Akademy

4 - 11 September 2020

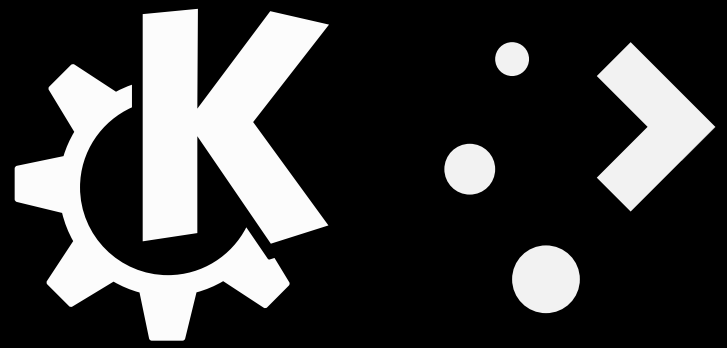
Getting into KWin and Wayland

How to get into developing our wayland experience and what to get from it

Aleix Pol Gonzalez

aleixpol@kde.org

Getting into KWin and Wayland



Aleix Pol i Gonzàlez <aleixpol@kde.org>

Who am I?

Aleix Pol i Gonzàlez <aleixpol@kde.org>
KDE e.V. President
KDE hacker
Work for Blue Systems
Barcelona

A bit of history

```
commit fd7f3549ac7ea1131a8567f4a10b44f0f933779e
Author: Simon Hausmann <hausmann@kde.org>
Date:   Fri Aug 20 14:44:38 1999 +0000

    - now it even compiles :-)
      (I like the BeOS-Style..)

    svn path=/trunk/kdebase/kwin/; revision=27900

commit 192ded1e6490176dee4bed2141c54d73b2990570
Author: Daniel M. Duley <daniel.duley@verizon.net>
Date:   Fri Aug 20 14:11:22 1999 +0000

    Removed Mattias's executable ;-)

    svn path=/trunk/kdebase/kwin/; revision=27896

commit 311db796c68e520e5b6d28829d6aaa4bfbcd1536
Author: Matthias Ettrich <ettrich@troll.no>
Date:   Thu Aug 19 23:26:42 1999 +0000

    Say hello to kwin. WARNING: NOT USABLE YET. See README.

    svn path=/trunk/kdebase/kwin/; revision=27871
(END)
```

```
commit bab5f16d3c3f25842884c4f2904930061e02d8c2
Author: Martin Gräßlin <mgraesslin@kde.org>
Date:   Wed May 15 13:47:27 2013 +0200

    Egl Backend using a Wayland surface for rendering

    This backend is able to composite on a Wayland surface instead of an X11
    overlay window. It can be considered as a prototype for a Wayland session
    compositor.

    For texture from X11 pixmap the backend uses XShm. This is far from
    optimal, but the KHR_image_pixmap extension is not available in Mesa's
    Wayland backend. It's a temporary solution till we have XWayland and
    texture from Wayland buffer.

    To use this backend one needs to specify the environment variable
    KWIN_OPENGL_INTERFACE with "egl_wayland". In future KWin should probably
    use this backend if the Wayland display env variable is defined.

    To use this setup:
    1. Have a normal X-Server running on e.g. VT7
    2. Start Weston on VT1
    3. Start a terminal on Weston
    4. start KWin with:

    DISPLAY=:0 KWIN_OPENGL_INTERFACE=egl_wayland kwin --replace &

    This should map a Wayland surface to Weston showing the content of the X
    setup. At the moment it's not yet possible to interact with the surface
    as input events are not yet recieved in the backend.
```

Why is Wayland so important?

Simplicity

Why is Wayland so important?

Security

Why is Wayland so important?

Control & integration

Why is Wayland so important?

Hardware support

What is Wayland?

What does a Wayland protocol look like?

```
<protocol name="potato">
  <interface name="potato" version="1">
      <request name="get_potato">
          <arg name="id" type="new_id" interface="wl_surface"/>
      </request>
      <event name="cooked" />
  </interface>
</protocol>
```

What does a Wayland protocol look like to us?

qtwaylandscanner

Relevant repositories:

kwin
kwayland
kwayland-server
kwayland-integration
qtwayland

KWin, relevant parts:

effects/
Output: plugins/platforms/
Input: input.cpp, libinput/

effects/

plugins/platforms/

drm/ < what everyone uses, most notably drmbackend.cpp and drmoutput.cpp

wayland/ < what you use when on windowed mode from wayland

Developing tips

```
dbus-run-session kwin_wayland --exit-
with-session konsole --socket akademy
```

Developing tips

Find the right place

Developing tips

Can we reproduce the feature/bug in windowed mode?

Developing tips

`WAYLAND_DEBUG=1`

Developing tips

GDB is a last resort

https://www.proli.net/2020/04/03/developing-kwin-wayland/

Join us!

plasma-devel@kde.org

kwin@kde.org

#kwin

#plasma

¿?

Aleix Pol i Gonzàlez <aleixpol@kde.org>