

WebGL + WebGPU Webinar

Summer 2024

July 23, 2024



WEBINARS
& MEETUPS



WebGL & WebGPU Updates

Ken Russell (Google) and Kelsey Gilbert (Mozilla)
On Behalf of the WebGL WG and WebGPU CG



WEBINARS
& MEETUPS



Agenda

WebGL Updates

- Extension Promotions
- ANGLE/Metal Progress
- Context Loss Robustness Improvements

WebGPU Updates

- Standardization
- Implementations
- Three.js' WebGPU Backend
- Partnerships, Resources and Contributions

Call to Action: Join WebGL & WebGPU Communities

khr.io/web202407

Extension Promotions

Several extensions promoted to community approved since the last Meetup:

- [OES sample variables](#)
- [OES shader multisample interpolation](#)
- [EXT render snorm](#)

This completes the set of smaller extensions Alexey Knyazev proposed.

Thanks especially to Alexey for driving these extensions!

Already shipping in Chrome and coming in Safari 18

ANGLE/Metal Progress

- Work is still ongoing in ANGLE's Metal backend
 - Used by WebKit's WebGL implementation on macOS/iOS, and soon, Chromium's on macOS
- Chrome is ramping up shipment on Intel CPU Macs
 - Already shipping on Apple Silicon
- Non-blocking program linking support is nearing completion
 - Already exposed as KHR_parallel_shader_compile
- Fixed flaky occlusion query results
- Thanks to Alexey Knyazev for many fixes, the full WebGL 2.0.1 conformance suite is passing on ANGLE's Metal backend!
 - (Not counting frontend or test bugs)
 - Paves the way for the next spec and test suite snapshot

Context Loss Robustness Improvements

- Kelsey Gilbert (Mozilla) is driving a [WebGL spec change](#) which will make applications more robust to WebGL context loss/restoration
- When the context is lost, the browser will now return non-null objects from operations like `createBuffer` or `createTexture`
- This will hopefully make it easier for applications to respond correctly to context loss events, and reduce the chances of null-pointer exceptions
- Thank you Kelsey for proposing and driving this change, and revising the many associated conformance tests!

WebGPU

A "modern" graphics API for the Web:

- A successor to WebGL, not a replacement.
- Compute shaders on the Web!
- Lower overhead API
- Foundation for future features (bindless, ray tracing, multithreading ...)

Development happens [on GitHub](#) and [at the W3C](#)

- Anybody can join and participate in the development!

WebGPU Standardization Updates

Current [API](#) and [WGSL](#) specifications are essentially ready for a v1.0 “Candidate Recommendation” snapshot! We are largely in the spec cleanup stage and finishing up to-dos.

Steady progress on core spec. Major progress on [Compatibility Mode](#) - see the following slides.

Continued progress on WebGPU Shading Language primitives for faster AI models on the web - 8-bit dot product just landed, thanks to Intel's Web Graphics Team in Shanghai!

WGSL pointer improvements through “unrestricted_pointer_parameters” and “pointer_composite_access” language features.

WebGPU Implementation Status

Safari

- Enabled in Safari Technology Preview - please test!

Firefox

- Enabled in Nightly on Windows and Linux, for testing and experimentation!
- Mac is in progress.
- Aiming to ship to Release by end of year!

Chromium

- Currently shipping on Windows, ChromeOS, Mac, and Android!
- Tracks the top-of-tree [WebGPU](#) and [WGSL](#) specifications
- web.dev/gpu for higher level details
- Looking forward to your feedback, and applications built using WebGPU!

Implementations are mostly interoperable already!

khr.io/web202407

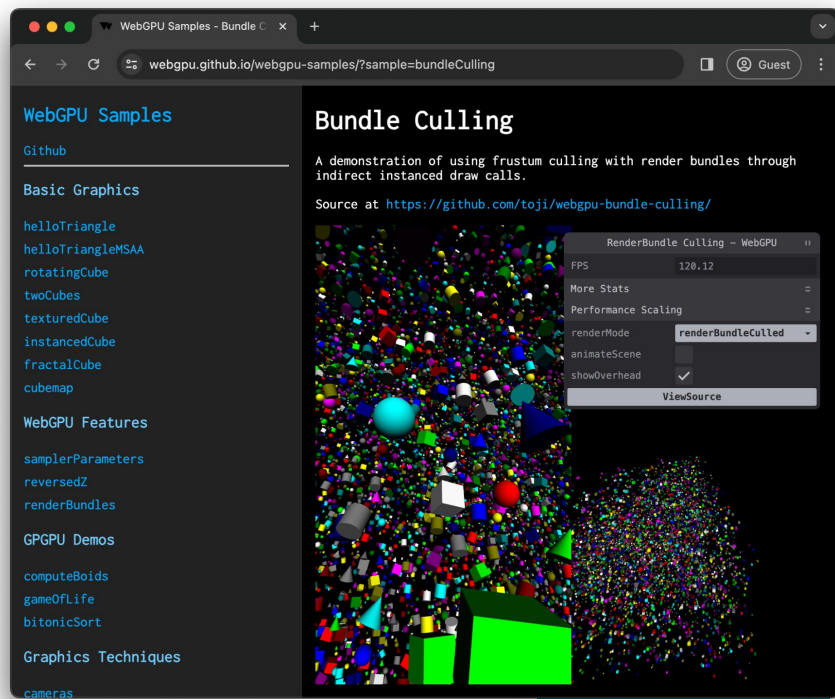
Three.js' WebGPU Backend

- Much recent progress on Three.js's WebGPU backend
- There are already many WebGPU examples in [Three's examples](#)
- A new [Three.js Shading Language](#) has been created to streamline writing the new node graph based material system
 - TSL produces both WebGPU's and WebGL's shading languages
- To ease developers' transition to the WebGPU backend, there are actually WebGL 2.0 fallbacks for most functionality!
- This is a major rearchitecture of Three.js; the team is aiming for a first official release of the WebGPURenderer later this year
- Thanks to [sunag](#) in particular for driving this work!
- [Follow Three.js](#) for the team's latest updates

WebGPU Samples

<https://webgpu.github.io/webgpu-samples>

- Refactored for easier participation
- Can add external examples too!
- Several new samples
 - [MSDF text](#)
 - [skinned mesh](#)
 - [render bundle culling](#)
 - [points](#)
 - [multiple canvases](#)
- [Submit yours!!!](#)



WebGPU Partnerships

Steady progress on WebGPU backends for popular web 3D libraries

[Three.js](#), [Babylon.js](#)

Ongoing partnerships with teams including Intel, [TensorFlow.js](#), [Google Meet](#), [MediaPipe](#), and more

[PlayCanvas](#) has been undertaking a major refactor of their engine in support of WebGPU

Tracking bug: <https://github.com/playcanvas/engine/issues/3986>

Fantastic feedback and collaboration with Unity, as they investigate porting existing shaders to WGSL and our new Uniformity Analysis requirements!

khr.io/web202407

WebGPU Resources

Tutorials:

- [WebGPU Fundamentals](#) by Gregg
- [WebGPU Best Practices](#) by Brandon

WebGPU Contributions!

Many ways to engage!

- Try the API and provide feedback (see later slides for channels)
- Try publishing sites using WebGPU
 - Can use WebGPU support in popular frameworks like Three.js, Babylon.js and TF.js
- Help with [conformance testing](#)
- Contribute samples / demos / articles using WebGPU

Join WebGL & WebGPU Communities

- The WebGL and WebGPU APIs are supported by vibrant online communities!
- If you're developing with these APIs, we would like to hear from you!
- On the WebGL side:
 - Please join the [WebGL Dev List](#): announcements of products, demos, new tools, job postings, questions, discussions - all are welcome!
 - Khronos' [public webgl](#) mailing list hosts lower-traffic spec announcements
 - The [WebGL Matrix chat room](#) offers a way to talk with browser implementers and other developers
 - You can find a lot of cool stuff by searching [#webgl on Twitter](#), [Mastodon](#) 🕶️

Join WebGL & WebGPU Communities

- On the WebGPU side:
 - Have API feedback ? See the [main WebGPU “gpuweb” repository](#) for options to communicate it to the community group
 - The [WebGPU Matrix chat room \(#WebGPU:matrix.org\)](#) also offers a great way to talk directly with browser implementers and other developers
 - There's an increasing amount of cool stuff showing up on [#webgpu on Twitter](#), [Mastodon](#) 🕶️
- We all look forward to hearing from you!

A recording of this presentation will be available at
<https://www.khronos.org/events/webgl-webgpu-meetup-july-2024>

For more information on WebGL, please visit
<https://www.khronos.org/webgl>

For more information on WebGPU, please visit
<https://github.com/gpuweb/gpuweb>



WEBINARS
& MEETUPS

