
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.
