Subject: Re: Core chat...
Posted by mdelfede on Sat, 27 Oct 2007 13:21:15 GMT
View Forum Message <> Reply to Message

I did ask the former question because I was lookin' inside MainWindow code.... Looking for the OpenGL bug.
But then I realized that Ctrl::Add() is quite different from Array::Add(), building an array of references instead of objects.

BTW, I still didn't find the bug there... The only thing I found (up to now) is shown in my code here :

```
 int zzz;
 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();

 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 0
 win->OpenMain();
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 1
 win2->OpenMain();
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 2
 delete win;
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 2 !!!
 delete win2;
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 2 !!!
   Ctrl::EventLoop();
```

If I suppress the lines :

```
 win->Add(gl.HSizePos(10, 10).VSizePos(10, 10));
 win2->Add(gl2.HSizePos(10, 10).VSizePos(10, 10));
```

The code works ok :

```
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 0
 win->OpenMain();
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 1
 win2->OpenMain();
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 2
 delete win;
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 1
 delete win2;
 zzz = Ctrl::GetTopCtrls().GetCount(); // zzz = 0
```

That works even if I leave both lines BUT OpenGLExample is *not* derived from GLControl.
I'd like to know if the bug is Linux-dependent or not... But I haven't an Ide setup on my win xp machine. Don't you have a bit time to test on windows ?

Back to refcounted objects. What about if Ctrl would be an object built with PIMPL idiom and refcounted ? You then could write :

```
aControl a;  // control is created
aControl b = a; // just reference to inner pimpl object is copied
```

or, what sound even better:

```
Vector<Ctrl>*a, *b;
Ctrl c; // control is created, RefCount == 1
a = new Vector<Ctrl>;
b = new Vector<Ctrl>;
a->Add(c); // a gets a *copy* of c, but in reality it adds just to refcount of c, that becomes 2
b->Add(c); // b gets a *copy* of c, but in reality it adds just to refcount of c, that becomes 3
delete a; // a gets destroyed, RefCount in c becomes 2
```

The advantage of this instead of references of an object ? Well... you must not care of ownership.... and you can be sure object is freed on last reference lost.
As usual, that brings some performance lost.

Ciao

Max