



Introducing wasi-gfx

Build Beautiful Graphics and
Safely Run AI with any GPU

Sean Isom & Mendy Berger, Renderlet

How GPUs work with Wasm today

- They don't!
- In a browser: JS Bindings
 - Emscripten
 - wgpu
- In pure WASI: ?
 - Manually expose functions through host runtime
 - memcpy in/out of linear memory



Solution:
Map the entire WebGPU API into WASI!

Why do we need this?

- Wasm is useful for more than just browser apps!
- Use Cases:
 - UI applications
 - Headless graphics
 - Rendering plugins
 - AI (cases not covered by wasi-nn)
 - Scientific computing
 - And more! Bring your own ideas

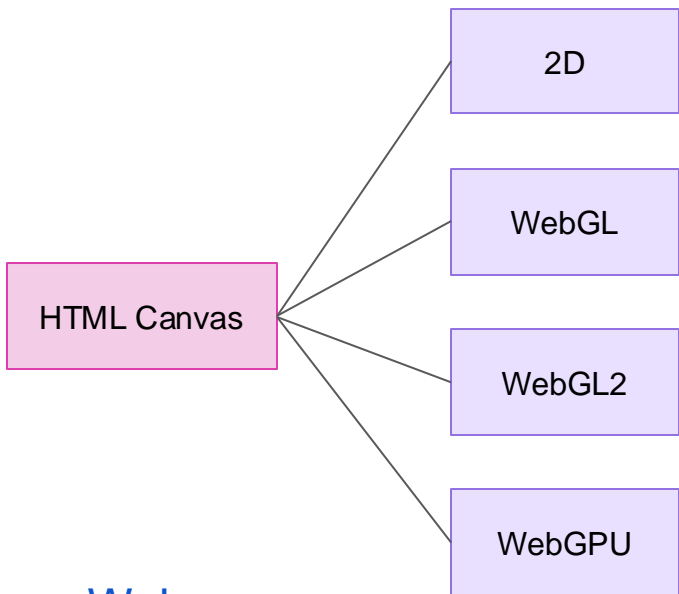
Package Overview

- wasi-gfx includes the following packages:
 - **wasi-webgpu**
 - gpu api generated from the WebGPU spec, with minor changes that don't change semantics
 - **wasi-surface**
 - a surface/window where you can draw to, and handle basic user input
 - **wasi-graphics-context**
 - point of connection between windowing system and graphics api
 - **wasi-frame-buffer**
 - for cpu based rendering

Connecting Packages

windowing-system

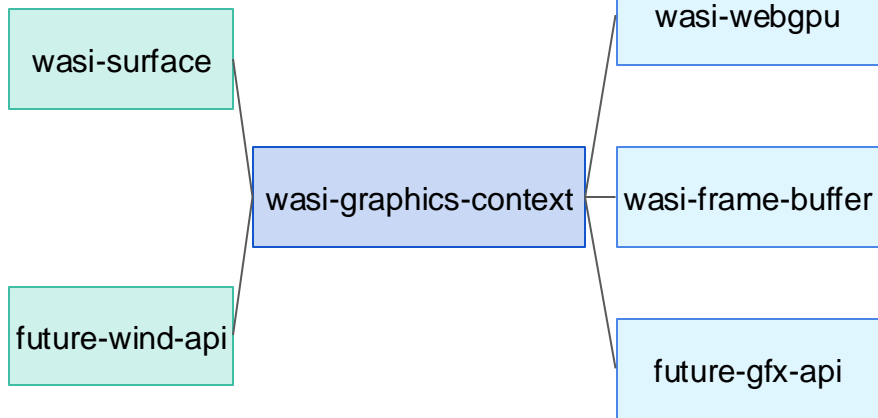
graphics-api



Web

windowing-system

graphics-api



WAS

Demo - wgpu example

wasi-gfx Architecture

Guest

Wit generation

WebGPU spec

webidl2wit

wit-transforms

Host

User Code
(Rust)

wgpu
wasi-backend

webgpu.wit

webgpu.wit

Runtime
(e.g. Graphtime)

User Code
(JavaScript)

User Code
(Go)

User Code
(C/C++)

webgpu.h

wasi-gfx-runtime

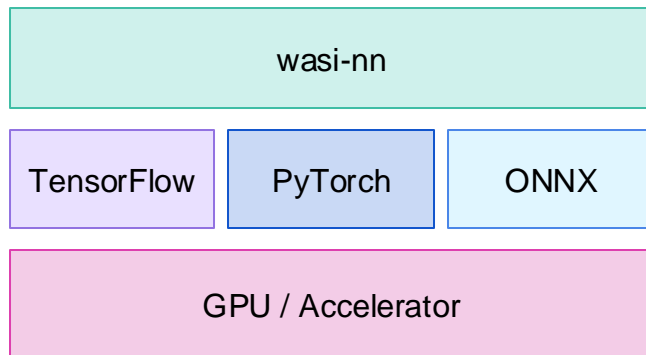
Wasmtime

WebGPU impl
(e.g. wgpu or dawn)

Demo - Bevy

Running AI in wasi-gfx

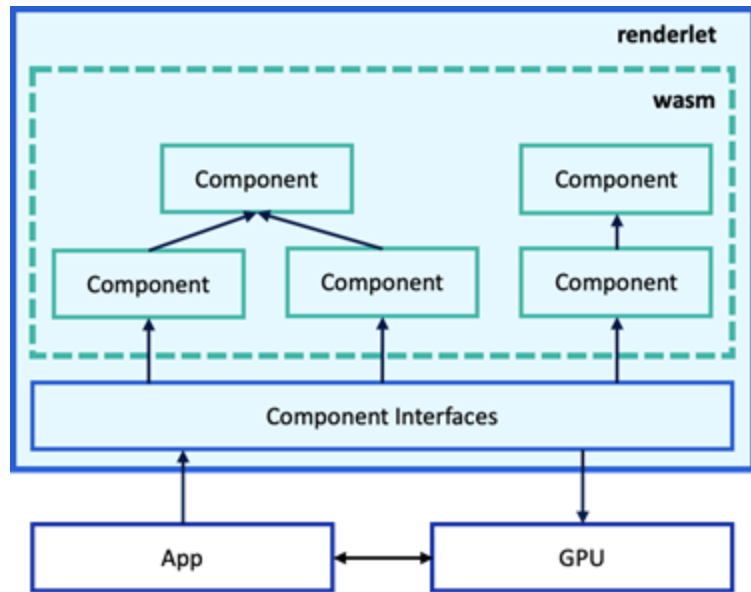
- Use wasi-nn:
 - ML inference on supported backends
 - Tensorflow, PyTorch, etc.
 - Fastest method
- Use wasi-gfx:
 - Unsupported backends
 - Training and tuning
 - Low level GPU access
 - Could build wasi-nn on wasi-gfx



Demo - ONNX

Plugin / component architectures

- 3rd-party code can need GPUs
- Define host / guest interface
- Sandboxed resource creation
- Plugin can:
 - Read: Scene/Document model
 - Write: Defined buffers
 - Write: Render Target



Demo - Renderlet

What's Next?

- Finalize wasi-gfx spec
- webgpu.h C bindings for wasi-webgpu
- Full wasi-gfx backend in wgpu
- Full wasi-gfx support in Bevy
- Complete runtime implementation of wasi-gfx
- Async for WebGPU promises
- Accessibility, VR, input and windowing support



wasi-gfx is now Phase 2!

Resources



[Demos & More](#)

Sean



x.com/theisomizer

Mendy



mastodon.online/@Mendy

Meetings every Tuesday @ 17:00 UTC

github.com/WebAssembly/wasi-gfx

github.com/wasi-gfx