
Subject: Re: Using Pen with U++

Posted by [Tom1](#) on Fri, 30 Apr 2021 15:08:56 GMT

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

Mirek,

This might only apply after release 2021.1, but I found a solution for history -- or rather the backlog of mouse events:

In GtkCreate.cpp on line 44 we should have:

```
gtk_widget_set_events(top->window, GDK_ALL_EVENTS_MASK &  
~GDK_POINTER_MOTION_HINT_MASK);
```

This drops the deprecated event mask for reducing the number of GDK_MOTION_NOTIFY events.

Additionally in GtkEvent.cpp we need:

...

```
static Point s_mousepos;
```

```
Point Ctrl::GetMouseInfo(GdkWindow *win, GdkModifierType& mod)
```

```
{  
#if GTK_CHECK_VERSION(3, 20, 0)  
GdkDisplay *display = gdk_window_get_display(win);  
GdkDevice *pointer = gdk_seat_get_pointer(gdk_display_get_default_seat(display));  
double x, y;  
gdk_window_get_device_position_double(win, pointer, &x, &y, &mod);  
return s_mousepos; //return Point((int)SCL(x), (int)SCL(y));  
#else  
gint x, y;  
gdk_window_get_pointer(win, &x, &y, &mod);  
return Point(SCL(x), SCL(y));  
#endif  
}
```

```
void Ctrl::AddEvent(gpointer user_data, int type, const Value& value, GdkEvent *event)
```

```
{  
if(Events.GetCount() > 50000)  
return;  
GEvent& e = Events.AddTail();  
e.windowid = (uint32)(uintptr_t)user_data;  
e.type = type;  
e.value = value;  
GdkModifierType mod;  
e.mousepos = GetMouseInfo(gdk_get_default_root_window(), mod);  
if(event->type == GDK_MOTION_NOTIFY){  
GdkEventMotion *mevent = (GdkEventMotion *)event;  
e.mousepos = s_mousepos = Point(SCL(mevent->x_root), SCL(mevent->y_root));  
}  
}
```

```

e.state = (mod & ~(GDK_BUTTON1_MASK|GDK_BUTTON2_MASK|GDK_BUTTON3_MASK)) |
MouseState;
e.count = 1;
e.event = NULL;
#if GTK_CHECK_VERSION(3, 22, 0)
GdkDevice *d = gdk_event_get_source_device(event);
int s=d?gdk_device_get_source(d):0;
if(d && ((s==GDK_SOURCE_PEN) || (s==GDK_SOURCE_TOUCHSCREEN))) {
    e.pen = true;
    e.pen_barrel = MouseState & GDK_BUTTON3_MASK;
    double *axes = NULL;
    switch(event->type){
        case GDK_BUTTON_PRESS:
            gdk_window_set_event_compression(((GdkEventButton *)event)->window, false);
        case GDK_2BUTTON_PRESS:
        case GDK_3BUTTON_PRESS:
            axes = ((GdkEventButton *)event)->axes;
            break;
        case GDK_BUTTON_RELEASE:
            gdk_window_set_event_compression(((GdkEventButton *)event)->window, true);
            axes = ((GdkEventButton *)event)->axes;
            break;
        case GDK_MOTION_NOTIFY:{
            GdkEventMotion *mevent = (GdkEventMotion *)event;
            e.mousepos = s_mousepos = Point(SCL(mevent->x_root), SCL(mevent->y_root));
            axes = ((GdkEventMotion *)event)->axes;
            break;
        }
    }
}
if(axes) {
    if(!gdk_device_get_axis(d, axes, GDK_AXIS_PRESSURE, &e.pen_pressure))
e.pen_pressure=NULL;
    if(!gdk_device_get_axis(d, axes, GDK_AXIS_ROTATION, &e.pen_rotation))
e.pen_rotation=NULL;
    if(!gdk_device_get_axis(d, axes, GDK_AXIS_XTILT, &e.pen_tilt.x)) e.pen_tilt.x=NULL;
    if(!gdk_device_get_axis(d, axes, GDK_AXIS_YTILT, &e.pen_tilt.y)) e.pen_tilt.y=NULL;
}
}
#endif
...

```

And finally also in GtkEvent.cpp:

```

...

bool Ctrl::ProcessEvent0(bool *quit, bool fetch)
{
    ASSERT(IsMainThread());
    bool r = false;
    if(IsWaitingEvent0(fetch)) {

```

```

while(Events.GetCount() > 1) { // GEvent compression (coalesce autorepeat, mouse
moves/wheel, configure)
    GEvent& a = Events[0];
    GEvent& b = Events[1];
    if(b.type == a.type && a.windowid == b.windowid && a.state == b.state) {
        if(a.type == GDK_KEY_PRESS && a.value == b.value)
            b.count += a.count;
        else
            if(a.type == GDK_SCROLL)
                b.value = (int)b.value + (int)a.value;
            else
                // if(findarg(a.type, GDK_MOTION_NOTIFY, GDK_CONFIGURE) < 0)
                if(findarg(a.type, GDK_CONFIGURE) < 0) // *** HERE: We cannot drop
                GDK_MOTION_NOTIFY events
                    break;
                Events.DropHead();

```

...

This last one dropped queued GDK_MOTION_NOTIFY events and ruined all attempts to get them through.

However, performance questions may need further considerations... It works here in virtual machine quite well on both Ubuntu 20.04 and Linux Mint 20.1, but hardware varies of course.

Best regards,

Tom