SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

KHRONOS® GROUP

WebGL

WebGPU

WebGL & WebGPU BOF

July 31, 2024

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

KHRONOS GROUP

WebGL

WebGPU

WebGL & WebGPU Updates

Ken Russell, Google
On Behalf of the WebGL WG and WebGPU CG

Agenda

WebGL Updates

- Extension Promotions
- ANGLE/Metal Progress
- Shader Pixel Local Storage Extension

WebGPU Updates

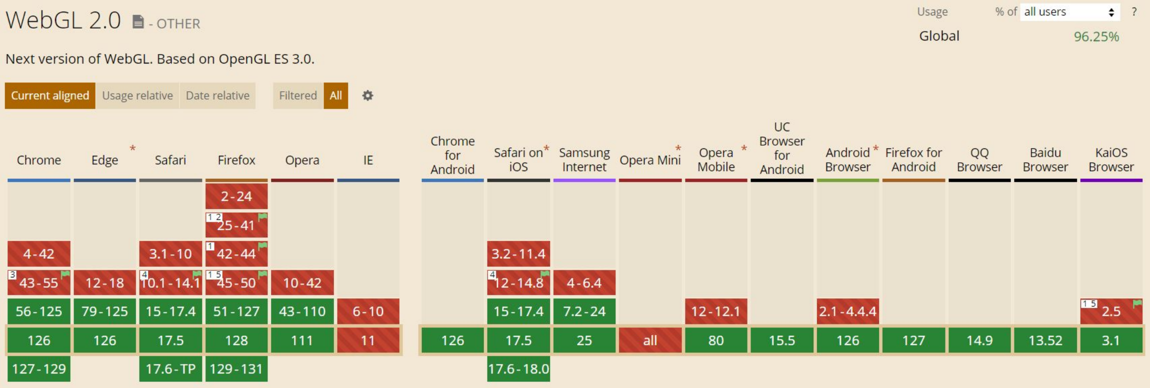
- Standardization
- Ecosystem
- Kotlin Bindings
- Three.js' WebGPU Backend
- WebGPU Samples
- Partnerships, Resources and Contributions

Call to Action: Join WebGL & WebGPU Communities

WebGL Update

- **Khronos is fully supporting development of WebGPU at W3C**
  - Working for a smooth transition for developers between WebGL and WebGPU
  - WebGPU brings GPU Compute to the Web using Vulkan/DX12/Metal backends
- **WebGL is pervasive and will be used by many applications for many years**
  - Khronos is evolving the WebGL specification and supporting multiple implementations
  - ANGLE's Metal backend supports WebGL 2.0 in Safari on macOS/iOS
    - Shipping in Chrome on Mac/ARM; coming to Mac/Intel
  - Firefox's WebGL implementation is similarly advancing

WebGL 2.0 is available on >96% of browsers



WebGL 2.0 - OTHER

Usage % of all users ?

Global 96.25%

Next version of WebGL. Based on OpenGL ES 3.0.

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
			2-24													
			[1] [2] 25-41													
4-42		3.1-10	[1] 42-44				3.2-11.4									
[3] 43-55	12-18	[4] 10.1-14.1	[1] [5] 45-50	10-42			[4] 12-14.8	4-6.4								
56-125	79-125	15-17.4	51-127	43-110	6-10		15-17.4	7.2-24		12-12.1		2.1-4.4.4				[1] [5] 2.5
126	126	17.5	128	111	11	126	17.5	25	all	80	15.5	126	127	14.9	13.52	3.1
127-129		17.6-TP	129-131				17.6-18.0									

New WebGL Extensions

- **Multiple useful extensions have been added to WebGL over the past year**
  - Ported from OpenGL ES to WebGL
- **These are now shipping in browsers**
  - Enhance, speed up, and simplify applications
  - Always test for the presence of the extension and include fallback paths

EXT_clip_control

EXT_conservative_depth

EXT_depth_clamp

EXT_polygon_offset_clamp

EXT_render_snorm

EXT_texture_mirror_clamp_to_edge

NV_shader_noperspective_interpolation

OES_sample_variables

OES_shader_multisample_interpolation

WEBGL_blend_func_extended

WEBGL_clip_cull_distance

WEBGL_polygon_mode

WEBGL_render_shared_exponent

WEBGL_stencil_texturing

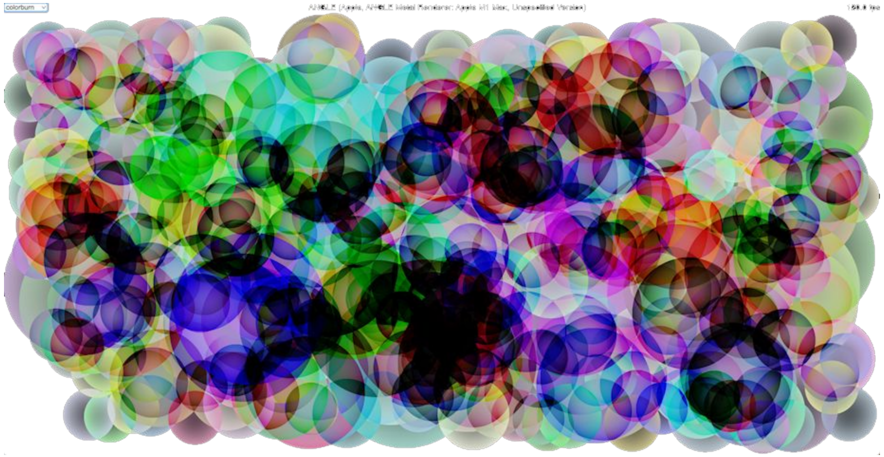
- **Thanks especially to Alexey Knyazev for driving these!**

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
  - Passes native OpenGL ES 2.0/3.0 CTS on Apple Silicon
- **Chrome is ramping up shipment on Intel CPU Macs**
  - Already shipping on Apple Silicon
- **Non-blocking program linking support has been implemented**
  - Already exposed as KHR_parallel_shader_compile
  - Available in Chrome Canary today
- **Thanks to Alexey Knyazev for many fixes, the full WebGL 2.0.1 conformance suite is passing on ANGLE's Metal backend!**
  - Paves the way for the next spec and test suite snapshot

Pixel Local Storage Extension

- **Programmable blending and other use cases**
  - Developed by Chris Dalton from Rive with significant contributions from Alexey Knyazev
- **Specification being finalized**
  - In Draft in Chrome Canary
  - Implementation in ANGLE is tracking the spec and can ship soon afterward
  - Specification | Source Code | Live demo implements blend_equation_advanced
  - Enable WebGL draft extensions in about:flags


colorburn
ANGLE (Apple, ANGLE Metal Renderer: Apple M1 Max, Unspecified Version)
180.0 fps

WebGPU

A "modern" graphics API for the Web:

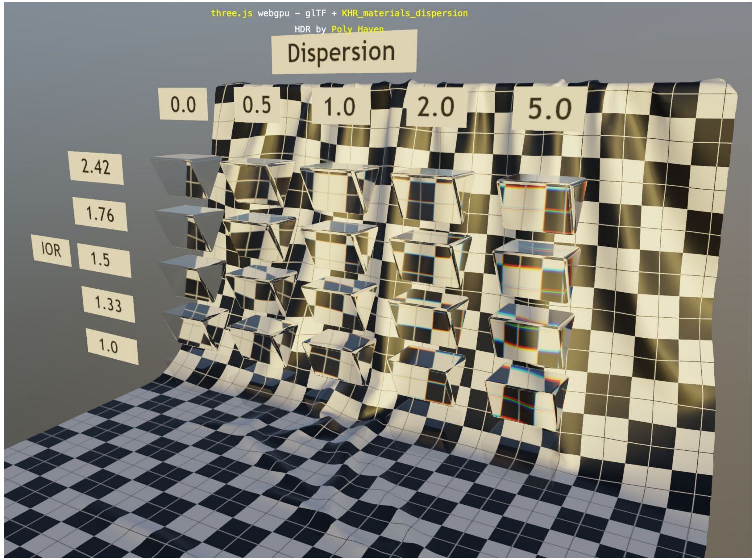
- A successor to WebGL, not a replacement.
- Compute shaders on the Web!
- Lower overhead API
- Foundation for future features (bindless, ray tracing, multithreading …)

Development happens on GitHub and at the W3C

- Anybody can join and participate in the development!

WebGPU Updates

- The WebGPU ecosystem is solidifying!
- Firefox Nightly and Safari Technology Preview have WebGPU support today
  - Try your content and ensure it works across browsers!
- WebGPU specification is nearing Candidate Recommendation!
  - Aim to transition to Living Standard afterward
- Lots of ecosystem progress including Three.js's WebGPU backend

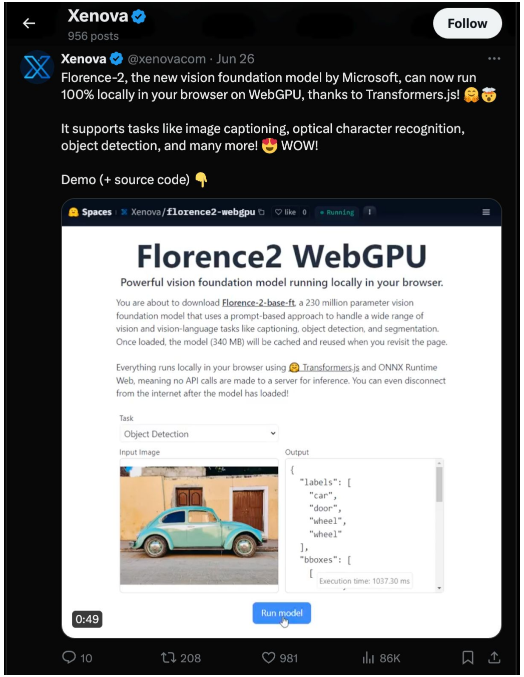


three.js webgpu – glTF + KHR_materials_dispersion

HDR by Poly Haven

WebGPU Updates

- Tons of excitement for WebGPU in the Web AI space
- Hugging Face's Transformers.js now runs AI models in the browser using WebGPU!
  - Thanks to Microsoft's ONNX Runtime and WebGPU backend
- See HF's Models/Datasets/Spaces using WebGPU, and especially Xenova's updates



Xenova @xenovacom · Jun 26

Florence-2, the new vision foundation model by Microsoft, can now run 100% locally in your browser on WebGPU, thanks to Transformers.js! 🤗🤯

It supports tasks like image captioning, optical character recognition, object detection, and many more! 😍 WOW!

Demo (+ source code) 👇

Florence2 WebGPU

Powerful vision foundation model running locally in your browser.

You are about to download Florence-2-base-ft, a 230 million parameter vision foundation model that uses a prompt-based approach to handle a wide range of vision and vision-language tasks like captioning, object detection, and segmentation. Once loaded, the model (340 MB) will be cached and reused when you revisit the page.

Everything runs locally in your browser using 🤗 Transformers.js and ONNX Runtime Web, meaning no API calls are made to a server for inference. You can even disconnect from the internet after the model has loaded!

Task: Object Detection

Input Image

Output

```
{
  "labels": [
    "car",
    "door",
    "wheel",
    "wheel"
  ],
  "bboxes": [
    [
```

Execution time: 1037.30 ms

Kotlin Bindings

- Google's Android team will be launching Kotlin bindings for WebGPU over the next several months!
- These Kotlin bindings will be the main way for Java and Kotlin applications to use WebGPU on Android (and available on other platforms through Kotlin-Multi-Platform)
- Initially this will be offered as an Android Jetpack library; longer term we plan to integrate it into the system

Kotlin Bindings

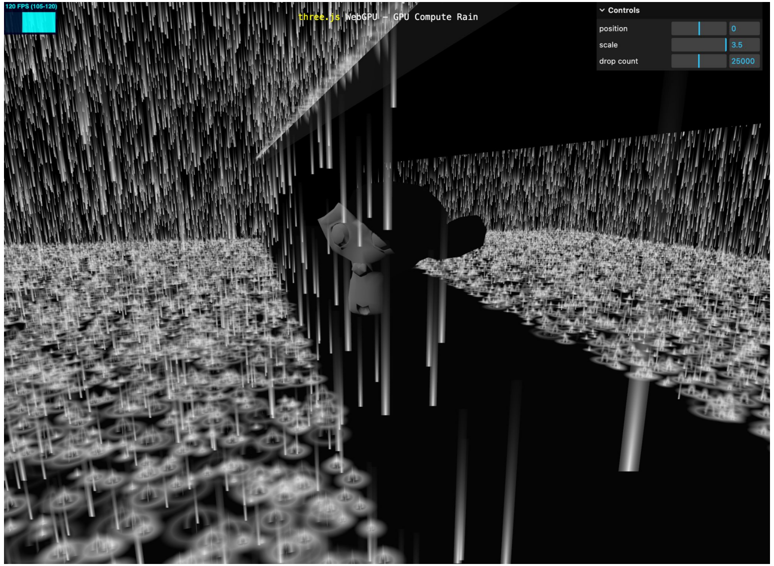
- **What does this mean for my Android development?**
  - If you are a high performance game, or written in C/C++, use Vulkan
    - It’s powerful, scalable, and gives in-depth control of the GPU
  - If you’re an application and need to use graphics, use WebGPU
    - It’s powerful, scalable, and easy-to-use, and you can write once and run (almost) anywhere
  - If you’re doing compute (ML, LLMs, AI!, …) you should be building on WebGPU rather than through OpenCL for long-term support and expansion

We will be trying to steer new content away from OpenGL ES

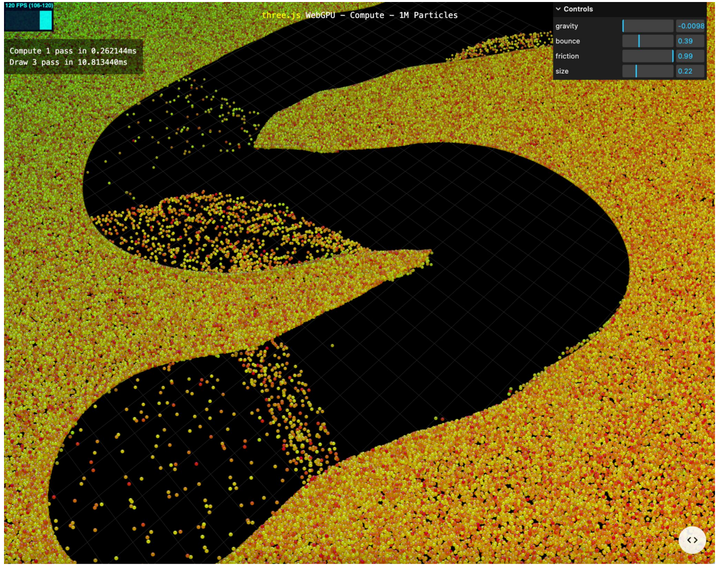
OpenGL ES will instead be supported on top of Vulkan through the ANGLE framework

Three.js' WebGPU Backend

- Compute + Rendering



position 0
scale 3.5
drop count 25000



```
Compute 1 pass in 0.262144ms
Draw 3 pass in 10.813440ms
```

gravity -0.0098
bounce 0.39
friction 0.99
size 0.22

```
async function animate() {
        stats.update();
        await renderer.computeAsync( computeParticles );
        await renderer.renderAsync( scene, camera );
```

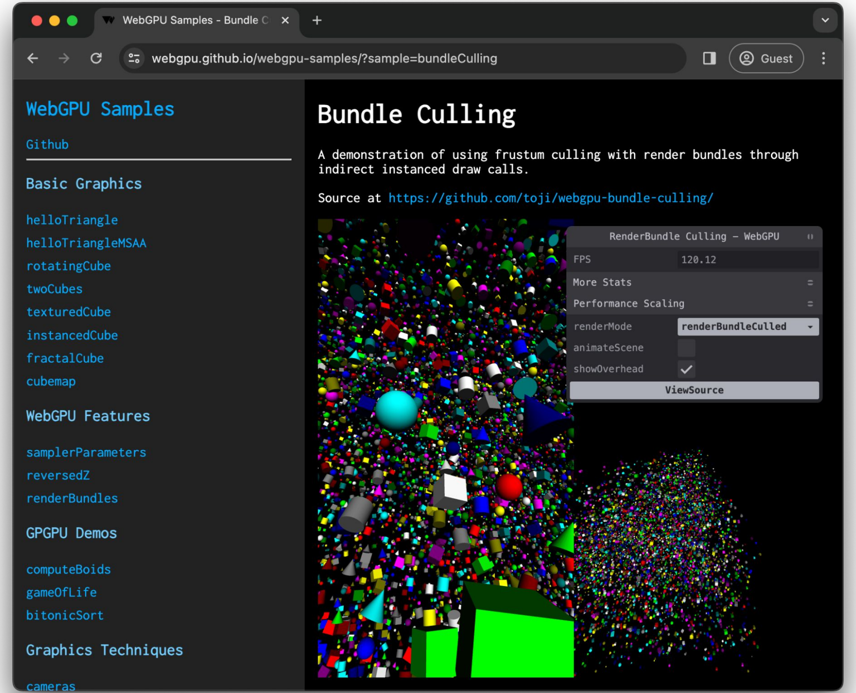
Three.js' WebGPU Backend

- **Much recent progress on Three.js's WebGPU backend**
- **There are already many WebGPU examples in Three's examples**
  - Several of which use compute!
- **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 re-architecture 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 Samples - Bundle C

webgpu.github.io/webgpu-samples/?sample=bundleCulling

Guest

WebGPU Samples

Github

Basic Graphics

helloTriangle
helloTriangleMSAA
rotatingCube
twoCubes
texturedCube
instancedCube
fractalCube
cubemap

WebGPU Features

samplerParameters
reversedZ
renderBundles

GPGPU Demos

computeBoids
gameOfLife
bitonicSort

Graphics Techniques

cameras

Bundle Culling

A demonstration of using frustum culling with render bundles through indirect instanced draw calls.

Source at https://github.com/toji/webgpu-bundle-culling/

RenderBundle Culling – WebGPU

FPS 120.12

More Stats

Performance Scaling

renderMode renderBundleCulled

animateScene

showOverhead ✓

ViewSource

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!

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!

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!**

Thank you!

A recording of this presentation will be available at
https://www.khronos.org/events/siggraph-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