Subject: Re: Proposed change to U++ to allow owning children.
Posted by Lance on Fri, 18 Mar 2011 15:22:08 GMT
View Forum Message <> Reply to Message

Now we have a much cleaner solution thanks to Kohait00's input.

Brief list of changes to the current libary code:

In CtrlCore.h

Added to Ctrl class

```
   bool owned : 1;
```

And 2 public members

```
 bool   IsOwned()const{ return owned; }
 Ctrl&   Owned(bool v=true){ owned=v; return *this; }
```

In Ctrl.cpp, initialized owned flag to false

```
Ctrl::Ctrl() {
 GuiLock __;
 LLOG("Ctrl::Ctrl");
 destroying = false;
 owned = false;  // <-----This line is added
 parent = prev = next = firstchild = lastchild = NULL;
 top = NULL;
 exitcode = 0;
 frame.Add().frame = &NullFrame();
 enabled = visible = wantfocus = initfocus = true;
 editable = true;
// GLX = false;
#ifdef PLATFORM_WIN32
 activex = false;
 isdhctrl = false;
#endif
 backpaint = IsCompositedGui() ? FULLBACKPAINT : TRANSPARENTBACKPAINT;
 inframe = false;
 ignoremouse = transparent = false;
 caretcx = caretcy = caretx = carety = 0;
 SetRect(Rect(0, 0, 0, 0));
 inloop = popup = isopen = false;
 modify = false;
 unicode = false;
 popupgrab = false;
 fullrefresh = false;
 akv = false;
 hasdhctrl = false;
}
```

And in CtrlChild.cpp

```
// @param: q , the child to be added
//         q,  an existing child to precede p
void  Ctrl::AddChild(Ctrl *q, Ctrl *p)
{
 GuiLock __;
 ASSERT(q);

 LLOG("Add " << UPP::Name(q) << " to: " << Name());
 if(p == q) return;
 bool updaterect = true;

 // remember and change
 bool owned=q->owned;
 q->Owned(false); // that way it's guarenteed not to be
    // accidently delete'd when possibly changing parents


 if(q->parent) {
  ASSERT(!q->inframe);
  if(q->parent == this) {
   RemoveChild0(q);
   updaterect = false;
  }
  else
   q->parent->RemoveChild(q);
 }
 q->parent = this;
 if(p) {
  ASSERT(p->parent == this);
  q->prev = p;
  q->next = p->next;
  if(p == lastchild)
   lastchild = q;
  else
   p->next->prev = q;
  p->next = q;
 }
 else
  if(firstchild) {
   q->prev = NULL;
   q->next = firstchild;
   firstchild->prev = q;
   firstchild = q;
  }
  else {
   ASSERT(lastchild == NULL);
   firstchild = lastchild = q;
   q->prev = q->next = NULL;
  }

 // succesfully added as children of *this, now
 // it's perfect time to restore saved owned flag
 q->Owned(owned);

 q->CancelModeDeep();
 if(updaterect)
  q->UpdateRect();
 ChildAdded(q);
 q->ParentChange();
 if(updaterect && GetTopCtrl()->IsOpen())
  q->StateH(OPEN);
 if(dynamic_cast<DHCtrl *>(q))
  SyncDHCtrl();
}

.....

void  Ctrl::RemoveChild0(Ctrl *q)
{
 GuiLock __;
 ChildRemoved(q);
 q->DoRemove();
 q->parent = NULL;
 if(q == firstchild)
  firstchild = firstchild->next;
 if(q == lastchild)
  lastchild = lastchild->prev;
 if(q->prev)
  q->prev->next = q->next;
 if(q->next)
  q->next->prev = q->prev;
 q->next = q->prev = NULL;

 if(dynamic_cast<DHCtrl *>(q))
  SyncDHCtrl();
 // code added to allowed owned child****
 if(q->owned)
  delete q;
 // end code added by Lance
```

```
}

void  Ctrl::RemoveChild(Ctrl *q)
{
 GuiLock __;
 if(q->parent != this) return;
 q->RefreshFrame();

 bool owned=q->IsOwned();
 q->Owned(false); // we still need it to be alive

 RemoveChild0(q);
 q->ParentChange();
 if(GetTopCtrl()->IsOpen())
  q->StateH(CLOSE);

 if( owned )
 {
  delete q; // this is why the new'd-only requirement.
 }
 // no need to restore q's owned flag, it's either destroyed or
 // its owned flag is correctly set
}
```

File Attachments

1) CtrlCore.rar, downloaded 395 times