WebGL + WebGPU Meetup

Fall 2022

October 4, 2022

KHRONOS GROUP / WEBINARS & MEETUPS

Ken Russell
Google,
WebGL WG Chair

Kelsey Gilbert
Mozilla

Ivan Popelyshev
Madcraft.io

Alexander Rose
Independent

Emmett Lalish
Google

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

Agenda

Join WebGL & WebGPU Communities

WebGL Updates

- General WebGL Updates
- Shader Pixel Local Storage extension
- Provoking Vertex extension
- ANGLE's Metal Backend - Stability & Other Work

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 repository for options to communicate it to the community group
  - The WebGPU Matrix chat room 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

Many fixes and enhancements to the WebGL conformance suite in the areas of:

- Transform feedback
- FBOs and depth/stencil
- clearBuffer[u][fi][v]
- Sampler objects
- Extensions and getParameter
- Draw functions and vertex attributes
- Floating-point framebuffer blending
- EXT_provoking vertex
- Splitting up long-running tests

Thanks in particular to Alexey Knyazev (Independent) and Gregg Tavares (Google) for the majority of these improvements

General WebGL Updates

- Chromium's WebGL implementation now restores WebGL contexts if they were lost and the application handles the `webglcontextlost` event
- Heuristics aren't tightly specified, but will restore one context every couple of minutes
  - More context losses than this, and the application will still be blocked from accessing WebGL, as before
- Improvement was made principally for Google Meet, and Visual Studio Code-based IDEs
  - VS Code's Terminal has a WebGL rendering backend which has been upgraded to handle context loss and restoration

Shader Pixel Local Storage extension

- Chris Dalton (Rive) is developing an ANGLE_shader_pixel_local_storage extension with the aim to expose it to WebGL
- This extension abstracts over many underlying implementation primitives and provides custom blending functionality to applications
  - Will be much faster than the currently available alternative of ping-ponging between two textures
  - Eliminates the need for KHR_blend_equation_advanced
- Will be available for prototyping soon in browsers
- Follow ANGLE bug 7279 if you're interested in progress on this extension

Provoking Vertex Extension

- EXT_provoking_vertex provides control over which vertex initiates a primitive
  - OpenGL convention = last vertex
  - Most other APIs = first vertex
  - Governs the behavior of flat shading
- Emulation is expensive on multiple WebGL implementations, making flat shading impractical to use
- WebKit's implementation contributed by Alexey Knyazev is unblocked and will show up in Safari Technology Preview soon
- Aiming to implement in Chromium soon as well
- Availability of this extension implies that it should be used for best performance

ANGLE's Metal Backend

- Work is still ongoing in ANGLE's Metal backend
- Used by WebKit's WebGL implementation on macOS/iOS, and soon, Chromium's on macOS

ANGLE/Metal Stability

- Chrome has been experimenting with ANGLE's Metal backend in the Canary channel on macOS
- Most significant shipment blocker has been an increased crash rate in Chrome's GPU process
  - Inside implementation of glTexImage2D and glReadPixels on AMD GPUs
- Suspected bugs similar to ones seen earlier in OpenGL drivers
- Postulated workarounds did not have an effect

ANGLE/Metal Stability

- Geoff Lang (Tech Lead/Manager of ANGLE project) studied these crashes and guessed what operations in the browser might cause them
  - Specifically - snapshots of tabs for the hover pop-ups
- Stress-tested switching among lots of tabs and was able to reproduce (!)
- Found that the crashes occurred when uploading to or reading back from IOSurface-backed textures
- Created two targeted workarounds for the upload and readback paths which have eliminated these crashes (!)
- This work has substantially unblocked shipment of ANGLE's Metal backend in Chrome

Other ANGLE/Metal Work

- Jonah Ryan-Davis (Google) and Zhenyao Mo (Google) are finishing support for dual-GPU MacBook Pros with ANGLE's Metal backend in Chrome
  - This is the last release blocker
- Gregg Tavares (Google) is fixing bottlenecks in glBufferSubData
- Kyle Piddington (Apple) is optimizing uniform buffer handling which is preventing some content (Unity's in particular) from running
- Dan Glastonbury (Apple) is implementing extensions for synchronizing with external Metal event objects
- We're grateful for this ongoing fruitful collaboration

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. Blockers are being addressed, and we're polishing the spec and reaching a decent amount of conformance testing. V1.0 specs in 2022Q4 hopefully!

WGSL standardization tackling hopefully its final few major issues:

- Allowing for implementations to trap/discard instead of e.g. clamp out-of-bounds access
- New static analysis pass to prevent data/pointer aliasing issues
- Discussion of how to satisfy function-out-param functionality (restricted pointers? `inout` keyword? tuple-destructuring assignment?)
- Lots and lots of clarifications (e.g. portability of out-of-domain inputs to builtins, such as sqrt(-1))

API standardization is also trying to finish up:

- Finalizing buffer mapping semantics
- Async shader module and pipeline creation
- GPUExternalTexture for automagically sampling rgb from e.g. <video> sources
- Lots of polish as well :)

WebGPU - Implementation status

Firefox

- In Nightly set **`dom.webgpu.enabled`** to **`true`** in **`about:config`**
- Not yet suitable for browsing securely with this flag enabled, but expect it to be on by default in Nightly later this year!

Chromium

- Windows, ChromeOS and Mac (Linux and Android later)
- The WebGPU Origin Trial allows publishing WebGPU apps on the Web today!
  - Breaking API / shading languages are happening by design. You must fix warnings surfaced in the devtools!
  - web.dev/gpu
- Aiming for release close to the v1.0 release of the standard this year.

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.

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 - Chromium 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

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

- Check out the up-to-date WebGPU Samples repo (Github) by Austin

WebGPU - Contributing!

Many ways to engage!

- Try the API and provide feedback on any channel
- 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/webgl-webgpu-meetup-october-4-2022

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

Email: public_webgl@khronos.org