
Subject: Re: Window without title bar

Posted by [Werner](#) on Wed, 13 Feb 2008 20:06:04 GMT

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

Meanwhile I finished my "single"-ruminations.

At the same time I improved my popup-window test application.

Perhaps the code is interesting to you because it deals with the possible GUI effects "slide" and "fade".

Here it is:

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
class PopUpWin : public LineEdit
```

```
{  
private:  
    bool poppedUp_;
```

```
    void RightDown(Point p, dword keyFlags)  
    {  
        Close();  
        poppedUp_ = false;  
    }
```

```
public:  
    PopUpWin() : poppedUp_(false) { }  
    void SetPoppedUp(bool yesNo) { poppedUp_ = yesNo; }  
    bool IsPoppedUp() const { return poppedUp_; }  
};
```

```
class MainWin : public TopWindow
```

```
{  
private:  
    typedef MainWin CLASSNAME;
```

```
    void NoEffect()  
    {  
        Single<PopUpWin>().SetPoppedUp(true);  
        Single<PopUpWin>().SetRect(300, 300, 600, 400);  
        Single<PopUpWin>().SetColor(TextCtrl::PAPER_NORMAL, White);  
        Single<PopUpWin>().SetColor(TextCtrl::INK_NORMAL, Black);  
        Single<PopUpWin>().PopUp  
        (  

```

```

    this,      // Ctrl* owner = NULL
    true,      // bool savebits = true
    true,      // bool activate = true
    true,      // bool dropshadow = false
    false     // bool topmost = false
);
}

void SlideEffect()
{
    Single<PopUpWin>().SetPoppedUp(true);
    Single<PopUpWin>().SetRect(300, 300, 6, 4);
    Single<PopUpWin>().SetColor(TextCtrl::PAPER_NORMAL, White);
    Single<PopUpWin>().SetColor(TextCtrl::INK_NORMAL, Black);
    Single<PopUpWin>().PopUp
    (
        this,      // Ctrl* owner = NULL
        true,      // bool savebits = true
        true,      // bool activate = true
        true,      // bool dropshadow = false
        false     // bool topmost = false
    );
    Ctrl::ProcessEvents();
    Animate(Single<PopUpWin>(), RectC(300, 300, 600, 400), GUIEFFECT_SLIDE);
}

void FadeEffect()
{
    Single<PopUpWin>().SetPoppedUp(true);
    Single<PopUpWin>().SetRect(300, 300, 600, 400);
    Single<PopUpWin>().SetColor(TextCtrl::PAPER_NORMAL, White);
    Single<PopUpWin>().SetColor(TextCtrl::INK_NORMAL, Black);
    Single<PopUpWin>().PopUp
    (
        this,      // Ctrl* owner = NULL
        true,      // bool savebits = true
        true,      // bool activate = true
        true,      // bool dropshadow = false
        false     // bool topmost = false
    );
    Ctrl::ProcessEvents();
    Animate(Single<PopUpWin>(), RectC(300, 300, 600, 400), GUIEFFECT_FADE);
}

void local_menu(Bar& bar)
{
    MenuBar local_menu;

```

```
bar.Add("no effect", THISBACK(NoEffect));
bar.Add("slide effect", THISBACK(SlideEffect));
bar.Add("fade effect", THISBACK(FadeEffect));
local_menu.Execute();
}
```

```
void LeftDown(Point p, dword keyFlags)
{
    if (Single<PopUpWin>().IsPoppedUp())
        return;
    MenuBar::Execute(THISBACK(local_menu));
}
};
```

```
GUI_APP_MAIN
{
    MainWin mainWin;
    mainWin.SetRect(0, 0, 800, 600);
    mainWin.Run();
}
```

Werner
