

# WebGL and Why You Should Target It

---

Khronos Developer Day, GDC 2018

The logo for Khronos Group, featuring the word "KHRONOS" in a bold, black, sans-serif font. The letter "O" is replaced by a red, stylized, curved shape that resembles a lens or a stylized "O". Below "KHRONOS" is the word "GROUP" in a smaller, black, sans-serif font.

**KHRONOS**  
GROUP

The WebGL logo, featuring the word "WebGL" in a bold, red, sans-serif font. The letter "W" is partially enclosed by a red, curved shape that resembles a lens or a stylized "W". A small "TM" trademark symbol is located to the right of "WebGL".

**WebGL**™

# THREE.js Demos

- The most commonly used WebGL library
- <http://demos.littleworkshop.fr/infinitown>
- <http://www.hellorun.helloenjoy.com/>
- <http://www.playmapscube.com/>
- <https://stinkmoji.cool/>
- <http://devx.ddd.it/en/experiments/>

# WebGL Fundamentals

- Built-in 3D graphics API in browsers
- Based on industry-standard OpenGL ES APIs
- Implement on one device, runs everywhere
- No installation step – completely frictionless
  - Get lots of eyeballs on your games
- Many options for deployment
  - Web pages
  - [PWAs](#)
  - [Electron/Cordova](#)
  - Native apps using WebView

# WebGL Fundamentals (continued)

- High API quality
  - Conformance suite: 2.0.0 contains 340,000 test cases
  - Workarounds for numerous driver bugs
- WebGL 1.0 implements OpenGL ES 2.0 APIs
  - Supported on Firefox, Chrome, Edge, Safari
  - Support rate: 98% [\*]
- WebGL 2.0 implements OpenGL ES 3.0 APIs
  - Currently supported on Firefox, Chrome
  - Edge and Safari intend to implement it
  - Support rate: 41% [\*] and increasing

[\*] Data from <https://webglstats.com/>

# WebGL 2.0 Features

3D textures

Compressed textures

Texture storage

Seamless cubemaps

NPOT textures

Float textures

Integer textures

sRGB textures

Instanced rendering

Transform feedback

Multiple render targets

Query objects

Sync objects

Sampler objects

Uniform blocks

Vertex array objects

Integer vertex attributes

GLSL ES 3.0 shaders

# WebGL + glTF

- glTF: a royalty-free specification for the efficient transmission and loading of 3D scenes and models by applications
  - <https://www.khronos.org/glTF/>
- WebGL + glTF: streamlines authoring workflows and enables interoperable use of content across the industry
- Two of the major WebGL libs, Three.js and Babylon.js (among others) support glTF 2.0

# WebGL + WebAssembly + Emscripten

1. WebAssembly (wasm): a new portable, size- and load-time-efficient format suitable for compilation to the web
  - <http://webassembly.org/>
  - [Now shipping in all major browsers](#)
2. WebGL + WebAssembly: almost native graphics apps on the web
3. Emscripten: source-to-source compiler that produces a subset of JavaScript (asm.js) or WebAssembly
  - Easily brings existing (C++) code bases to the web

# A Small Engine That Can Be Compiled with WASM

- <https://github.com/flooh/oryol>
- Demos
  - <http://flooh.github.io/oryol/asmjs/InfiniteSpheres.html>
  - <http://flooh.github.io/oryol/asmjs/GPUParticles.html>
  - <http://flooh.github.io/oryol-samples/asmjs/BulletPhysicsCloth.html>



# Unity and Unreal Engine Support WASM Export

- Unity WASM demos
  - <http://webassembly.org/demo/>
  - [https://files.unity3d.com/christopheri/webgl\\_linear/index.html](https://files.unity3d.com/christopheri/webgl_linear/index.html)
- Unreal Engine WASM demo
  - <https://s3.amazonaws.com/mozilla-games/ZenGarden/EpicZenGarden.html>

# PlayCanvas: Another Game Engine for the Web

- Pure JavaScript game engine with full functionality and small download size
- <https://robostorm.io/>
- After the Flood: <https://playcanv.as/b/SFit5Ode/>
- Editor: <https://playcanvas.com/features>

# Transform Feedback / Particle System Demos

- Babylon.js particle systems
  - Documentation: <https://doc.babylonjs.com/babylon101/particles>
  - Demo: <https://www.babylonjs-playground.com/?12>
- More from the web
  - <https://gpfault.net/assets/jsdemos/webgl2-particles/03/index.html>
  - [http://webglsamples.org/WebGL2Samples/#transform\\_feedback\\_instanced](http://webglsamples.org/WebGL2Samples/#transform_feedback_instanced)
  - <http://uber.github.io/deck.gl/examples/wind/>

# Visualization and Authoring

- Uber visualization toolkits
  - <https://uber.github.io/deck.gl/#/>
  - <https://uber.github.io/luma.gl/#/>
  - <https://uber.github.io/react-map-gl/#/>
  - <https://uber.github.io/react-vis/>
- Autodesk Forge
  - <https://viewer-rocks.autodesk.io/>
- SketchFab 3D models
  - <https://sketchfab.com/>
- Figma: Interactive collaborative design tools
  - <https://www.figma.com/>

# Advanced Rendering

- Subsurface scattering and translucency
  - <https://blog.sketchfab.com/sketchfab-now-supports-subsurface-scattering-translucency/>
- Path Tracing
  - <http://madebyevan.com/webgl-path-tracing/>
  - <https://www.tyro.github.io/caffeine/>
- Physically Based Rendering
  - <https://labs.sketchfab.com/siggraph2014/>
  - Casino: <http://playcanv.as/p/LpmXGUe6/?overlay=false&scrolling=true>
- Water Simulation
  - <http://madebyevan.com/webgl-water/>
- Lighting
  - <http://codeflow.org/webgl/deferred-irradiance-volumes/www/>
- Animation
  - [Sketchfab: Lily & Snout](#)

# Useful Links

- WebGL specs: <https://www.khronos.org/registry/webgl/specs/latest/>
- WebGL frameworks: [https://en.wikipedia.org/wiki/List\\_of\\_WebGL\\_frameworks](https://en.wikipedia.org/wiki/List_of_WebGL_frameworks)
- WebGL engines: [https://en.wikipedia.org/wiki/List\\_of\\_game\\_engines](https://en.wikipedia.org/wiki/List_of_game_engines)
- Chrome experiments: <https://experiments.withgoogle.com/chrome?tag=WebGL>
- WebGL 2.0 samples: <http://webglsamples.org/WebGL2Samples/>
- ShaderToy: <https://www.shadertoy.com/>
- WebGL stats: <https://webglstats.com/>
- WebGL dev list: <https://groups.google.com/forum/#!forum/webgl-dev-list>
- WebGL spec discussion list: <https://www.khronos.org/webgl/public-mailing-list/>