WebGL and WebGPU Updates

On Behalf of the WebGL WG and WebGPU CG
WebGL+WebGPU Meetup, January 2022
Agenda

Join WebGL & WebGPU Communities
KTX 2.0 Updates
Firefox WebGL Updates
WebGL 2.0 in Safari / Chrome Updates
New & Upcoming WebGL Extensions
WebGPU Updates
WebGPU Origin Trial
WebGPU Samples
Presentations
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](https://lists.khronos.org/mailman/listinfo/webgl-dev): announcements of products, demos, new tools, job postings, questions, discussions - all are welcome!
  - Khronos' [public_webgl](https://lists.khronos.org/mailman/listinfo/public_webgl) mailing list hosts lower-traffic spec announcements
  - The [WebGL Matrix chatroom](https://matrix.org/#/join/https://matrix.org/leftroom/810335) offers a way to talk with browser implementers and other developers
  - You can find a lot of cool stuff by searching [#webgl on Twitter](https://twitter.com/search?q=%23webgl) 😎
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 chatroom also offers a way to talk 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!
KTX 2.0 Updates

- **KTX 2.0** — standardized image container format for GPU textures
  - Supports all pixel formats and texture types: arrays, cubemaps, etc
  - Ratified in April 2021
- Built-in support for **Basis Universal** codecs
  - ETC1S and UASTC, including optional Zstandard pass
- **KHR_texture_basisu** adds portable GPU-compressed textures to glTF 2.0
  - Implemented in Three.js, Babylon.js, <model-viewer>, more engines to come soon
- **KTX-Parse** — TypeScript parser library
  - Can be used with Binomial’s container-independent transcoders or with Khronos’ transcoders optimized for the Web.
- **KTX-Software** — Reference C/C++ implementation
  - Precompiled cmd-line tools, including WebAssembly (emscripten) builds
- Best practices and usage guidelines
  - For artists
  - For developers
KTX 2.0 Supercompressed Textures

Basis Universal Textures in KTX 2.0 Container - April 2021

- Compact, high-quality textures with efficient, on-the-fly conversion to diverse GPU native compressed textures to reduce download AND GPU memory size
- glTF assets can now use JPG, PNG OR KTX 2.0 Textures
- UASTC mode for higher quality (e.g. normal maps), ETC1S mode for smallest file and memory sizes

https://www.khronos.org/ktx/

Models downloadable [here](https://www.khronos.org/ktx/)

This work is licensed under a Creative Commons Attribution 4.0 International License
Firefox WebGL Updates

- Miscellaneous performance improvements
- Many specifically targeted at Google Meet's background blur/replace effects
- Working on implementations of new WebGL extensions
WebGL 2.0 In Safari / Chrome Updates

- WebGL 2.0 shipped in Safari 15 on both macOS and iOS last year! 🎉 🎉 🎉
- Running on top of Metal on all recent Mac and iPhone hardware and OSs
- Apple and Google actively collaborating on the common substrate
  - ANGLE's Metal backend, originally contributed by Quyên Lê
- Highlights of recent work
WebGL 2.0 In Safari / Chrome Updates

- Safari team is rapidly **diagnosing and fixing** any regressions that have been reported compared to the OpenGL backend
  - Both correctness and performance
- Fixing **WebGL 2.0 conformance failures**
  - Highlight: major transform feedback bug fix by Gregg Tavares from Google and Kyle Piddington from Apple
- **Supporting dual-GPU MacBook Pros**
- Fixing end user test cases like **Unity's HTML5 export path**
- Working to switch Chrome to use ANGLE's Metal backend as well
WebGL 2.0 In Safari / Chrome Updates

- WebGL 2.0 can now be considered universally available across browsers, operating systems and devices
- As an application author, you can target WebGL 2.0 with confidence
- WebGL 2.0 has resolved many corner cases and behavioral differences compared to the combination of WebGL 1.0 + its many extensions
- We encourage you to migrate to WebGL 2.0
- It's no longer necessary to maintain a WebGL 1.0 fallback path unless you need to reach absolutely every device
  - In particular, older Windows machines and Android devices
2  consider bolding this. shout it from the rooftops! That's the main message!
   David Neto, 1/21/2022

1  This leaves out Firefox.
Consider changing to
"WebGL 2.0 is everywhere" or similar.
   David Neto, 1/21/2022
WebGL 2.0 In Safari / Chrome Updates

- You can test Chrome on top of ANGLE's Metal backend today by launching it from the command line:
  - `/Applications/Google\ Chrome\ Canary.app/Contents/MacOS/Google\ Chrome\ Canary --use-cmd-decoder=passthrough --use-angle=metal`
- and compare its behavior to the OpenGL backend:
  - `/Applications/Google\ Chrome\ Canary.app/Contents/MacOS/Google\ Chrome\ Canary --use-cmd-decoder=passthrough --use-angle=gl`
- Some regressions exist, like low-power/high-performance GPU selection not working yet
- Please file any bugs you find with the Metal backend on ANGLE's issue tracker
- Please file any bugs you see in WebGL in Safari 15 on bugs.webkit.org, component "WebGL"
- (For other browsers' bugs, consult "How to get a WebGL Implementation")
New & Upcoming WebGL Extensions

**OES_drawBuffersIndexed**

- Enhances multiple draw buffer functionality
- This extension provides the ability to:
  - enable or disable blending
  - set the blend equations
  - set the blend functions
  - set the color write masks
  - all per color output!
- This extension was specifically requested by the 3D Formats working group to implement advanced materials (e.g., that use dual depth peeling) more efficiently
- Extension approved by WebGL WG; coming to all browsers soon
- Can be tested in Chrome today by enabling WebGL Draft Extensions in `about:flags`
  - Please file any bugs on [crbug.com](http://crbug.com), WebGL component
Upcoming WebGL Extensions

**Base Vertex/Base Instance & Multi-Draw Variation**

- Provide control of BaseVertex, for indexed draw calls, and BaseInstance, for instanced draw calls
- Multi-draw variants are provided as well
- Allow reuse of index buffers to draw multiple different geometries from the same set of vertex buffers
- Reduce CPU and memory overhead in certain scenarios
- If you've needed these draw parameters, please try the extensions and provide your feedback
- Can be tested in Chrome today by enabling WebGL Draft Extensions in `about:flags`
  - Please file any bugs on [crbug.com](http://crbug.com)
- Will come to all browsers shortly after community approval
WebGPU

- An upcoming “modern” style graphics API for the Web
  - “Prevalidated” style - pipeline objects, bind groups
  - Compute shaders, shader storage
  - No global state
  - ... and much more
  - Foundation for future features like bindless, raytracing, shader features

- Under development on GitHub at the W3C
  - Thank you to Khronos for hosting us here!
WebGPU - Standardization Updates

- Standardization continues; conformance testing in high gear
  - Aiming to reach 1.0 in 2022 Q2 (spec and conformance tests)
- Shading language under rapid development
  - Feature completeness and numerous language refinements
  - Ergonomics (reduced verbosity, type inference)
  - Static analysis of control flow and uniformity
  - Cross-compilers from Mozilla and Google
  - [https://www.w3.org/TR/WGSL/](https://www.w3.org/TR/WGSL/)
- API spec driving toward 1.0 - recent areas of focus:
  - Video and canvas interop, color management
  - Privacy and security
  - Ergonomics, lifting restrictions, defining optional features
  - Bringing the spec up to date, fleshing out details
  - [https://www.w3.org/TR/webgpu/](https://www.w3.org/TR/webgpu/)
WebGPU - Implementation Updates

- Available to try today in Chrome+Firefox
  - For local development, test the latest browser code:
    - Chrome Canary: `enable-unsafe-webgpu` in `about:flags`
    - Firefox Nightly: set `dom.webgpu.enabled` in `about:config`
      (don’t leave these enabled while browsing the web)
  - Mostly, but not fully, interoperable, due to changing spec

- Chrome
  - Origin Trial allows you to publish WebGPU apps directly to end users
    - Mac, Windows, Chrome OS
    - Extended through Chrome 101 (ending mid-May)
    - **Breaking changes by design - you must keep your content up to date**
    - On Chrome Stable, so you may need temporary polyfills for newer API changes
    - Instructions: [web.dev/gpu](https://web.dev/gpu)
  - Aiming for 1.0 release around Chrome 102~103
    - Linux/Android soon afterward
WebGPU - Resources

- Brandon Jones’ presentation in this meetup

Articles:
  - [web.dev/gpu](https://web.dev/gpu) Tutorial on getting started with WebGPU
    - More articles linked from here

Samples:
  - Austin Eng’s [WebGPU Samples](https://github.com/AustinEng/webgpu-samples) (Github)
    - Journeys from your first triangle, to real-world compute & graphics examples
  - Brandon Jones’ [Clustered Shading, Metaballs, and Spookyball](https://brandonjones.github.io/webgpu-clustering/) demos
    - Real-world usage of the WebGPU API, compute shaders, and rendering techniques
  - Shrek Shao’s [WebGPU Deferred Renderer](https://shrekshao.github.io/webgpu-deferred-renderer/)
    - Shows how to do deferred shading in WebGPU, complete with debug views
WebGPU - Resources

Projects with WebGPU backends well under development:
- Babylon.js, Three.js, TensorFlow.js, and others
- wgpu and the ecosystem of Rust WebGPU projects

Shader compilers
- Compile from \{ WGSL, Vulkan SPIR-V \} to \{ WGSL, SPIR-V, HLSL, MSL \}
- Good for:
  - Seeing what the WGSL language looks like
  - Converting your existing shaders
    - Use Glslang to compile existing GLSL to Vulkan SPIR-V
      - (glslangValidator or glsllc from the Vulkan SDK)
- Google’s Tint: https://dawn.googlesource.com/tint
- Mozilla’s Naga: https://github.com/gfx-rs/naga
WebGPU - Contributing

Contributions welcome!

- Try the API! File API issues and browser bugs
- Try out WebGPU via Babylon.js, Three.js, TensorFlow.js, etc.
- Publish sites using WebGPU Origin Trial
- Help with conformance testing
- Contribute samples/demos using WebGPU
- Join the conversations on the Matrix chat
Presentations

Today’s presenters:
- Brandon Jones (Google) - WebGPU Best Practices & Demos
- Takahiro Aoyagi (Mozilla) - online editor for WebGPU’s WGSL shading language
- Jaume Sánchez (Independent) - Codevember Series
- Donovan Hutchence (PlayCanvas) - New Rendering Features in PlayCanvas

Feel free to type your questions into the Q&A panel at any time!

We’ll answer them live at the end of the session.