
Subject: Re: Using Pen with U++
Posted by [Tom1](#) on Wed, 24 Mar 2021 15:51:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mirek,

I was only thinking about declaring:

```
virtual bool PenDown(Point p, dword keyflags){ return false; }  
virtual bool PenUp(Point p, dword keyflags){ return false; }  
virtual bool PenMove(Point p, dword keyflags){ return false; }
```

and calling them directly like this

```
switch(message) {  
case WM_POINTERDOWN:  
ClickActivateWnd();  
if(PenDown(Point(p), GetMouseFlags())) return 0L;  
break;  
case WM_POINTERUP:  
if(PenUp(Point(p), GetMouseFlags())) return 0L;  
break;  
case WM_POINTERUPDATE:  
if(PenMove(Point(p), GetMouseFlags())) return 0L;  
break;  
}  
break;
```

...and using the Pen -functions you made earlier.

However, my simple approach above would inevitably have its shortcomings as properly tracking the pen/mouse position (e.g. GetMousePos()) cannot be taken care of this way. RectTracker would likely also be broken in this case.

Despite working several tens of hours on this, my understanding on Ctrl and mouse processing is not sufficient yet to do it the right way. (Despite the several tens of hours I have spent on this during the few last days. Shame on me.)

So, absolutely, if you have time, please do it the right way. (I have a strong feeling that it will eventually save time for both of us.)

In any case, I'll be here ready to test as soon as you post an update.

Best regards,

Tom

EDIT: BTW: How did you think about handling barrel-button and other flags? After playing around with the pen, I have noticed that it is very difficult to click barrel and other buttons on the pen while keeping it steady. So, IMO, barrel should be just a flag to read when pen goes down/up or moves.

Just like Shift/Ctrl/Alt, which are used and needed as modifiers similarly for pen and mouse.

Also, I feel PenMove + PenDown + PenUp make it easier to write code for the pen, instead of a single PenUpdate function.

Further on, readily supporting PenHold, PenDouble (and possibly PenTriple) would be useful. Decoding them in application would require passing timestamps too, which is not very convenient.

EDIT2: Actually, I would rather have PEN defined as 3 or 4, i.e. in parallel to LEFT, RIGHT and MIDDLE. This way we can have PEN|DOWN, PEN|UP, PEN|DOUBLE, ... and just PEN for move.
