

WebGL + WebGPU Meetup Spring 2023 - Live at GDC!

March 22, 2023

KHRONOS
GROUP

WEBINARS
& MEETUPS



WebGL & WebGPU Updates

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



WEBINARS
& MEETUPS



Agenda

Join WebGL & WebGPU Communities

WebGL Updates

- General WebGL Updates
- New Extension Proposals
- Clip Distance Extension Draft
- Depth Bias Clamp Extension Draft
- Shader Pixel Local Storage Draft
- ANGLE/Metal Progress

WebGPU Updates

- Standardization
- Implementations
- Resources and Contributing

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 consider joining 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](#) 🕶️

Join WebGL & WebGPU Communities

- On the WebGPU side:
 - If you have feedback on the API, please 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](#) 😎
- We all look forward to hearing from you!

General WebGL Updates

- Browsers continue to maintain and enhance their WebGL implementations
- Smaller fixes to conformance tests and draft extensions
- Various fixes for partners including Google Meet
- GPU timer queries have been implemented in the ANGLE/Metal backend and are now available both in Chromium and WebKit

New Extension Proposals

- Alexey Knyazev (Independent/Khronos) is working on cost/benefit analysis of the remaining OpenGL ES extensions
- Working group is considering the following:
 - [ANGLE stencil texturing](#)
 - [EXT blend func extended](#)
 - [EXT clip control](#)
 - [EXT conservative depth](#)
 - [EXT depth clamp](#)
 - [EXT render snorm](#)
 - [EXT texture mirror clamp to edge](#)
 - [OES texture border clamp](#)
 - [OES texture stencil8](#)
 - [QCOM render shared exponent](#)
- Some have portability caveats and would be exposed in a restricted form

Clip Distance Extension Draft

- [WEBGL clip cull distance](#) provides support for hardware clip and optional cull distances
 - New ESSL 3.00 vertex shader outputs and fragment shader inputs
 - `gl_ClipDistance[]`
 - `gl_CullDistance[]`
 - New toggleable state `CLIP_DISTANCEi_WEBGL`
- Enables per-primitive clipping and culling, avoids expensive fragment discard
- Extension is present in draft form (behind a flag) in WebKit and Chromium
 - WebKit Nightly
 - Chromium Beta
- Please try in your apps and provide feedback to browser implementers

Depth Bias Clamp Extension Draft

- [EXT polygon offset clamp](#) adds a new parameter to the polygon offset function that clamps the calculated offset to a minimum or maximum value
- Prevents shadow artifacts by limiting depth bias values, which may get too high for steep polygons
- Trivial adoption
 - `polygonOffset(factor, units)` ⇨ `polygonOffsetClampEXT(factor, units, clamp)`
- Extension is present in draft form (behind a flag) in WebKit Nightly
- Chromium implementation is coming soon
- Please try and provide your feedback

Shader Pixel Local Storage Extension

- Chris Dalton (Rive) is developing a [WEBGL shader pixel local storage extension](#)
- Abstracts over many underlying implementation primitives and provides programmable blending functionality to applications
 - Subsumes `blend_equation_advanced_coherent`, and is more general
- Demo is [checked in](#) to [Khronos' WebGL repository](#)
- Draft implementation nearly complete in Chromium
 - Follow crbug.com/1421437 for updates
- Chris will talk more about this and demonstrate it later in the presentation!

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
- Team proceeding toward shipment in Chromium
- Major recent fixes:
 - [Shaders failing to compile exceeding available registers](#)
 - Thanks to Kyle Piddington from Apple for this fix
 - [Progress on dual-GPU support](#) on Intel/AMD MacBook Pros
- A few remaining blockers have arisen
 - Startup time regression - workaround in progress
 - User attrition using Metal backend - work in progress
 - Relanding an [important data upload optimization](#) from Gregg Tavares with new memory allocation heuristics
- Will turn on by default as soon as possible

WebGPU

An upcoming "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.
- Thanks to Khronos for hosting us here!

WebGPU standardization updates

Standardization of v1.0 is nearing completion. At the most recent WebGPU F2F meeting in February, we reached consensus on all major outstanding API and WGSL spec issues!

Now we finish our implementations, and iron out smaller issues considered for v1.0.

[WGSL standardization](#) proceeding excellently. Longstanding issues such as [trapping as an option for out-of-bounds access](#) being resolved.

[API standardization](#) is also proceeding superbly. Longstanding issues such as aliasing of storage buffers have been resolved, and implemented in at least one browser.

WebGPU - Implementation status

Firefox

- In Nightly, set `dom.webgpu.enabled` to `true` in `about:config`
- Not yet suitable for general secure browsing with this flag enabled, but expect it to be on by default in Nightly very soon for Windows! Other OSes to follow.
- Shipping to Release closer to the middle of the year!

Chromium

- 🎉🎉🎉 Releasing first version in [Chromium 113 this May!](#) 🎉🎉🎉
- Windows, ChromeOS and Mac (Linux and Android later)
- Tracks the top-of-tree [WebGPU](#) and [WGSL](#) specifications
- Will evolve as the specifications do!
- web.dev/gpu for higher level details
- Looking forward to your feedback, and applications built using WebGPU!

Implementations are mostly interoperable already!

khr.io/web202303

WebGPU - Using it in JS without a browser!

Many reasons to use WebGPU outside of a browser:

- Automated testing.
- Offline rendering using the same tech stack.
- "Native" frameworks like [Electron](#), [BabylonNative](#), etc.

Deno:

- Deno is a Javascript runtime with [built-in WebGPU support](#)
- Uses [wgpu](#) under the hood.
- Though heads-up: Deno is disabling webgpu for now until we resolve a startup time overhead issue.

Node.js

- [Dawn](#) has a [dawn.node](#) Node.js module.
- In a WIP but fairly good state (99% on par with Chromium for tests)

WebGPU - Partnerships and Collaboration

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 from and collaboration with Unity - exciting news later in the presentation!

khr.io/web202303

WebGPU - Resources

Tutorials:

- [Get started with GPU Compute on the web](#) by Francois
- [WebGPU - All of the cores, none of the canvas](#) by Surma
- [Raw WebGPU](#) by Alain
- [WebGPU Best Practices](#) by Brandon

Samples

- Austin's WebGPU Samples have been brought into the [github.com/webgpu](https://github.com/webgpu/webgpu) org, and are now at <https://github.com/webgpu/webgpu-samples/>!
- Check them out live at <https://webgpu.github.io/webgpu-samples/>!

WebGPU - Contributing!

Many ways to engage!

- Try the API and provide feedback via one of the channels mentioned
- Try out publishing sites using WebGPU using Chrome's WebGPU Origin Trial
 - Could use WebGPU support in popular frameworks like Three.js Babylon.js and TF.js
- Help with [conformance testing](#)
- Contribute sample / demos / articles using WebGPU
- Join the conversations on the [Matrix chat!](#)

A recording of this presentation will be available at
<https://www.khronos.org/events/webglwebgpu-meetup-at-the-2023-game-developers-conference>

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

Email: public_webgl@khronos.org



WEBINARS
& MEETUPS

