

---

Subject: GLCtrl and Docking: do they work together as expected?

Posted by [281264](#) on Tue, 14 Sep 2010 11:45:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I have spotted a, IMO, weird behaviour of several GLCtrls when working with Docking. Please test attached code. Two GLCtrl structs are created: one is made dockable. When this one is made auto-hide and you further take advantage of it, the main Topwindow (which contains the other GLCtrl) fails to re-paint correctly.

I have noticed that GLPane::Init() function is called every time the auto-hide feature is used, and consequently a new OpenGL rendering context is generated and made current. Is this a desirable behaviour? Why is going on this weird behaviour?.

For example, when the other dockable controls (trees) are made auto-hide, this feature does not interfere at all with the OpenGL Ctrl in the main application Topwindow.

Cheers,

Javier

---

### File Attachments

1) [prueba\\_OpenGL\\_DockWindow.7z](#), downloaded 280 times

---

---

Subject: Re: GLCtrl and Docking: do they work together as expected?

Posted by [mrjt](#) on Tue, 14 Sep 2010 14:16:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Autohide and GLCtrl really aren't very compatible. The Autohide uses a popup control, which means the GLCtrl is moving from one native window to another. This is what's causing the repeated GLInit calls and there isn't anything you can do to stop it since it required for the GLCtrl to work

My advice would be to disable Autohide either on the whole app or just the DockableCtrls that hold GLCtrls.

---

---

Subject: Re: GLCtrl and Docking: do they work together as expected?

Posted by [281264](#) on Tue, 14 Sep 2010 14:50:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Roger. I shall do that.

Allow me to ask some more questions (I have poste one of them in this forum):

1.- How is Docking dealing with the OpenGL rendering context creation? I guess that for using the main TopWindow as OpenGL device context, GLCtrl does it (this is obvious); but what about the hRc created?

2.- Then if we have two OpenGL hRC, which of them is the current?

Allow me to ask some more questions (I have poste don of them in this forum):

1.- How is Docking dealing with the OpenGL rendering context creation? I guess that for using the main TopWindow as OpenGL device context, GLCtrl does it (this is obvious); but what about the

2.- Then if we have two OpenGL hRC, which of them is the current?

rendering/interaction with the user?

4.- How a multithreading feature can work in this context?

5.- When a dockable ctrl is not docked, how is it considered? Like a new TopWindow?

Many thanks.

Best wishes,

Javier

Many thanks.

Best wishes,

Javier

---

Subject: Re: GLCtrl and Docking: do they work together as expected?

Posted by [281264](#) on Tue, 14 Sep 2010 17:21:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

An additional post to the previous:

How a GLCtrl of the attached example can access one of the others controls? For instance, If I desired to write a message in one of the arrayctrls after a click in the GLCtrl window is made, how to do it?

Cheers,

Javier

---

Subject: Re: GLCtrl and Docking: do they work together as expected?

Posted by [mrjt](#) on Thu, 16 Sep 2010 08:45:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

1- Yes

2- When Painting begins wglMakeCurrent is called to make the GL context associated with the GLCtrl active. wglGetContext can determine which context is active, but this is really only useful for checking that you have a context active at all (modal windows can change the context for instance).

3- wglMakeCurrent. This is not currently exposed by the standard GLCtrl. I suggest just branching GLCtrl as I did in the package I posted in the other thread: 

```
void StartDC() const {
wglMakeCurrent(GetDC(), GetRC()); }
```

```
void StopDC() const { wglMakeCurrent(NULL, NULL); }
```

 There will be other stuff you want to add also.

4- You will have to synchronise as OpenGL (ASFAIK) will only accept calls from the main thread.

5- Yes

6- Design decision. Store the controls globally, create an event system, use callbacks.

7- As above.

---

Subject: Re: GLCtrl and Docking: do they work together as expected?

Posted by [281264](#) on Thu, 16 Sep 2010 10:16:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Many thanks for the answer.

With regard to the visibility of GLCtrls, I have created pointers, so every control points the others. With this, one control can control the others, even being a TopWindow under Docking.

What amazes me is why auto-hide distorts the central GLCtrl.

Best wishes,

Javier