KWinFT in 2021

Latest development, Next Steps

₩.	Projects Groups Snippets Help	٩	@•~	Sign in / Register			
Win	f KWinFT						
û ₽	KWinFT 😁 Group ID: 6888960						
IJ	Official home of the KWinFT project Read more						
8	Search by name Vame V						
	Subgroups and projects Shared projects Archived projects						
	□ C ci-images ⊕ Image creation bench for projects' CI	★ 0		4 months ago			
	Disman 🔀 Qt/C++ display management library	★ 9		6 days ago			
	□ KDisplay ⊕ App and daemon for display managing	★ 4		3 months ago			
	□ K KWinFT ⊕ Wayland compositor and X11 window man	* 67		1 week ago			
	□ T tooling ⊕ Common tooling for KWinFT projects	★ 0		4 months ago			
	风 Wrapland ⊕ Qt/C++ library wrapping libwayland	* 19		3 months ago			

Roman Gilg

Prologue

- KWin since 1999
- KDE Plasma's default window manager
- KWin Wayland since 2011
- I worked on it since 2016



Atomic Mode Setting / Universal Plane preliminary support

https://phabricator.kde.org/D2370

Prologue



4 / 31

Initial Motivation for KWinFT

- Social and project organization:
 - KDE Community's "anarchic" theory versus practice
 - Transparent long-term planning for KWin impossible, short-term coding overvalued
 - Willingness to document code and define development standards
 - Mindset towards upstream freedesktop.org
- Disagreement on key technical directions
 - Qt reliance versus modern C++ and upstream
 - Focus on Wayland session, acceptance of temporary regressions

First Year 2020



Development Tooling

- Pre-merge CI pipelines: master branch always "clean"
- Nightly builds with additional static analysis
- clang-format and clang-tidy
- Test coverage reporting

Compliance	Build	Analysis	Deploy	Deploy	Downstream
⊘ Clang-Format Ø	⊘ Regular	Clang-Tidy	O Add Coverage	pages:deploy	
	Sanitized D	Regular Tests	🕢 Master Image 🥥		(Multi-project)
		Sanitized Te	⊘ pages ②		



First Year 2020



Universal Display Management



Drag screens	to re-arrange them	
Sharp Corporation (eDP-	 liyama North America PL2395W (DP-1) 	
(1920×1080)	(1920×1200)	
۵	Identify	
Device: Sharp	Corporation (eDP-1) $$	
🔳 Enal	bled	
Scale:	200% (2
Orientation: De Au		
<u>~</u>	39 29 29	
Resolution: 🦳 Au	to	
3840×	2160 (16:9) $$	
Refresh rate: 🔍 Au	ito	
≈ 60 H	z ~	
Replica of: None	~	
	• •	
Save displays' properties: For	any display arrangement	
O For	only this specific display arrangement	

Lightning Talk at XDC 20



First Year 2020



Second Year 2021

Major Goals in 2021 and Beyond

- Deeper integration with wlroots
- Library split out

Milestones for 2021

Milestones for 2022

The Library Split Out

- Distill reusable C++ libraries out of KWinFT
- To build X11 or Wayland compositors
- These libraries should only depend on the STL and:
 - XCB if you want to build an X11 compositor
 - wlroots and Wrapland for a Wayland compositor

Split Out or New

Good programmers know what to write. Great ones know what to rewrite (and reuse).

http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s02.html

KWinFT Ecosystem

Ecosystem After Library Split Out

The Library Split Out

Preparations ongoing already for months in the background

The Library Split Out: Preparations

- Unify X11/Wayland logic
- Code quality
- More STL, less Qt
- Namespacing

KWinFT's New Windowing Infrastructure

21/31

- Major project from October 2020 to February 2021
- Radically modernize and remodel the "heart" of KWinFT
- Improves windowing logic of KWinFT under Wayland and X

The Library Split Out: Preparations

- Unify X11/Wayland logic
- Code quality
- More STL, less Qt
- Namespacing
- Relocating code

	autotests	refactor: adapt to Wrapland's surface signals removal	5 days ago	
	🖿 cmake	build: define wiroots version variable	1 month ago	
	Colorcorrection	refactor: store input redirect in input platform	4 weeks ago	
	🖿 data	feat(input): add logging category	1 month ago	
	decorations	refactor: drop Toplevel's windowId	3 weeks ago	
	🖿 doc	feat(space): remove cascade placement	3 months ago	
	effects	refactor: adapt to Wrapland's surface signals removal	5 days ago	
	🖿 helpers	docs: correct use of KUIT markup in kill dialog text	3 months ago	
[🖨 input	refactor: adapt to Wrapland's surface property chang	5 days ago]
	🖿 kcmkwin	refactor(input): move X11 platform input to input dire	1 month ago	
	kconf_update	feat(space): remove cascade placement	3 months ago	
	Libkwineffects	feat(scene): add Gl timer query support information	5 days ago	
	🖿 perf	fix: correct ftrace function name	1 year ago	
[🖨 platform	refactor: split out X11 event filter from Workspace	1 month ago]
	Deplatformsupport	refactor: adapt to Wrapland's surface property chang	5 days ago	
	🖿 plugins	refactor: adapt to Wrapland's surface property chang	5 days ago	
	🖴 qml	refactor: port kwin from Plasma Components2	3 months ago	
[🖨 render	feat: add swap time debugging	5 days ago]
[🖿 rules	refactor: remove unused include	2 months ago]
	🖨 scripting	refactor: drop Toplevel's windowId	3 weeks ago	
	🖨 scripts	refactor: port kwin from Plasma Components2	3 months ago	
[🖿 seat	refactor: unvirtualize take session control function	1 month ago]
	🖿 tabbox	refactor: store input redirect in input platform	4 weeks ago	
	tests	build: remove libinput test source file	1 month ago	
	🖿 tooling	style: clang-format autotests	1 month ago	
[🖴 win	fix: serve frame callbacks also without damage	5 days ago]
	🖿 xwl	refactor: adapt to Wrapland restricting data access	1 week ago	

The Library Split Out: Reasons

- Share functionality with KWin
- Improve robustness
- Help the ecosystem to prosper
- Abstraction gap in current libraries

Bridging the Abstraction Gap

- Selective compilation with CMake
- Functional layering
- Aggressive use of C++ templates

Basic Features Goal

```
#include <kwinft-libs/platform/wayland/base>
#include <kwinft-libs/platform/wayland/server>
#include <kwinft-libs/win/wayland/window>
#include <kwinft-libs/win/workspace>
```

```
int main(int argc, char* argv[])
{
    auto platform_base = kwinft::platform::wayland::base();
    auto& wayland_server = platform_base.server();
    namespace ftwin = kwinft::win;
    auto workspace = ftwin::workspace<ftwin::wayland::window>(wayland_server.loop());
    return wayland_server.run();
}
```

More Features Goal

```
#include <kwinft-libs/platform/wayland/base>
#include <kwinft-libs/platform/wayland/server>
#include <kwinft-libs/platform/wayland/xwl>
#include <kwinft-libs/win/wayland/window>
#include <kwinft-libs/win/wayland/xwl/window>
#include <kwinft-libs/win/workspace>
```

```
int main(int argc, char* argv[])
{
    auto platform_base = kwinft::platform::wayland::base();
    auto& wayland_server = platform_base.server();
    auto xwl = kwinft::platform::wayland::xwl(wayland_server);
    namespace ftwin = kwinft::win;
    using workspace_t = ftwin::workspace<ftwin::wayland::window, ftwin::wayland::xwl::window>;
    auto workspace = workspace_t(wayland_server.loop());
```

return wayland_server.run();

}

Custom Features Goal

```
#include <kwinft-libs/platform/wayland/base>
#include <kwinft-libs/platform/wayland/server>
#include <kwinft-libs/platform/wayland/xwl>
#include <kwinft-libs/win/wayland/window>
#include <kwinft-libs/win/wayland/xwl/window>
#include <kwinft-libs/win/workspace>
```

```
#include <other-platform-lib/input>
#include <other-win-lib/window>
```

```
int main(int argc, char* argv[])
{
    auto platform_base = kwinft::platform::wayland::base<other_platform::input>();
    auto& wayland_server = platform_base.server();
    auto xwl = kwinft::platform::wayland::xwl(wayland_server);
```

```
namespace ftwin = kwinft::win;
using workspace_t = ftwin::workspace<otherwin::window, ftwin::wayland::xwl::window>;
auto workspace = workspace_t(wayland_server.loop());
```

```
return wayland_server.run();
```

Final Abstraction Span

Feature Richness

- Window management
- Virtual desktops
- Effects
- Scripts
- Window Rules
- etc.

Scaling the Project

- Two fulltime developers, multiple independent contributors
- Work on fundamentals fine, but specific Wayland features missing:
 - Input methods (work ongoing)
 - Graphical tablet support
 - Pipewire video stream
- Goal: team of 4-5 fulltime developers
- Long-term: federation with maintainers (libraries)
- Support us!

Project Organization

- Project home: https://gitlab.com/kwinft
- Open development via GitLab issues
- Developer community on Gitter and Matrix:
 - https://gitter.im/kwinft/community
 - #kwinft:matrix.org
- In the future:
 - Official homepage
 - Founding a non-profit limited