WebGL + WebGPU Meetup Summer 2023

July 11, 2023

KHRONOS GROUP / WEBINARS & MEETUPS

WebGL™ WebGPU

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
- Small Extensions
- 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
- Several smaller WebGL extensions being proposed for inclusion
- Pixel local storage extension progress
- Performance, correctness and security fixes for partners including Adobe and Google Earth

Small Extensions—Overview

- WebGL 1.0/2.0, being based on OpenGL ES 2.0/3.0, inherited lots of limitations of the latter that no longer exist in the underlying implementations (or never existed at all)
- There are several graphics pipeline features missing from WebGL that are still in ES 3.x scope and could be exposed in browser frontends, greatly improving developers' quality of life and capabilities
- Implementations in all applicable ANGLE backends (done)
- Passthrough WebGL frontend implementations (in-progress)

Small Extensions—Features & Origins

- **Vertex Shading**
  - Custom clip & cull distances (2006, D3D10)
- **Fixed-Function Vertex Post-Processing**
  - Clip space control (2013, ARB)
  - Depth clip control (2006, D3D10)
- **Fixed-function Rasterization**
  - Wireframe polygon rendering (2001, ARB)
  - Depth bias clamping (2006, D3D10)
- **Fragment Shading**
  - Conservative depth (2009, D3D11)
  - Noperspective interpolation (2006, D3D10)
  - Sample Built-in Variables (2009, D3D11)
  - Multisample interpolation (2009, D3D11)

- **Per-Fragment Operations**
  - Dual-Source Blending (2006, D3D10)
- **Renderable Formats**
  - Signed Normalized Buffers (2006, D3D10)
  - Shared Exponent Buffers (2014, Apple)
- **Stencil Texturing**
  - Sampling from combined DS (2006, D3D10)
- **Extra Texture Wrapping Mode**
  - Mirrored Clamp to Edge (2000, D3D8)

khr.io/web202307

Small Extensions—Specifications

- Vertex Shading
  - WEBGL_clip_cull_distance
- Fixed-Function Vertex Post-Processing
  - EXT_clip_control
  - EXT_depth_clamp
- Fixed-function Rasterization
  - WEBGL_polygon_mode
  - EXT_polygon_offset_clamp
- Fragment Shading
  - EXT_conservative_depth
  - NV_shader_noperspective_interpolation
  - OES_sample_variables
  - OES_shader_multisample_interpolation
- Per-Fragment Operations
  - EXT_blend_func_extended
- Renderable Formats
  - EXT_render_snorm
  - WEBGL_render_shared_exponent
- Stencil Texturing
  - WEBGL_stencil_texturing
- Extra Texture Wrapping Mode
  - EXT_texture_mirror_clamp_to_edge

khr.io/web202307

Shader Pixel Local Storage Extension

- Chris Dalton (Rive) is developing a WEBGL_shader_pixel_local_storage extension
- Provides programmable blending functionality to applications
  - Subsumes blend_equation_advanced_coherent, and is more general
- Draft implementation is available in Chrome Canary!
  - Enable **WebGL draft extensions** in **`about:flags`**
  - (Don't browse the open web with this turned on!)
- Demo is checked in to Khronos' WebGL repository
  - Shows how to implement all advanced blend modes
- Please try and provide your feedback!
  - Spec: WebGL repository
  - Implementation: crbug.com (Blink>WebGL)

ANGLE/Metal Progress

- Actively continuing work on ANGLE's Metal backend
- Used by WebKit's WebGL implementation on macOS/iOS
- Major recent changes:
  - Gregg Tavares' important data upload optimization relanded with new memory allocation heuristics
  - Dual-GPU support on Intel/AMD MacBook Pros is complete thanks to Jonah Ryan-Davis
  - Geoff Lang landed several large memory usage improvements
  - Scott Violet and Geoff improved startup time by precompiling built-in shaders
  - User attrition with Metal backend was diagnosed and solved
- Now on by default in Chromium; shipping soon in Chrome

WebGPU

An imminent "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

Current spec is essentially considered v1.0! Minor updates to API and WGSL specifications are continuing. Meeting cadence has slowed now that we are focusing on implementation.

Implementations are at varying stages of completeness (see next slide).

API standardization: Community group is currently discussing optional features and backward-compatible enhancements to the core specification, and generally polishing some rough edges in ways that shouldn’t break anyone.

WGSL standardization: Addressing editorial issues found by end users. Iterating on plans for how to more ergonomically enable extensions.

WebGPU - Implementation status

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!
- Notable todos: Indirect draws, bind group compatibility/deduplication, const/override expressions, abstract wgsl types.

Chromium

- 🎉🎉🎉 First version shipped in Chromium 113!! 🎉🎉🎉
- Windows, ChromeOS and Mac (Linux and Android coming)
- 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!

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.

Node.js

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

Deno:

- Deno is a Javascript runtime with built-in WebGPU support, though this was disabled until we can resolve a startup time overhead issue.
- Uses wgpu under the hood.

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

Unity has very early, experimental WebGPU support in 2023.2 alpha

See this forum thread for instructions on testing it

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 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 or Firefox Nightly!
  - 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
www.khronos.org/events/webglwebgpu-meetup-july-2023

For more information on WebGL, please visit
www.khronos.org/webgl

Email: public_webgl@khronos.org