Subject: [SOLVED] MouseMove events in Windows and Linux
Posted by pvictor on Thu, 06 Aug 2020 06:42:37 GMT

View Forum Message <> Reply to Message

Hello.

I've made a small app that allows moving and resizing graphic data with mouse.
It works fine under Linux.
However, under Windows, it works fine only when the app window is small.
When increasing the window size, starting from a certain size, moving and resizing becomes uncomfortable.
Here's a small testcase to reproduce. You can set delay and move the mouse over the window.

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

using namespace Upp;

int delay;

struct MyImageCtrl: ImageCtrl {
 virtual void MouseMove(Point pos, dword flags) {
  Sleep(delay);
  Size sz = GetSize();
  ImagePainter ip(sz.cx, sz.cy);
  ip.DrawRect(0, 0, sz.cx, sz.cy, White());
  ip.DrawRect(pos.x, pos.y, 50, 50, Black());
  SetImage(ip);
 }
};

struct MyApp : TopWindow {
 EditIntSpin ei;
 MyImageCtrl img;
 MyApp() {
  Zoomable().Sizeable();
  Add(ei.Min(0).LeftPos(10,60).TopPos(10,20));
  Add(img.HSizePos(10,10).VSizePos(40,10).SetFrame(BlackFrame()));
  ei <<= delay = 10;
  ei << [=] { delay = ~ei; };
 }
};

GUI_APP_MAIN
{
 MyApp().Run();
}
```

Sleep(delay) imitates calculating and drawing of complex data.

Under Linux (GCC), it works adequately even with delay=100.
Under Windows (CLANG and MSBT17) it works fine only with delay=0..6 and very ugly with delay>=7 (on my computer).
Try it and you'll see what I mean.
Is this a Windows property?
Or is this a bug in UPP?

Best regards.
Victor

Subject: Re: MouseMove events in Windows and Linux
Posted by mirek on Thu, 06 Aug 2020 11:18:35 GMT

View Forum Message <> Reply to Message

Short answer: Do not put too much processing into event handling routines...

Longer: ...especially MoiseMove, because that will stall the event queue...

There might be some differences between event queue in gtk and Win32 that make Linux work better here, but the fundamental reason is the same and the code simply should not do what it is doing. You might get better results by adding Sync after SetImage, but still this should be coded differently.

Subject: Re: MouseMove events in Windows and Linux
Posted by pvictor on Thu, 06 Aug 2020 13:45:10 GMT

View Forum Message <> Reply to Message

mirek wrote on Thu, 06 August 2020 14:18Short answer: Do not put too much processing into event handling routines...

Longer: ...especially MoiseMove, because that will stall the event queue...

There might be some differences between event queue in gtk and Win32 that make Linux work better here, but the fundamental reason is the same and the code simply should not do what it is doing. You might get better results by adding Sync after SetImage, but still this should be coded differently.
Thank you, in my case Sync() really helps.
BTW, I believe that OS is smart enough to not place the bunch of MouseMove events into the queue. It should replace an old unprocessed MouseMove event with the new one. In this case a few dozens ms of event processing should be OK.

Best regards,
Victor