
Subject: Re: Focus problem

Posted by [Silvan](#) on Sun, 19 Dec 2021 11:23:17 GMT

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

Here is a working code as intended:

some thought:

- 1) I used a new class derived from ImageCtrl, I suppose that is a little bit overkill for a simply overload of two method/event;
- 2) I had to use a global variable (Point p) otherwise not accessible from the new class Mypanel. Really bad.
- 3) I had to use the override Close method otherwise the program terminate with an error.

I suppose that U++ allow to do all this much better... I'm searching how hoping in some help!

Thank you

Silvan

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

using namespace Upp;

Point p;

struct MyPanel : ImageCtrl {

    virtual void Paint(Draw& w)override
    {
        w.DrawRect(GetSize(), White());
        w.DrawText(p.x, p.y, "#", Arial(30), Red);
        Refresh();
    }

    virtual bool Key(dword key, int count) override
    {
        switch (key)
        {
            case K_W:
                p.y-=1;
                break;
            case K_S:
                p.y+=1;
                break;
            case K_A:
                p.x-=1;
                break;
            case K_D:
                p.x+=1;
                break;
        }
    }
}
```

```
p.x+=1;
break;
default:
;
}
Refresh();
    return true;
}

void LeftDown(Point p, dword keyflags) override
{
    SetFocus();
}

};

struct MainWindow : TopWindow {

EditString inputtext;
MyPanel panel;

void Close() override
{
    delete this;
}

// Costruttore dove inserisci le inizializzazioni
MainWindow()
{
    Title("Test Focus").Zoomable().Sizeable();
    Add(inputtext.TopPosZ(0, 16).HSizePos());
    Add(panel.VSizePos(26, 0).HSizePos(0, 0));

    inputtext <<= "test";

    SetRect(0, 0, 300, 300);
    p.x = 150;
    p.y = 150;
}
};

GUI_APP_MAIN
{
(new MainWindow)->OpenMain();
Ctrl::EventLoop();
}
```
