

---

Subject: Re: Using Pen with U++  
Posted by [mirek](#) on Fri, 26 Mar 2021 07:57:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Thu, 25 March 2021 22:28Mirek,

Is pn.action intentionally merely a state flag showing continuously PEN\_UP or PEN\_DOWN? Or should we reset it along with other PenInfo items in the beginning of processing WM\_POINTER\* messages?

I would like to have it reset, in order to more easily detect UP/DOWN state changes without a state variable and treat zero action as 'PEN\_MOVE'.

Then we could do this to decode all kinds of pen events:

```
#define PENMOVE 0
#define PENDOWN 1
#define PENUP 2
#define PENHOLD 3
#define PENDRAG 4
#define PENDOUBLE 5
#define PENTRIPLE 6
```

```
bool IsNear(Point p, Point o){
    Point delta(p-o);
    if(abs(delta.x)<5 && abs(delta.y)<5) return true;
    return false;
}
```

```
int PenDecoder(Point p, const PenInfo& pn, dword keyflags){
    static int action=0;
    static int downt=0;
    static Point downp=NULL;
    static bool doubleClick=true;
    static bool down=false;
    static bool solved=false;
    int now=msecs();
    int deltat=now-downt;
```

```
bool isNear=IsNear(p,downp);
```

```
switch(pn.action){
case PEN_DOWN:
    down=true;
    solved=false;
    downp=p;
    downt=now;
```

```

if(isNear){
  if(now-downt < 500){
    if(doubleClick){
      return PENTRIPL;
    }
    else{
      doubleClick=true;
      return PENDOUBLE;
    }
  }
}
doubleClick=false;
return PENDOWN;

case PEN_UP:
  down=false;
  return PENUP;

default:
  if(down && !solved){
    if(isNear){
      if(now-downt > 1000){
        solved=true;
        return PENHOLD;
      }
    }
    else{
      solved=true;
      return PENDRAG;
    }
  }
  return PENMOVE;
}
}

```

Best regards,

Tom

Would it solve any real problem? I mean, I believe that the application can do all these on its own, if it needs to.

---