
Subject: Re: Focus problem

Posted by [Lance](#) on Sat, 18 Dec 2021 04:39:50 GMT

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

Hi Silvan:

I manage to create the following example.

THISBACK is legacy code. Event<>, Gate, etc now are alias of Function.

In IDE, Ctrl+Left Click on the Event type name will bring you to its definition, you should see something like this

```
template <typename... ArgTypes>
using Event = Function<void (ArgTypes...)>;
```

```
template <typename... ArgTypes>
using Gate = Function<bool (ArgTypes...)>;
```

```
template <class Ptr, class Class, class Res, class... ArgTypes>
Function<Res (ArgTypes...)> MemFn(Ptr object, Res (Class::*method)(ArgTypes...))
{
    return [=](ArgTypes... args) { return (object->*method)(args...); };
}
```

BTW, revised code to do what you asked and demonstrate a callback.

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

```
using namespace Upp;
```

```
struct MainWindow : TopWindow {
    Point p;
    EditString inputtext;

    virtual void Paint(Draw& w)override
    {
        int x,y;

        w.DrawRect(GetSize(), SWhite());

        w.DrawText(p.x, p.y, "#", Arial(30), Red);

    }
}
```

```

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

// void Close()override
// {
// // delete this;
// }

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;
    case K_UP:
    case K_DOWN:
    case K_LEFT:
    case K_RIGHT:
        WhenSuspiciousKey();
    }

    Refresh();
    return true;
}

// Costruttore dove inserisci le inizializzazioni
MainWindow()
{

    Title("Test Focus").Zoomable().Sizeable().WantFocus();
    Add(inputtext.TopPosZ(0, 16).HSizePos());
    inputtext <<= "test";

    SetRect(0, 0, 300, 300);
}

```

```
p.x = 150;
p.y = 150;
}
```

```
Event<> WhenSuspiciousKey; // accepts a callback with prototype like
// void keypressed();
};
GUI_APP_MAIN
{
    MainWindow m;
    m.WhenSuspiciousKey<<[] { PromptOK("Use upper case W,A,S,D to navigate!"); };
    m.Run();
    //
    // MainWindow *m=new MainWindow;
    // m->OpenMain();
    /// m->SetFocus();
    /// DUMP(m->HasFocus());
    // Ctrl::EventLoop();
}
```

BTW, there is nothing wrong with allocating your MainWindow or other Ctrl's from stack. But in U++, you don't have to. In your case, there is no reason to do it that way.
