
Subject: Re: Using Pen with U++
Posted by [mirek](#) on Tue, 09 Mar 2021 11:04:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Should be implemented in windows. This works on my computer:

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

using namespace Upp;

struct MyApp : TopWindow {
    Point pos;

    Vector<Vector<Tuple<double, Pointf>>> drawing;

    virtual void LeftDown(Point p, dword)
    {
        if(IsPointerPen())
            drawing.Add().Add(MakeTuple(GetPenPressure(), p));
        Refresh();
    }

    virtual void MouseMove(Point p, dword keyflags) {
        pos = p;
        if(IsPointerPen() && drawing.GetCount())
            drawing.Top().Add(MakeTuple(GetPenPressure(), p));
        Refresh();
    }

    virtual void Paint(Draw& w0)
    {
        DrawPainter w(w0, GetSize());
        w.Clear(SColorPaper());

        w.LineCap(LINECAP_ROUND);
        for(const auto& stroke : drawing)
            if(stroke.GetCount())
                for(int i = 0; i < stroke.GetCount() - 1; i++) {
                    w.Move(stroke[i].b);
                    w.Line(stroke[i + 1].b);
                    w.Stroke(DPI(20) * stroke[i].a, Black());
                }

        int fcy = GetStdFontCy();
        int y = 10;
        auto Text = [&] (const String& text) {
            w.DrawText(10, y, text);
        };
```

```

    y += fcy;
};
Text(AsString(pos));
Text(String() << "Pen: " << IsPointerPen());
Text(String() << "Pressure: " << GetPenPressure());
Text(String() << "Rotation: " << GetPenRotation());
Text(String() << "Tilt: " << GetPenTilt());
Text(String() << "Barrel: " << IsPenBarrelPressed());
Text(String() << "Inverted: " << IsPenInverted());
Text(String() << "Eraser: " << IsPenInverted());
Refresh();
}
};

GUI_APP_MAIN
{
    MyApp().Run();
}

```

Cheap tablet I have bought does not have tilt/rotation support so I cannot test it, but the code is quite straightforward so there is a good chance it will work - it is good that you have that HW in yours :)

These changes should not break Win7 compatibility, once again would be nice if you tested that.
