
Subject: App "swallowing" mouseclicks
Posted by [peterh](#) on Tue, 30 Aug 2022 04:21:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi I compiled the tutorial sample "Gui21" "creating and using custom widgets".
It derives a widget from "Ctrl".

Just the code as a reminder:

```
struct MyCtrl : public Ctrl {
    int count = 0;

    virtual void Paint(Draw& w) override {
        w.DrawRect(GetSize(), White());
        w.DrawText(2, 2, AsString(count));
    }

    virtual void LeftDown(Point, dword) override {
        count++;
        Refresh();
    }
};
```

I made then an experimental program, deriving not from "Ctrl", but deriving from "EditString".
Using this, I noticed, it does pretty often "swallow" mouseclicks, this means it does not react to them.

Then I tried the original code again and this "swallows" mouseclicks too, not so often but randomly about 1 out of 10 clicks.

It happens (reliably) both in debug and release mode.
I am on nightly build 16323, Win10 64 bit, using Clang 64 bit.

The computer is pretty fast, Ryzen 7 and CPU load in Task Manager is close to zero on all 16 cores.

Testproject code:

Main:

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

```
using namespace Upp;
```

```
struct MyCtrl : public EditString {
    int count = 0;

    virtual void Paint(Draw& w) override {
        //w.DrawRect(GetSize(), White());
        EditString::Paint(w);
    }
};
```

```

w.DrawText(2, 2, AsString(count));
}

virtual void LeftDown(Point p, dword d) override {
    count++;
    //Refresh();
    EditString::LeftDown(p,d);
}

};

#define LAYOUTFILE <Gui21/Gui21.lay>
#include <CtrlCore/lay.h>

struct Gui21 : public WithGui21Layout<TopWindow> {
    Gui21();
};

Gui21::Gui21()
{
    CtrlLayout(*this, "Window title");
}

GUI_APP_MAIN
{
    Gui21().Run();
}

CtrlLib.usc
ctrl MyCtrl {
    >EditString;
}
Gui21.lay
LAYOUT(Gui21Layout, 216, 144)
ITEM(MyCtrl, myctrl, Tip(t_("Test")).LeftPosZ(44, 120).TopPosZ(36, 52))
ITEM(Upp::EditString, dv___1, LeftPosZ(44, 64).TopPosZ(96, 19))
END_LAYOUT

```

It "swallows" fast mouseclick sequences preferably.

Edit:

I believe, I know the reason: EditString "swallows" clicks, when it detects a double click.

Subject: Re: App "swallowing" mouseclicks

Posted by [jjacksonRIAB](#) on Wed, 31 Aug 2022 16:51:48 GMT

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

If you don't feel you're double-clicking and it's showing that event anyway, I wouldn't rule out a mouse issue. As microswitches age the metal contacts within them begin to wobble and bounce on the contact. Various OSes have software to try to correct for this that they call a debouncer but it's not perfect. Eventually the only solution is to replace the switch as it begins creating double-clicks events too frequently to ignore.

I only bring this up because I've had the same mouse for around 16 years and I have to replace the LMB microswitch every 2-3 years to keep it going. If you have a spare mouse or want to check a different button to test this, it's worth considering.

Subject: Re: App "swallowing" mouseclicks

Posted by [peterh](#) on Wed, 31 Aug 2022 17:20:34 GMT

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

Thank you. I do not think so. The mouse is about 1/2 year old and I do not see this with other applications.

Also the effect is dramatically reduced, if I do not derive from EditField, but from Ctrl or if I click very slowly, e.g. in a 2 second distance. Maybe it vanishes completely then, I am not sure.

If I click as fast as I can, it swallows almost all clicks.

It is not very important for me. Finally I do not intend to override Leftdown, but other methods. I do this just for learning how to do it and to begin somewhere.

Subject: Re: App "swallowing" mouseclicks

Posted by [jjacksonRIAB](#) on Wed, 31 Aug 2022 17:41:10 GMT

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

Just checked your example - yes, you are probably registering a double-click.

Try all three separately:

```
void LeftDouble(Point p, dword d) override {
    count++;
    EditString::LeftDouble(p, d);
}
```

```
void LeftDown(Point p, dword d) override {
    count++;
    EditString::LeftDown(p,d);
}
```

```
void LeftUp(Point p, dword d) override {  
    count++;  
    EditString::LeftUp(p, d);  
}
```

If you just do LeftUp you'll probably see it works reliably. No events are being lost.

Subject: Re: App "swallowing" mouseclicks
Posted by [peterh](#) on Wed, 31 Aug 2022 17:45:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, confirmed, that clears it up for me.
Thank you.
