init Divider</td>
</tr>
<tr>
<td>+ Resize</td>
<td>+ Resize</td>
<td>+ Resize</td>
</tr>
</tbody>
</table>
What if splitter has a splitter?
A Scaling VI

Object
+ Init Object
+ Resize

Viewport
+ Init Viewport
+ Resize

Splitter
+ Init Splitter
+ Resize

Splitter and Viewport mostly the same!
How does IContainer know how to scale?
OBJECT HIERARCHY
Adding interaction
Adding interaction
Adding interaction
Adding interaction
Object

IObject

...# Hit Splitter

IContainer

...# Hit Splitter

Viewport

...Splitter

...# Hit Splitter

Interaction

...+ On Mouse Move
Adding interaction

Start splitter VIs
Handled by class
Adding interaction
Adding interaction

DEMO 4
Decoupling
Decoupling
Object

…

IObject

…

IContainer

…

Interaction by reference

+ Init Interaction by reference

ALL METHODS

Viewport

…

Splitter

…

Interaction

…
Decoupling
Decoupling
Decoupling
Afterthoughts
Afterthoughts

- Plays nice with SubPanels
- More icing on the cake
  - Min, max
  - Hit detect
  - Show Hide is a reg.ex.
  - Query visibility
  - Events
  - Alternatives container
  - Update objects in a container
  - Load\store screen (coming)
https://github.com/Carya-Automatisering/Dynamic-Scaling

https://www.youtube.com/watch?v=8T-lfM7kpXg
GDevCon 2018 presentation
Questions