
Subject: Re: Using Pen with U++

Posted by [Tom1](#) on Thu, 11 Mar 2021 13:54:33 GMT

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

OK, here's a fix for the testing code:

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

```
using namespace Upp;
```

```
struct MyApp : TopWindow {  
    Point pos;
```

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

```
    bool pendown;
```

```
    MyApp(){  
        pendown=false;  
    }
```

```
    virtual void LeftUp(Point p, dword)  
    {  
        if(IsPointerPen()) {  
            DLOG("End line");  
            pendown=false;  
        }  
        Refresh();  
    }
```

```
    virtual void LeftDown(Point p, dword)  
    {  
        if(IsPointerPen()) {  
            DLOG("Start line");  
            pendown=true;  
            drawing.Add().Add(MakeTuple(GetPenPressure(), p));  
        }  
        Refresh();  
    }
```

```
    virtual void MouseMove(Point p, dword keyflags) {  
        pos = p;  
        if(IsPointerPen() && drawing.GetCount() && pendown) {  
            DLOG("Drawing line, pressure: " << GetPenPressure());  
            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: " << IsPenEraserPressed()); // FIXED to IsPenEraserPressed()
    //Refresh(); // REMOVED
}
};

GUI_APP_MAIN
{
    MyApp().Run();
}

```

And results for single line draw:

```

WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERDOWN
Start line
WM_POINTERUPDATE

```

Drawing line, pressure: 0.107421875
WM_POINTERUPDATE
Drawing line, pressure: 0.154296875
WM_POINTERUPDATE
Drawing line, pressure: 0.205078125
WM_POINTERUPDATE
Drawing line, pressure: 0.2255859375
WM_POINTERUPDATE
Drawing line, pressure: 0.2255859375
WM_POINTERUPDATE
Drawing line, pressure: 0.2255859375
WM_POINTERUPDATE
Drawing line, pressure: 0.228515625
WM_POINTERUPDATE
Drawing line, pressure: 0.2451171875
WM_POINTERUPDATE
Drawing line, pressure: 0.2451171875
WM_MOUSEMOVE
WM_LBUTTONDOWN
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.244140625
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.263671875
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.2802734375
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.29296875
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.3056640625
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.3193359375

...

Drawing line, pressure: 0.4169921875
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.4052734375
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.37890625
WM_MOUSEMOVE

WM_POINTERUPDATE
Drawing line, pressure: 0.365234375
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.3359375
WM_MOUSEMOVE
WM_POINTERUPDATE
Drawing line, pressure: 0.0625
WM_MOUSEMOVE
WM_POINTERUP
End line
WM_LBUTTONUP
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_MOUSEMOVE
WM_POINTERUPDATE
WM_MOUSEMOVE

Best regards,

Tom
