
Subject: Re: Using Pen with U++
Posted by [mirek](#) on Sun, 04 Apr 2021 18:42:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tom1 wrote on Sun, 04 April 2021 20:35Hi Mirek,

Thanks for accepting the changes.

It was a stupid mistake of me to use `Image::Arrow()` instead of `CursorImage()`. It should be in `CtrlMouse.cpp(145)` like:
`if(supports_pen && is_pen_event) return CursorImage(p, keyflags); // Avoid duplicated pen events`

Without 'supports_pen' it would become a tedious battle in client code to avoid various pen originated duplicate mouse events, so I rather block them just before. Anyway, making 'bool supports_pen;' a configurable flag in `Ctrl::` in `CtrlCore.h` is fine with me:

I understand that, but would that be really that hard? Esp. now that we have `K_PEN`?

Quote:

```
Ctrl& EnablePenSupport(bool b = true)          { supports_pen = b; return *this; }
```

Then, to avoid `Pen()` return value assignment in `Win32Proc.cpp(147)`, we should remove 'supports_pen =' and have just:
`q->Pen(p, pen, GetMouseFlags());`

This works here. (I will of course add 'EnablePenSupport();' to constructors of `Pen()` enabled widgets.)

This includes `RectTracker::RectTracker()` in `LocalLoop.cpp`:

```
RectTracker::RectTracker(Ctrl& master)
{
    EnablePenSupport();
}
```

Best regards,

Tom

Actually, maybe you could rather not call `EnablePenSupport` in `RectTracker` constructor and leave that to client code?

But then, should not that prevent call of `Pen` virtual method too?

BTW, given what it does, should not that rather be called "DiasbleMouse"? :)

Mirek
