Wacom Support for Wayland

Stephen Chandler Paul
thatslyude@gmail.com
“Lyude” on FreeNode

March 21st, 2014

1 Synopsis

With Wayland being the intended successor to the X server, extending Wayland support to as many devices as possible is crucial. To a user who relies on their Wacom device, switching from X to Wayland would be seen as a step backwards, even with all of the issues Wayland provides a solution for. This is especially true for owners of pen enabled devices such as the Surface Pro and the ThinkPad Yoga, whose only reason for buying one of these devices might be their pen functionality. These sort of devices are only becoming more popular, and the market for them continues to increase. It is because of this that Wacom support in Wayland is essential.

2 Success Criteria

- Come up with a well-thought out protocol for clients to make use of Wacom tablets
- Enable the majority of Wacom tablets to work out of the box with Wayland
- Ensure basic tablet features work properly (erasers, pressure sensitivity, buttons, etc.)

3 Road-map

- **Design a protocol for Wayland to allow clients to handle input from Wacom devices**
  This is actually going to be one of the easier parts. Peter Hutterer has already proposed a possible protocol for doing this. It may be possible to use his proposal with the potential of a few small changes being needed.
- **Modify libinput to support Wacom devices**
  This is the part I think I'll find to be the most challenging. Here I'm going to have to figure out how to communicate with the device through the interfaces provided by the kernel and provide a decent API for accessing them through Wayland. This is where I'm also going to have to learn about all the tweaks and quirks that come with making tablets work properly. My plan for figuring this out is to use the xf86-input-wacom source code

as reference for all of these tweaks. Obviously we can't copy the design for the drivers from xf86-input-wacom since copying a design intended for the X11 server has the potential to lead us to making design decisions we're likely to regret later, but nevertheless the source code will prove valuable in figuring out the quirks tablets have, and how we should be dealing with them.

- **Implement support in Weston**
  This part's going to be slightly easier. We're not going to really have to do much design in terms of protocol here, we just need to figure out how to make Weston play nice with the API we've come up with.
- **Write client-side testing for, GTK+, Qt, etc.**
  This portion mostly serves as a kind of real-world testing. We'll get to see how what we've made will actually work with client-side toolkits. This will allow us to see what portions of the API were actually a good idea, and which one's weren't in a way that's only possible by testing out the implementation.

4 Estimated Timeline

- May 19th – May 28th: Begin decisions on the protocol that will be used. Look at Peter's proposal for a protocol, and see what changes might need to be made to it.
- May 29th – June 25th: Begin modifications to libinput to handle wacom devices
- June 25th – July 1st: Implement Wacom support in Weston
- July 2nd – July 8th: Downtime; will be taking the only trip I have planned during the summer
- July 9th – July 23rd: Continue implementing Wacom support in Weston
- July 24th – August 18th: Write client-side tests, fix any bugs and issues with the API

5 About me

I am currently 18 years of age, and I am enrolled as a full-time student with Springfield Technical Community College. Once I get my core classes out of the way, I am planning on transferring to a four year school, most likely Western New England University. I have been working with computers about as long as I can remember, and began programming them about four years ago after I began running Linux. I ended up starting with C as a programming language, and as a result the majority of my programming experience is in C.

6 Why I'm interested in working with X.org for the Google Summer of Code

I really believe that Linux based operating systems have a huge amount of potential in the desktop and laptop market, and I believe that it may very well be a key factor in deciding the direction that technology will take in the future. If an open source Linux distribution were to manage to take the stage years from now as the most popular operating system in the market, I believe that the influence it would have on how software is developed would be extraordinary. Already Linux has dominated the server market and the embedded market, and has inspired many others to become involved in open source programming. I do believe however, for such a huge milestone to ever be reached Linux distributions will need to reach a point where they work out of the box on almost any setup, and stability with any sort of necessary utility on the operating system is guaranteed to be stable. One of the things we need for this to happen is a well-designed window server, free of the design flaws the X11 protocol suffers. And of course, for this to work we're going to need good support for all commonly used input devices. Since input devices, although definitely not the simplest, are much easier to contribute support for since they require only minor processing on the host's end unlike things such as graphics drivers, which makes it the perfect area for me to start with.

7 Previous experiences with X.org related development

I worked alongside Peter Hutterer, Benjamin Tissoires, and Hans de Goede to add support for the TrackPoints found in Lenovo's latest line of ThinkPads. Although a lot of my code didn't end up being used due to some decisions to change how we were going to handle the secondary area of soft buttons used for the TrackPoint, some of my code did still went upstream and I provided valuable input for figuring out the final implementation of the new TrackPoint support.

8 Why choose me?

As you've probably noticed by now, open source is a huge interest of mine. I love being able to program and improve on things, especially when I know I can improve such a core piece of the average user's Linux system. In addition, I'm a quick learner who's highly motivated to work on the project, and is willing to take the time to dive into source code and teach myself whatever I might need to learn along the way.

9 Previous experience and accomplishments

- Head programmer of the Minnechaug Robotics Team 2011-2013
- Became AP Java certified in 2013
- Won the Minnechaug Regional High School Senior Technology Department Computer Science Award in 2013

You can also find various bits and pieces of the work I've done over time to help myself learn programming and improve on some of the things I use on my Github.