Chrome will support the WebGPU API by default—here's why that's important



Andrew Cunningham/Google

reader comments

66 with

Share this story

Google announced today that it would enable WebGPU support in its Chrome browser by default starting in version 113, currently in beta. In development since 2017, WebGPU is a next-generation graphics API that aims to bring the benefits of low-overhead APIs like Microsoft's Direct3D 12, Apple's Metal, and Vulkan to web browsers and other apps.

WebGPU support has been available but off by default in Chrome for a while now, because the API wasn't finalized and things could break from update to update. Google says that Mozilla and Apple will eventually support WebGPU in Firefox and Safari, and browsers like Microsoft Edge and Opera that rely on the Chromium browser engine can presumably choose to switch it on just as Google has.

Chrome 113 supports WebGPU on Windows, macOS, and ChromeOS to start, with "support for other platforms" like Linux and Android "coming later this year." This browser version should roll out to all Chrome users sometime in May.

Support in Chrome is a big boost for any new standard, whether it's a new or updated API, image format, or video codec. Chrome accounts for roughly two-thirds of browser market share worldwide, according to StatCounter data, and nearly 80 percent of all browser market share if you count other Chromium-based browsers. Once Safari and Firefox add support, virtually all web browsers everywhere will be able to run WebGPU code, so it's worth explaining in brief what WebGPU is and why it exists.

What is WebGPU?

WebGPU is a successor of sorts to WebGL, a much older API that allows OpenGL-based graphics to be rendered in your browser without requiring additional third-party plugins like Adobe's Flash. Though revolutionary when it was announced in 2009, WebGL today suffers from many of the same problems OpenGL does: It doesn't take advantage of all the features of current GPUs, it can lose performance because of driver overhead, and it has only limited and kludgey support for GPU compute workloads.

"Whereas WebGL is mostly for drawing images but can be repurposed (with great effort) to do other kinds of computations, WebGPU has first-class support for performing

general computations on the GPU," says the draft document explaining why WebGPU exists. The new API will enable improved graphics in frameworks like Babylon.js and game engines like Unity; will support GPU-accelerated workloads like the ones used for machine learning and AI; and can make it easier for web-based photo and video apps (among others) to use the kind of GPU acceleration that native apps benefit from.

Advertisement

WebGPU is being developed by a working group that includes members from Google, Microsoft, Apple, Mozilla, Intel, and many other companies, and as a result, the API has been designed to be as widely compatible as possible. WebGPU isn't a direct implementation of any existing graphics API, but rather is its own API that sits on top of and interfaces with Direct3D 12, Metal, and Vulkan. This explainer from Surma, an engineer at Shopify and former Googler, sums it up well (emphasis ours):

> *"While WebGL is just a thin wrapper around OpenGL, WebGPU chose a different approach.* ***It introduces its own abstractions and doesn't directly mirror any of these native APIs****. This is partially because no single API is available on all systems, but also because many concepts (such as extremely low-level memory management) aren't idiomatic for a web-facing API. Instead, WebGPU was designed to both feel "webby" and* ***to comfortably sit on top of any of the native graphics APIs while abstracting their idiosyncrasies****. It's being standardized in the W3C with all major browser vendors having a seat at the table."*

This has the benefit of making WebGPU code very portable—write WebGPU code, and as long as users have hardware and a browser that supports it, you should get the same result on a Windows system running Direct3D and an Android phone running Vulkan. And it avoids some of the hacks that keep WebGL running—for instance, WebGL on Windows and macOS doesn't even use OpenGL

directly, but rather a translator called ANGLE that turns OpenGL API calls into Direct3D and Metal API calls instead. With WebGPU, there's no need to start a separate project to write around a slow or poorly maintained OpenGL implementation.

WebGPU also has its own shader language (WGSL), which the Chrome team is still working to support—today's announcement says that Google "plans to provide deeper access to shader cores" via WGSL in a future release. Like the rest of the WebGPU API, WGSL is intended to be platform-agnostic and easily translatable to the preferred shader language of whatever operating system you're running it on.

Finally, despite the "web" in its name, the WebGPU API isn't restricted to browsers. The wgpu project is a version of WebGPU for Rust that allows WebGPU apps to be written and run outside of web browsers.

If you want to experiment with WebGPU now, the beta version of Chrome 113 is available here. A GitHub repo of code samples is available here, and there's also official documentation about both WebGPU and WGSL.

https://arstechnica.com/?p=1929889