Implementing WebGPU For Unity

Brendan Duncan
Senior Graphics Engineer
Unity Technologies
Unity web player

→ WebAssembly
  — C++ and C# compiled via Emscripten
  — Javascript bindings integrate into browser

→ Graphics Device
  — Graphics API abstraction layer in Unity
  — WebGL implemented with GLES device

→ Shader Compilation
  — HLSL translated to target shading language
  — WebGL compiles HLSL to GLSL
Unity WebGPU

→ **WebAssembly**
  - Custom WebGPU Emscripten binding
  - [https://github.com/jui/wasm_webgpu](https://github.com/jui/wasm_webgpu)

→ **Graphics Device**
  - Integrates WebGPU into Unity
  - Pipeline and BindGroup caching
  - Resource management

→ **Shader Compilation**
  - HLSL compiles to WGSL
  - $\text{HLSL} \rightarrow \text{HLSLcc} \rightarrow \text{GLSL} \rightarrow \text{GLSlang} \rightarrow \text{SPIR-V} \rightarrow \text{Tint} \rightarrow \text{WGSL}$

→ **Unity 2023.3+**
  - Enabled via project settings
  - Can fall back to WebGL if WebGPU is not available
WebGPU enabled features

**Compute Shaders**
- GPU Skinning
- VFX Graph
- General purpose compute

**Forward+ Rendering**
- No per-object limit for lights.

**Pipeline Caching**
- Pipeline creation and shader compilation is cached to improve performance.
Challenges

→ WebGPU Limits / Missing Features
  - WebGPU limits more restrictive than other APIs.
  - No synchronous gpu readback.
  - Strict validation errors.
  - Texture format limitations.
  - Missing features such as texel buffers.

→ Shader Issues
  - Uniformity Analysis produces false negatives.
San Francisco ➔ March 18-22

Debugging: WebGPU Inspector

→ No existing tools for debugging WebGPU
→ Dawn for native builds, debug with VS, RenderDoc
→ Developed **WebGPU Inspector**
  - Chrome extension graphics debugger
  - Frame captures
  - Live GPU object inspection: textures, buffers, shaders, etc.
  - Recording playback
  - Bug repo generation
→ Developed Outside of Unity, open source

https://github.com/brendan-duncan/webgpu_inspector
More information at