



SV WebGL/WebVR GDC 2019 Meetup

The Khronos Group

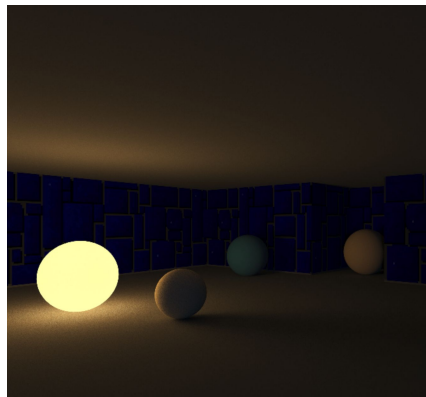
WebGL Update from Browser Implementers

- Demos
- WebGL 2.0 Update
- WebGL 2.0 Compute
- Key [extensions](#) being developed
 - [KHR_parallel_shader_compile](#)
 - [WEBGL_multi_draw](#) and [WEBGL_multi_draw_instanced](#)
 - [WEBGL_video_texture](#)
- WebGL in multithreaded WebAssembly
- Acknowledgments
- Speaker List

Demos



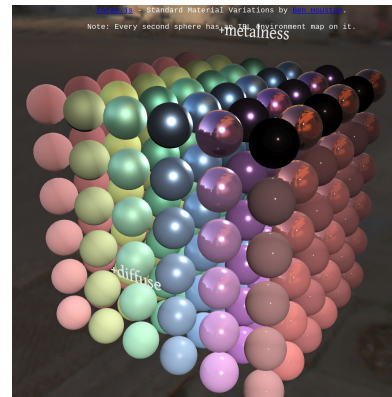
[Filament](#)



[Wolfenstein](#)
[Ray-Traced with](#)
[WebGL 1.0](#)



[Babylon.js](#)



[Three.js](#)

WebGL 2.0 Update

- WebGL working group is focusing on conformance - getting all implementations to pass the top-of-tree conformance test suites
 - James Darpinian, Google; Jeff Gilbert, Mozilla; Lin Sun, Intel
- Many corner cases of the OpenGL and OpenGL ES specs have been uncovered and resolved since last snapshot
- Will lead to improved portability of applications
- Also resolving bug reports from customers and turning these into conformance tests where applicable and possible
- Please keep these coming!

WebGL 2.0 Compute

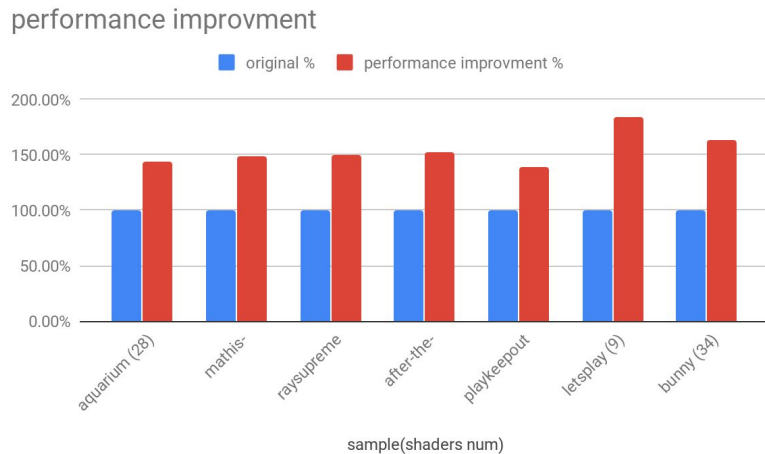
- Single largest recent WebGL advancement is support for compute shaders
- Developed by Intel's Web Graphics team in Shanghai
 - Jiajia/Jiawei/Xinghua/Jie/Jiajie/Yunfei/Yizhou/Yunchao
- Adds OpenGL ES 3.1 compute shaders to WebGL
- [Draft specification](#) is online
- Available in current Chromium builds on Windows and Linux

Trying WebGL 2.0 Compute

- Use Chrome Canary on Windows or Dev Channel on Linux
- On Windows:
 - `--use-cmd-decoder=passthrough --enable-webgl2-compute-context`
 - Optionally: `--use-angle=gl`
- On Linux:
 - `--use-cmd-decoder=passthrough --enable-webgl2-compute-context --use-gl=angle`
- First [ComputeBoids demo from Intel](#)
- More [compute shader demos](#) coming online
- Good way to start experimenting with compute shaders on the web today
- Discuss on [webgl-dev-list](#)

KHR_parallel_shader_compile

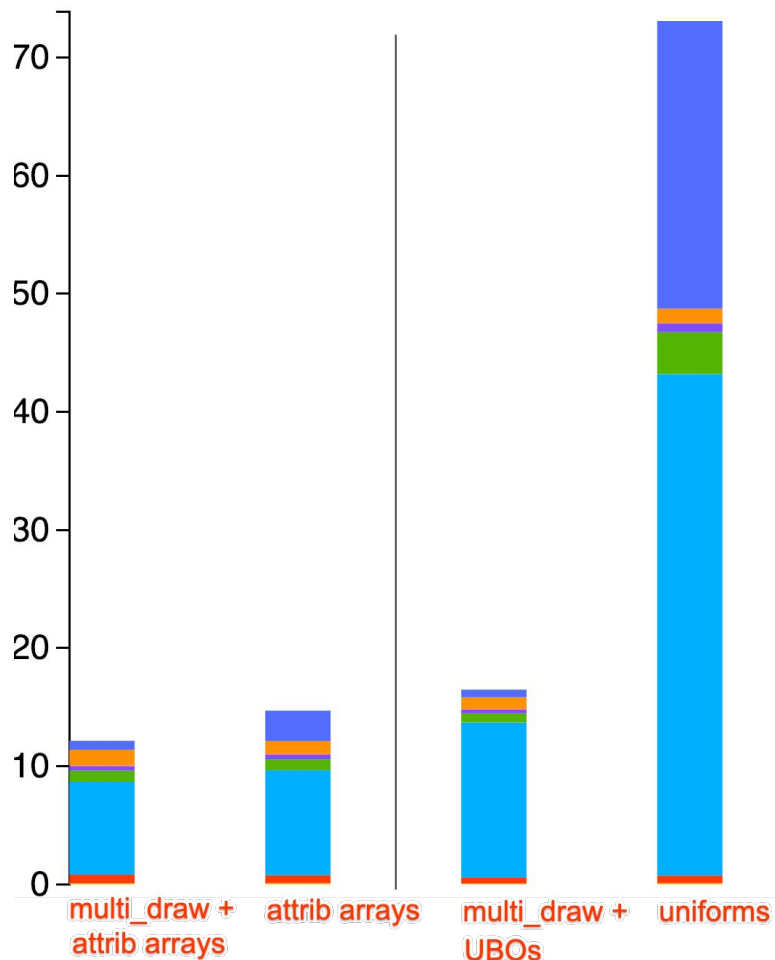
- [Extension](#) being developed by Intel's Web Graphics team in Shanghai
 - Jie Chen
- Fixes longstanding customer complaints about slow WebGL shader compiles
- Compiles become async and parallel with small app changes
- Try in Chrome today with `--enable-webgl-draft-extensions`



Multi-Draw

- [WEBGL_multi_draw](#) and [WEBGL_multi_draw_instanced](#) decrease the CPU overhead of issuing draw calls
 - Austin Eng and Kai Ninomiya, Google
- Application receives gl_DrawID in shaders; works well with uniform updates batched into UBOs with WebGL 2.0
- Supported via emulation even on platforms without the native multi-draw extensions
- [Results from microbenchmarks](#) are impressive: 3-6x improvements in common case, up to 70x (!) in some situations

ms Mac AMD Thread Breakdown



- thread_renderer_main_cpu_time_per_frame
- thread_renderer_compositor_cpu_time_per_frame
- thread_raster_cpu_time_per_frame
- thread_other_cpu_time_per_frame
- thread_IO_cpu_time_per_frame
- thread_GPU_cpu_time_per_frame
- thread_display_compositor_cpu_time_per_frame
- thread_browser_cpu_time_per_frame

Animometer w/ Attrib Arrays

	Control (ms)	MultiDraw (ms)	Speedup
renderer_main	2.50	0.75	3.33 x
GPU	8.85	7.69	1.15 x
frame_time	25.32	23.27	1.09 x

Animometer w/ Uniforms

	Control (ms)	MultiDraw (ms)	Speedup
renderer_main	19.44	0.66	29.45 x
GPU	34.59	13.24	2.61 x
frame_time	38.90	19.64	1.98 x

Trying multi-draw today

- Test in Chromium with `--enable-webgl-draft-extensions`
- Please tell us how it's working for you on [webgl-dev-list](#)

WEBGL_video_texture

- [Extension](#) being developed by Intel's Web Graphics team in Shanghai
 - Shaobo Yan
- Supports zero-copy video uploads into WebGL textures
- [Implementation](#) is landing in Chromium now
- Seeing up to 47% speedups on some content
- Will appreciate your feedback; please watch Khronos' [public_webgl](#) list

WebGL in Multi-threaded WebAssembly

- Support for multithreading in WebAssembly is beginning to ship in browsers
- This makes it increasingly possible to take large, complex C++ code bases and just “compile for the web”
- [Unreal Engine](#) and [Unity](#)’s HTML5 export paths both contain experimental WebAssembly multithreading support
- Can even access this functionality from [Rust](#)!
- Try it in your projects via [Emscripten](#) and please report any problems you encounter

Acknowledgments

Regular Working Group Attendees

Austin Eng, Google
Dean Jackson, Apple
Geoff Lang, Google
James Darpinian, Google
James Helferty, NVIDIA
Jamie Madill, Google
Jeff Gilbert, Mozilla
Jonah Ryan-Davis, Google
Justin Fan, Apple
Kai Ninomiya, Google
Ken Russell, Google
Kimmo Kinnunen, NVIDIA
Rafael Cintron, Microsoft
Shrek Shao, Google

Intel's Web Graphics Team, Shanghai

Yang Gu, Manager
Hao Li
Jiajia Qin
Jiajie Hu
Jiawei Shao
Jie Chen
Lin Sun
Shaobo Yan
Xing Xu
Xinghua Cao
Yizhou Jiang
Yunchao He
Yunfei Hao

Speaker Lineup

- Dave Evans, PlayCanvas
- Andrew Best, Toyota Research Inc
- Alban Denoyel, Sketchfab
- Ricardo Cabello, Three.js
- Gary Hsu, Microsoft
- Robert Long, Mozilla