

Masonry: An imperative widget toolkit in Rust

Daniel McNab

 DJMcNab






About me

Open source maintainer in Linebender

Linebender is an open-source community working on 2d graphics and user interfaces in Rust, founded and led by [Raph Levien](#)

Connect with Linebender:

-  [Linebender](#)
-  [xi.zulipchat.com](#)
-  [linebender.org](#)

My work is currently funded by Google Fonts



Agenda

- Context
 - User's expectations of behaviour sets a baseline for app quality
 - Reaching this baseline is necessary but otherwise adds little value
 - Makes developing (non-web) frameworks expensive
- Masonry
 - Open source (Apache 2.0), application UI library
 - Foundation for higher-level UI frameworks
 - Allows work enabling this baseline to be shared between projects
 - Pre-Alpha, under active development
- Addendum: Reusable Components
 - 2d rendering and text layout libraries
- Q&A



Context

Cross-platform application UI



Baseline

There is still room for innovation in UI frameworks

Users of an app don't care what UI framework was used

They want to understand how to use the app

This is easier when the app meets their expectations

These expectations set a baseline for how the app should work

Meeting this baseline requires work by the app developer

However, for an app to meet these expectation, they must be supported
by the framework being used



Baseline: User expectations

- macOS Menu bar integration
- Input Method integration
- Focus handling
- Touch gestures
- Tabs
- Internationalisation
- Light/Dark mode
- Keybinding customisation
- Notifications
- Among many others



Baseline: Accessibility

Accessibility is also a core requirement

What good is an app which your users can't use?

- Platform Accessibility APIs
 - Screen readers, voice control, other accessibility tools
- Keyboard only navigation
- High contrast modes
- Different input paradigms (dictation/speech to text)

Potential legal requirements



Duplicated work

Getting all of these integrations right requires a lot of work

Existing libraries for some of the sub-problems:

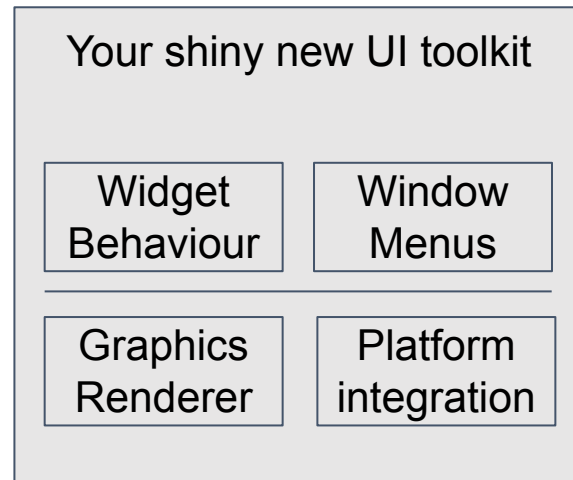
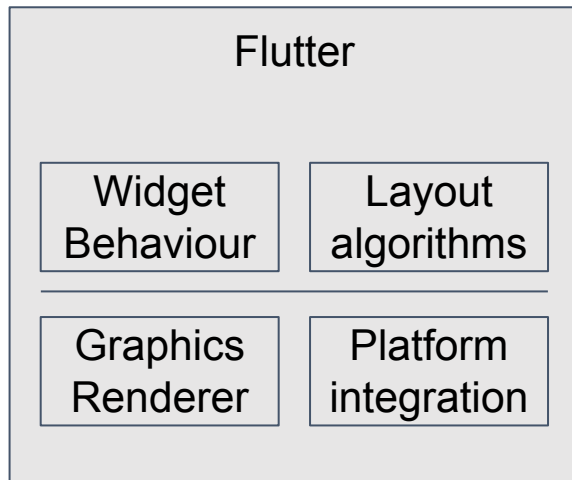
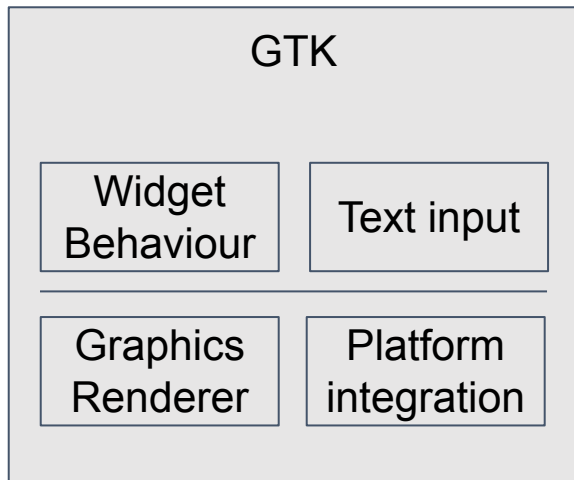
- [muda](#): macOS menu bar
- [AccessKit](#): Platform accessibility API integration (screen readers, etc.)
- [Fluent](#): Internationalisation/translations

But many must be implemented by the toolkit itself:

- Window menus (non-macOS)
- Keyboard shortcut customisation
- Text input



Duplicated work



Supporting the baseline adds no value



Existing Implementations

This work within existing UI toolkits has not been designed for re-use by other toolkits

Interleaving of toolkit wiring and widget behaviour

The most used existing implementation of this baseline is the web DOM

This has a number of shortcomings for cross-platform applications

- Designed for documents - lack of flexibility for advanced use cases
- Large dependencies or browser engine inconsistency



Masonry

crates.io/crates/masonry



987654

CE	C	DEL	÷
7	8	9	×
4	5	6	-
1	2	3	+
±	0	.	=

Here

UTC+01:00:00

22:30:37

Increase Decrease Minimum Maximum

11:30:37

Pitcairn Islands

13:30:37

Arizona

14:30:37

Saskatchewan

15:30:37

Peru

16:30:37

Barbados

17:30:37

Martinique

17:30:37

UTC-04:00:00

308 Permanent Redirect

Select

400 Bad Request

Select

401 Unauthorized

Select

402 Payment Required

Select

403 Forbidden

Select

404 Not Found

Select

405 Method Not Allowed

Select

406 Not Acceptable

Select

407 Proxy Authentication Required

Select

408 Request Timeout

Select

409 Conflict

Select

410 Gone

Select

HTTP Status Code: 102

Processing



102

Processing

Copyright © <https://http.cat>

Some of Masonry's current examples. Image copyright © http.cat



Masonry

Retained mode user interface library for cross-platform toolkits

Primary goal is to provide a core for higher-level toolkits

App focused

Cross-platform: macOS, Windows, Desktop linux, Android, iOS

Written in Rust



A new layer in the stack

Implements features common to all UI toolkits

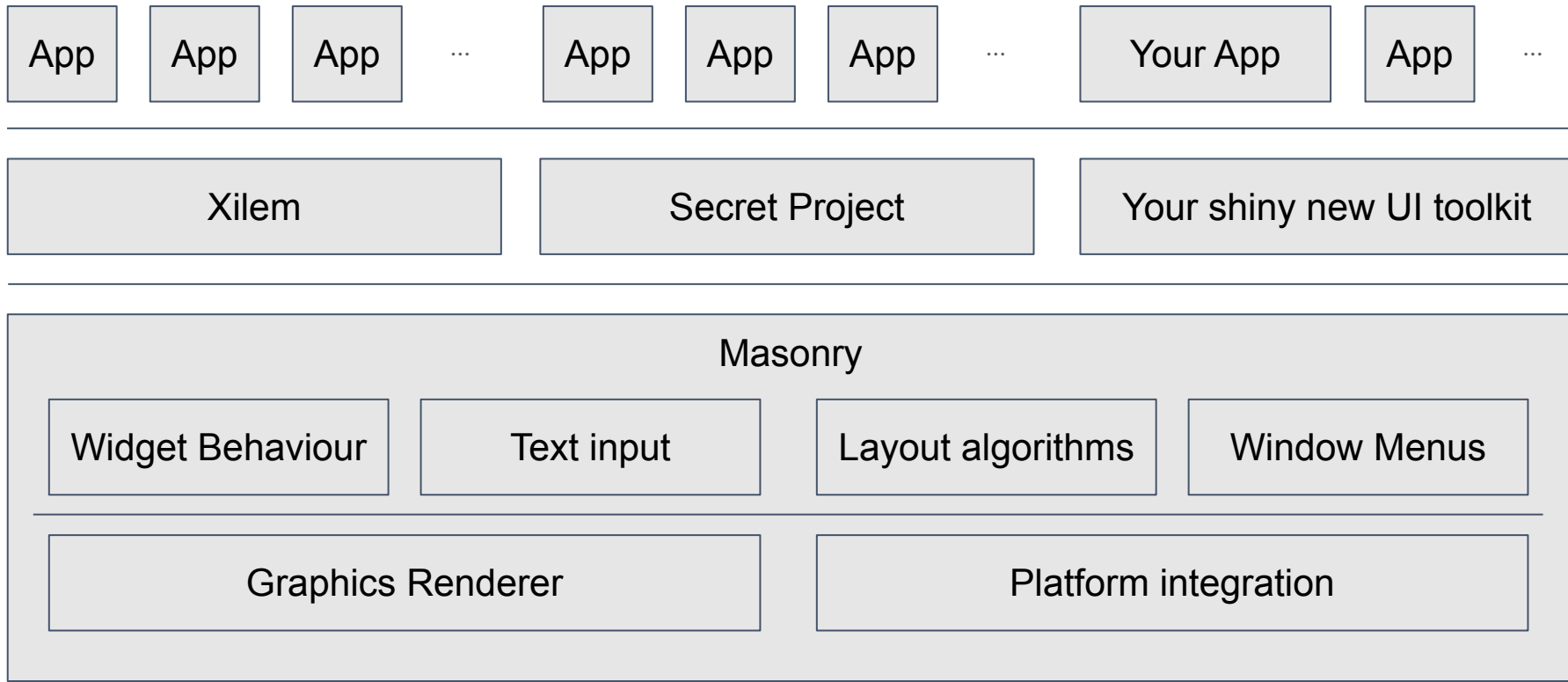
Similar level to the DOM, but specifically designed for application UI

Common vocabulary of widgets between higher-level libraries

Allows toolkits to innovate on developer experience



A new layer in the stack



Imperative



Robustness

- Testing
- Automatic tracing to file in development builds
- Pass system
 - Each individual step has well-defined boundaries
 - Fail early when assumptions broken
 - Continue with error in release mode
- Other debugging tools planned, pending implementation
 - Full inspector
 - Record/Replay



Performance

- Pass system runs updates only as-needed
- 2d renderer using GPU
 - Performance characteristics raise ceiling of possible apps
 - Vello, discussed later
- Planned:
 - Virtual scrolling
 - Compositor integration



Widgets

Application focused set of built-in widgets

- Well-behaved text input widget
- Scrolling, layout implementations
- Images, spinners
- Buttons, checkboxes, sliders

Custom widgets well-supported

- Fully custom drawing, layout, pointer/input handling and accessibility
- Animation support



Wider Rust Ecosystem

Dependencies:

- AccessKit
 - Platform accessibility APIs
- Winit
 - Creates windows and other platform integrations
- wgpu
 - Implements WebGPU spec cross-platform (incl passthrough on Web)
 - Used to implement Vello, enabling Masonry to be portable



Status

- Strong foundations in place; high quality rendering and text handling
- Used by Xilem
 - For more information, see Raph's 2024 talk: [Xilem: Let's build high performance Rust UI](#)
- Exploratory styling support
- Early stages

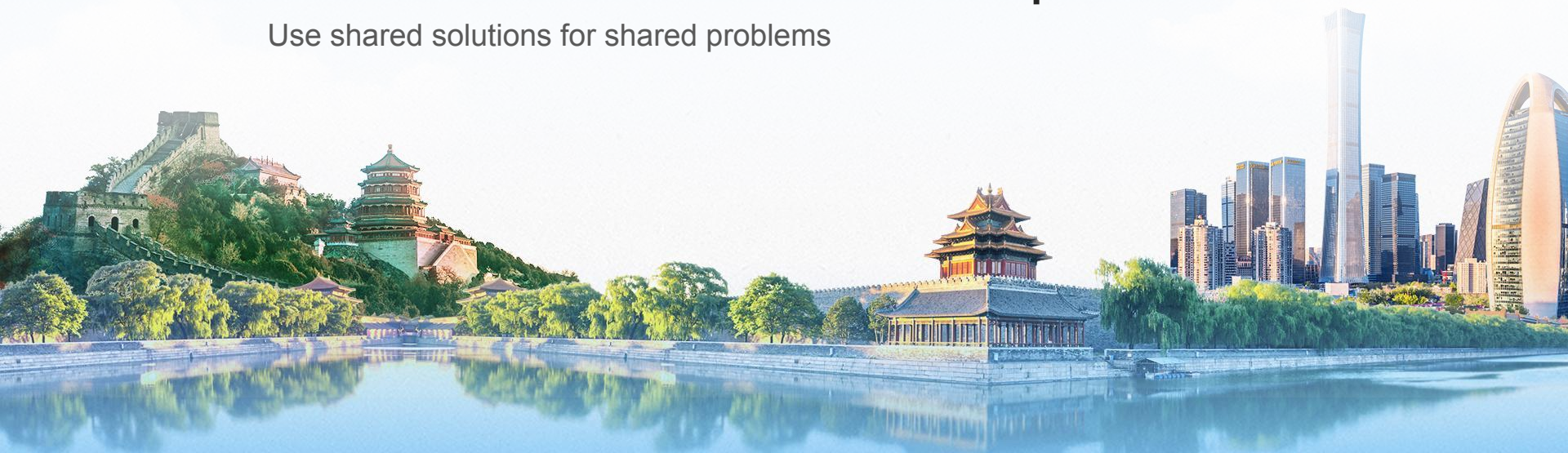
Help wanted!

- Additional high-level projects (Elm-like, Solid-like)
- Room for experimentation in styling
- Support for additional platforms
 - Web and OpenHarmony desired



Addendum: Reusable components

Use shared solutions for shared problems



2d rendering

crates.io/crates/vello

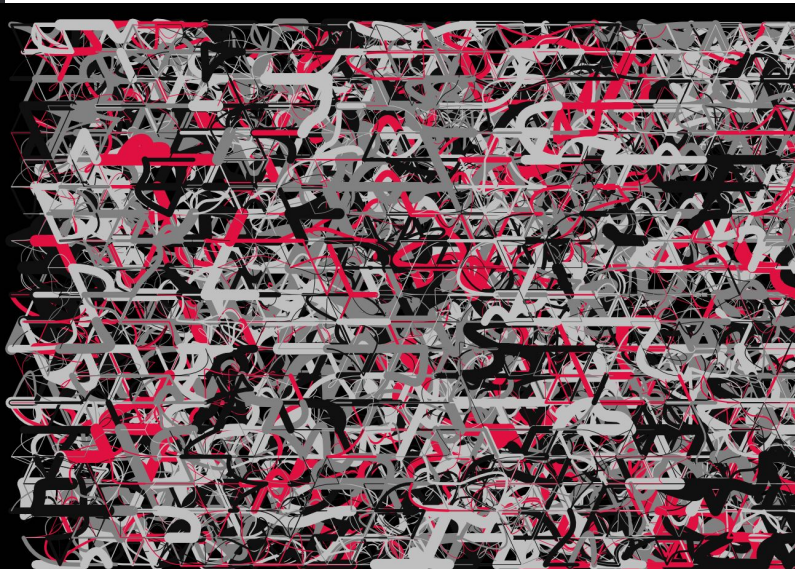
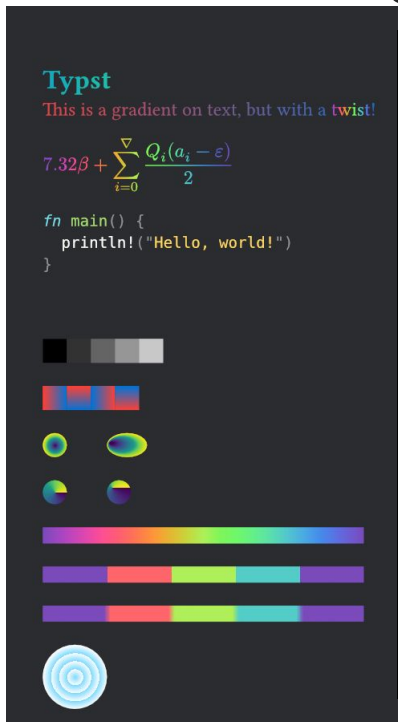


Vello is the 2d graphics renderer built for Masonry

- Vector graphics imaging model
- Developed with user interfaces in mind
- Performance characteristics raise ceiling of possible apps
- Version 0.3.0 recently released, first alpha
- See Raph's 2023 talk for more details
- [Vello: High performance 2d graphics](#)



2d rendering



mmark test: 20000 path elements (up/down to adjust)



Left to right: [Velyst](#), Vello's mmark example, [Velato](#) rendering a Lottie file



Text layout

crates.io/crates/parley



Parley is a robust text layout library in Rust

Based on [Fontations](#) work

Has support for editing as well as static text layout

Version 0.2.0 recently released



todos

Introduce Vello

Introduce Masonry

Introduce Blitz

1 item left

All

Active

Completed

Clear completed

Double-click to edit a todo

Created by jkelleyrtp

Part of TodoMVC


[Blitz](#) uses Vello for 2d rendering, and Parley for text layout






Contributions welcome

Weekly public meeting, see [#office hours](#) channel

Contact me directly:

-  [DJMcNab](#)
- Appointments welcome
- Daniel McNab on [Zulip](#)

Connect with Linebender:

-  [Linebender](#)
-  [xi.zulipchat.com](#)
-  [linebender.org](#)



Q&A

xi.zulipchat.com

[#masonry](#) > Gosim China 2024 Talk



Slides: talks.djmcnab.com

