

GPU Web 2018-02-28

Chair: Corentin

Scribe: Kai, Corentin

Location: Google Hangout

[Minutes from last meeting](#)

TL;DR

- Dean will make an advisory mailing list where we can ask specific questions to ISV / IHVs
- Google talked to Unity but don't have concrete feedback to share.
- String enums in the IDL
 - Investigations in JavascriptCore and V8 indicate that WASM string enums could be made as fast and as fast in WASM and JS.
 - Consensus: design for JS first with strings, while pressuring the WASM CG to find an answer for all Web APIs. Optimize functions if on the critical path.

Tentative agenda

- Advisory mailing list
- Feedback from Unity
- Enum in the IDL
- More on buffer mapping
- Agenda for next meeting

Attendance

- Apple
 - Dean Jackson
 - Myles C. Maxfield
- Google
 - Corentin Wallez
 - James Darpinian
 - Kai Ninomiya
- Microsoft
 - Rafael Cintron
- Mozilla
 - Dzmitry Malyshau

- Jeff Gilbert
- Markus Siglreithmaier
- Joshua Groves

Administrative

- CW: Vulkan released adopter program for SPIRV (and file formats). Allows projects/companies to become adopters of SPIRV (granting trademarks, etc.) without submitting conformance because there is no CTS.

Advisory mailing list

- CW: In meeting with Unity, they recommended we create an advisory list where we can ask targeted questions to ISVs (engine developers)
- DJ: AI to take care of this
- CW: We invite add Unity and everyone else we've been talking too.

Feedback from Unity

- CW: Had a meeting with Unity about WebGPU. Didn't take notes, not a lot of concrete feedback yet. 3 things though:
 - Unity: Metal is specific to Apple GPUs and don't necessarily map to others, but Vulkan/D3D12 don't necessarily provide the portability we want.
 - DM: Roblox feedback was about Metal 1. Metal 2 should provide a lot more.
 - CW: We should ask Roblox and Unity. The discussion was informal so we cannot extract actionable feedback right now.
 - Unity: Use a non-text language. Same reason as Khronos. If the compiler frontend changes it can break your application - huge pain point for developers and drivers. They're not attached to SPIR-V in particular but out of the existing designs it's definitely what they like most.
 - They use a HLSL-like language for built-in shaders, also allow users to write GLSL / HLSL / MSL if they want to go platform specific.
 - Unity: Would love a native API abstraction library so they can port the engine to the new API **without** simultaneously porting to web.
 - Unity: Would love to be involved, suggest an advisory mailing list.

Enums in the IDL

- KN: Brad Nelson thinks that string enums in WASM can be as almost fast as in JS. Reasonable to rely on them as long as they are fast in the bindings. All we need is inside WASM a mapping from integer to string reference. How fast that is is unclear but probably pretty fast. Everything other than that is same between JS and WASM.

- MM: In reply in email, answered that JS already has a way to do this with strings so we should do it. New API in WASM so we can make a different API.
- KN: Problem is that the WASM API is done by providing JS functions. Nobody wants APIs only visible to WASM because it would create a new API surface, and duplicate APIs for any group that exposes APIs to WASM. Idea is to bind Javascript API directly in WASM and have guarantees of when the call will be done without going through the JS VM.
- DJ: Not sure if having a different API for WASM and JS is an issue. Can design for JS first then fix them when we get to WASM bindings.
- KN: Already talking about WASM specific stuff with descriptors vs. dicts. Cannot have an API visible to WASM but not JS. Could make an API that takes integer but JS will have to be the one that pass that API to WASM.
- RC: MS's TAG representative said we should focus on JS first then figure out WASM. For MVP we should go JS and pick a solution there.
- JG: Agree and if we find WASM perf that's not good enough we can amend things.
- CW: concern that doing it post MVP would be a big change.
- JG: Maybe not, only need to change the perf critical path.
- KN: Could just take enums out of the critical path.
- CW: Concern that we can't postpone because shape of WASM API takes a while to figure out.
- KN: Basically we should keep pushing the WASM CG so that it stays on their plate.
- RC: Agreed, there's a lot of APIs on the Web that rely on string enums. We're not unique. We should pressure the WASM CG to find an answer just not for us but for everyone.
- JG: Don't think we want two ways for everything but maybe fore some things that are on the critical path. Matrix transform is one example.
- DJ: CSS provided a typed API for that. We should do the simple way then optimize.
- MM: We are not suggesting to parse stuff only string compares
- CW: We can do simple then optimize
- CW: Consensus: pressure WASM group and go string + optimize critical

More on buffer mapping

<https://github.com/gpuweb/gpuweb/pull/49>

- RC: Question about is in a usage vs. has a usage.
- Everyone will take a look for next week.
- AI: JG, talk make a propo
- AI: Google ask unity to weigh on the big questions.

Agenda for next meeting

More buffer mapping

Something else from the big questions?