Spiffo (2020 Oct 22)

Mmmmmm, that's interesting I was solving this particular problem by using uiGroup Modulators on top of each other and then just setting them to visible / invisible like this:

```
panel:getModulatorByName("plate_reverb_group"):getComponent():setVisible(false)
panel:getModulatorByName("chamber_reverb_group"):getComponent():setVisible(true)
panel:getModulatorByName("infinite_reverb_group"):getComponent():setVisible(false)
```

So what is the killer advantage of using Layers then? I mean what am I missing out on?

Mmmmm, I spoke too soon.
Placing one uiGroup on top of another one automatically makes the one on top a member of the lower group. So that means that setting the lower uiGroup invisible automatically makes all the other uiGroups on top of it invisible as well. If you set both uiGroups visible then you end up with a jumble of modulators from both groups all visible at the same time, which is not what I was after at all!
Removing the 'Component Generic – Group' setting or unticking the 'Is component a member of a group' doesn't seem to make any difference. If you have one uiGroup on top of another one then it belongs to the lower group and that's that!
OK, after more testing, please find attached 2 Test Panels, one with uiGroups on top of one another and another with uiTabs
It is definitely possible to layer (for want of a better word) Tabs and Groups on top of one another and then make selective ones Visible / Invisible.

I was struggling with my PCM80 Panel but doing things in the following order seems to work:
1 Delete any text from the 'Component Generic – Group' entry
2 Un-Tick the 'is component a member of a group', the uiGroup will leap to the 0,0 position on the Panel
3 Drag the uiGroup back down to its original position
4 Check that in the Component Generic section that there is nothing in the 'Group' entry and 'is component a member of a group' is stil un-ticked

At the moment that is working, I can put 4 uiGroups in the same physical place on the Panel and make only one visible at any one time.
Attachments:

1. Tabs-Experiment.bpanelz
2. Groups-Experiment.bpanelz

DONALDOOG

Thanks for that Spiffo!
Something weird going on with that Tabs-Experiment panel though.
When you first open it, there's a tab group inside another one, but when you click on the comboBox it disappears forever!

SPIFFO

Strange, I just opened it there and it works fine for me!
I've noticed that with the uiGroups you have to be careful not to accidently move one that's on top of another because if you do it gets automatically assigned to the lower group again and the whole thing stops working again.

Basically you're right, probably better off with layers, with that in mind, some questions:
1 When in Layer Editor the Layer listed at the bottom is the only one you can Edit, is that the only consequence of the list order?
2 Does a Layer always cover the entire area of the Panel, with you're example they seem to?
3 If I used layers then personally I would want certain areas of the Panel to remain unaffected whilst other localised areas changed layers depending on something else, is that possible?

OK I'll answer some of this myself, please find attached a Layers Experiment.
The Order of the Layers is important even when not in Edit Mode, as it affects what is Visible and what is NOT Visible.
As far as I can work out, Layers at the bottom of the Layer Editor window are in the Foreground and Layers at the top of the list are in the Background, so if you have Modulators in the same physical X, Y space then you need to make sure that any modulator which needs to be visible on top of another is lower down in the list, closer to the Foreground if you will, or it could get hidden behind!

For instance with this panel attached if you move Layers 3 or 4 up higher in the list than Layers 1 and 2 then they end up getting hidden behind the tabs (which are on Layers
Attachments:
1. Layers-Experiment.bpanelz

DNALDOOG

Hi Spiffo,
That's right – a layer covers the whole panel, but if a component is not covered by another object in a layer above it, it can be accessed, so with a bit of design, so it's actually a very useful feature.
Also, in those if else statements, that last else is not necessary.
function switch_tab_contents()

```
mod1 = panel:getModulatorByName("modulator-2"):getModulatorValue()
if mod1 == 0 then
panel:getCanvas():getLayerByName("Layer3"):setVisible(true)
panel:getCanvas():getLayerByName("Layer4"):setVisible(false)
elseif mod1 == 1 then
panel:getCanvas():getLayerByName("Layer3"):setVisible(false)
panel:getCanvas():getLayerByName("Layer4"):setVisible(true)
else end
end — function
```

Hey Spiffo,
here's another simple panel that uses layers. I posted it here

https://ctrlr.org/forums/topic/did-i-mess-up-my-panel-size/
but will attach it here again.

Attachments:

1. roland-fa06-dnaldoog_1_0_Hell-O-Kitty_2020-10-21_12-21-1.bpanelz

I created 4 layers for four partials, each with a separate control.
A global variable k3 is set in an init script that changes the sysex for each partial and is changed by the uiCombo callback function.
It should give you a start for working out how to change sysex depending on which partial is selected. It might be safer to just edit each sysex value per partial manually, but that will be a lot of work.
I haven't consulted the manual for correct figures.
I put the offsets into a lua table partialOffset={0x20,0x40,0x60,0x7f} with arbitrary figures. Unfortunately this does involve lua!
Regards,
John