
Subject: A (maybe) bug in GLControl (on Ubuntu)
Posted by [mdelfede](#) on Sun, 14 Oct 2007 20:26:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Following the example of making an app with more than one topwindow, i changed the Opengl example in reference like this :

```
GUI_APP_MAIN
{
    MyAppWindow *win, *win2;
    win = new MyAppWindow;
    win2 = new MyAppWindow;
    OpenGLExample gl, gl2;
    gl.SetFrame(InsetFrame());
    gl2.SetFrame(InsetFrame());
    win->Add(gl.HSizePos(10, 10).VSizePos(10, 10));
    win2->Add(gl2.HSizePos(10, 10).VSizePos(10, 10));
    win->Sizeable().Zoomable();
    win2->Sizeable().Zoomable();
    win->OpenMain();
    win2->OpenMain();
    Ctrl::EventLoop();
}
```

That works, but hangs on window closing. It even hangs the debugger so I've to kill the process. Investigating a bit more, I've seen that the destructor for GLControl NEVER get called. If I change the code like that :

```
GUI_APP_MAIN
{
    MyAppWindow *win, *win2;
    win = new MyAppWindow;
    win2 = new MyAppWindow;
    OpenGLExample gl, gl2;
    gl.SetFrame(InsetFrame());
    gl2.SetFrame(InsetFrame());
    // win->Add(gl.HSizePos(10, 10).VSizePos(10, 10));
    // win2->Add(gl2.HSizePos(10, 10).VSizePos(10, 10));
    win->Sizeable().Zoomable();
    win2->Sizeable().Zoomable();
    win->OpenMain();
    win2->OpenMain();
    Ctrl::EventLoop();
}
```

It works (without opengl window on screen, of course) and destructors are called as expected. It seems that in former code the event loop still runs on closed windows.

I've the feeling that closing the windows don't detach the child OpenGLExample controls inside, but I'm not sure about it.

If OpenGLExample control is not derived from GLControl all goes ok.

Ciao

Max

EDIT : It seems to stay in EventLoop because of

```
GetTopCtrls().GetCount()
```

returning 1 even when last top window is closed.

Subject: Re: A (maybe) bug in GLControl (on Ubuntu)

Posted by [mirek](#) on Wed, 17 Oct 2007 17:05:34 GMT

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

mdelfede wrote on Sun, 14 October 2007 16:26 Following the example of making an app with more than one topwindow, i changed the Opengl example in reference like this :

GUI_APP_MAIN

```
{
MyAppWindow *win, *win2;
win = new MyAppWindow;
win2 = new MyAppWindow;
OpenGLExample gl, gl2;
gl.SetFrame(InsetFrame());
gl2.SetFrame(InsetFrame());
win->Add(gl.HSizePos(10, 10).VSizePos(10, 10));
win2->Add(gl2.HSizePos(10, 10).VSizePos(10, 10));
win->Sizeable().Zoomable();
win2->Sizeable().Zoomable();
win->OpenMain();
win2->OpenMain();
    Ctrl::EventLoop();
}
```

That works, but hangs on window closing. It even hangs the debugger so I've to kill the process. Investigating a bit more, I've seen that the destructor for GLControl NEVER get called.

Do you "delete" the window somewhere?

Mirek

Subject: Re: A (maybe) bug in GLControl (on Ubuntu)
Posted by [mdelfede](#) on Thu, 18 Oct 2007 10:29:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Wed, 17 October 2007 19:05

Do you "delete" the window somewhere?

Of course, as in example, in MyAppWindow destructor....

At first I tried also to delete it at the end, but that of course doesn't work because of eventloop still running.

As I said, if the control is NOT derived from GLCtrl, all that works. I think that's something missing on GLCtrl class, but I had no time to investigate it more in depth.
Or, maybe, something missing in core classes when a control is built with a window handler as in GLCtrl.

Ciao

Max

Subject: Re: A (maybe) bug in GLControl (on Ubuntu)
Posted by [mirek](#) on Thu, 18 Oct 2007 12:37:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

mdelfede wrote on Thu, 18 October 2007 06:29luzr wrote on Wed, 17 October 2007 19:05

Do you "delete" the window somewhere?

Of course, as in example, in MyAppWindow destructor....

At first I tried also to delete it at the end, but that of course doesn't work because of eventloop still running.

As I said, if the control is NOT derived from GLCtrl, all that works. I think that's something missing on GLCtrl class, but I had no time to investigate it more in depth.
Or, maybe, something missing in core classes when a control is built with a window handler as in GLCtrl.

Ciao

Max

Hm, do you think you could post the complete package (to save the time recreating the testcase....)?

Mirek

Subject: Re: A (maybe) bug in GLControl (on Ubuntu)
Posted by [mdelfede](#) on Thu, 18 Oct 2007 13:06:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Thu, 18 October 2007 14:37

Hm, do you think you could post the complete package (to save the time recreating the testcase....)?

Of course, here is it.

But it's really simple, It's a copy of the tutorial's example with multiple main windows combined with the OpenGL example.

Pay attention, I modified on-the-fly the OpenGL example, so it has the same name of the original Opengl reference package... I'm not at home so I have not the original package.

But this one behaves exactly the same.

Ciao

Max

File Attachments

1) [OpenGL.zip](#), downloaded 362 times

Subject: Re: A (maybe) bug in GLControl (on Ubuntu)
Posted by [mdelfede](#) on Mon, 22 Oct 2007 10:38:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Some news about this ?

Subject: Re: A (maybe) bug in GLControl (on Ubuntu)
Posted by [mdelfede](#) on Fri, 26 Oct 2007 18:51:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, investigating a bit more, I've seen that if control is derived from GLControl, the

GetTopCtrls().GetCount() still returns 2 controls (in my example with 2 windows...) even on window close, so the Ctrl::EventLoop() never returns.

I'll try to investigate a bit more this bug....

Ciao

Max

Subject: [SOLVED] A (maybe) bug in GLControl (on Ubuntu)

Posted by [mdelfede](#) on Sat, 27 Oct 2007 18:47:31 GMT

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

Well, at the end, I found a solution for that.

The GLCtrl::State() function must be extended with a call of CloseGL() on CLOSE state :

```
void GLCtrl::State(int reason)
{
    if( InitializationProblem )
        return;

    if( IsInitialized )
    {
        switch( reason )
        {
            case SHOW:
            {
                if( IsShown() && !IsMapped )
                    MapWindow();

                if( !IsShown() && IsMapped )
                    UnMapWindow();
            }; break;

            case LAYOUTPOS:
            {
                MoveSubWindow();
                OpenGLResize();
            }; break;

            case CLOSE:
            {
                CloseGL();
            }; break;
        }
    }
}
```

added-->

```
}; break;
```

```
<--added
```

```
    default:  
        break;  
    }  
}  
else  
    if( GetTopWindow() && GetTopWindow()->GetWindow() )  
        OpenGL();  
}
```

Whith this change, the list of top controls is updated correctly closing windows that contains GLControl.

Ciao

Max

Subject: Re: [SOLVED] A (maybe) bug in GLControl (on Ubuntu)
Posted by [mirek](#) on Sat, 27 Oct 2007 19:20:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks, patch applied.

Mirek