WebGL

WebGL training

History

- Open Graphics Library (OpenGL) (1992.06)
- OpenGL for Embedded Systems (OpenGL ES)
    - OpenGL ES 2.0, based on OpenGL 2.0 (2007.03)
        - Eliminates most of the fixed-function rendering pipeline
    - OpenGL ES 3.0, based on OpenGL 3.3 and 4.2 (2012.08)
- Web Graphics Library (WebGL)
    - WebGL 1.0, based on OpenGL ES 2.0 (2009 ~ 2011.03)
        - Chrome and Firefox use ANGLE to run OpenGL on DirectX
        - Chrome 9 (2011.02), Firefox 4 (2011.03)
    - WebGL 2.0, based on OpenGL ES 3.0 (2013 ~ 2017.01)
        - Chrome 56 (2017.01), Firefox 51 (2017.01)

WebGL 2.0

- Promoted Extensions
  - Multiple Render Targets (WEBGL_draw_buffers) (Deferred rendering)
  - Instanced Drawing (ANGLE_instanced_arrays) (Performance)
  - Vertex Array Objects (OES_vertex_array_object) (Performance)
  - Fragment Depth (EXT_frag_depth) (gl_FragDepth)
  - sRGB Texture (EXT_SRGB) (HDR, PBR)
  - Floating-point textures (OES_texture_float) (Deferred rendering)
  - Depth Textures (WEBGL_depth_texture) (Shadow Mapping)
  - UNSIGNED_INT indices (OES_element_index_uint) (Performance)
  - Standard Derivatives (OES_standard_derivatives) (dFdx, dFdy, and fwidth)
  - Direct texture LOD access (EXT_shader_texture_lod) (textureLod)

WebGL 2.0

- New Features
    - Multisampled Renderbuffers (antialiasing)
    - 2D texture arrays & 3D textures (Cascaded Shadow maps)
    - Transform feedback (Skin, Particle)
    - Uniform Buffer Objects (Performance)
    - Sampler Objects (Performance)
    - Sync Objects (synchronizing between workers)
    - Query Objects (Occlusion queries)
    - Immutable Texture (Performance & Memory)
    - Full support for non-power-of-two textures
    - Texture LOD (TEXTURE_MIN_LOD, TEXTURE_MAX_LOD)

WebGL 1.0 to 2.0

- Can I use WebGL 2.0
- MoveWebGL1EngineToWebGL2
- What's coming in WebGL 2.0
- Babylon.js WebGL2.0
- WebGL 2.0 is Here: What You Need To Know
- WebGL2 What's New
- WebGL2 from WebGL1

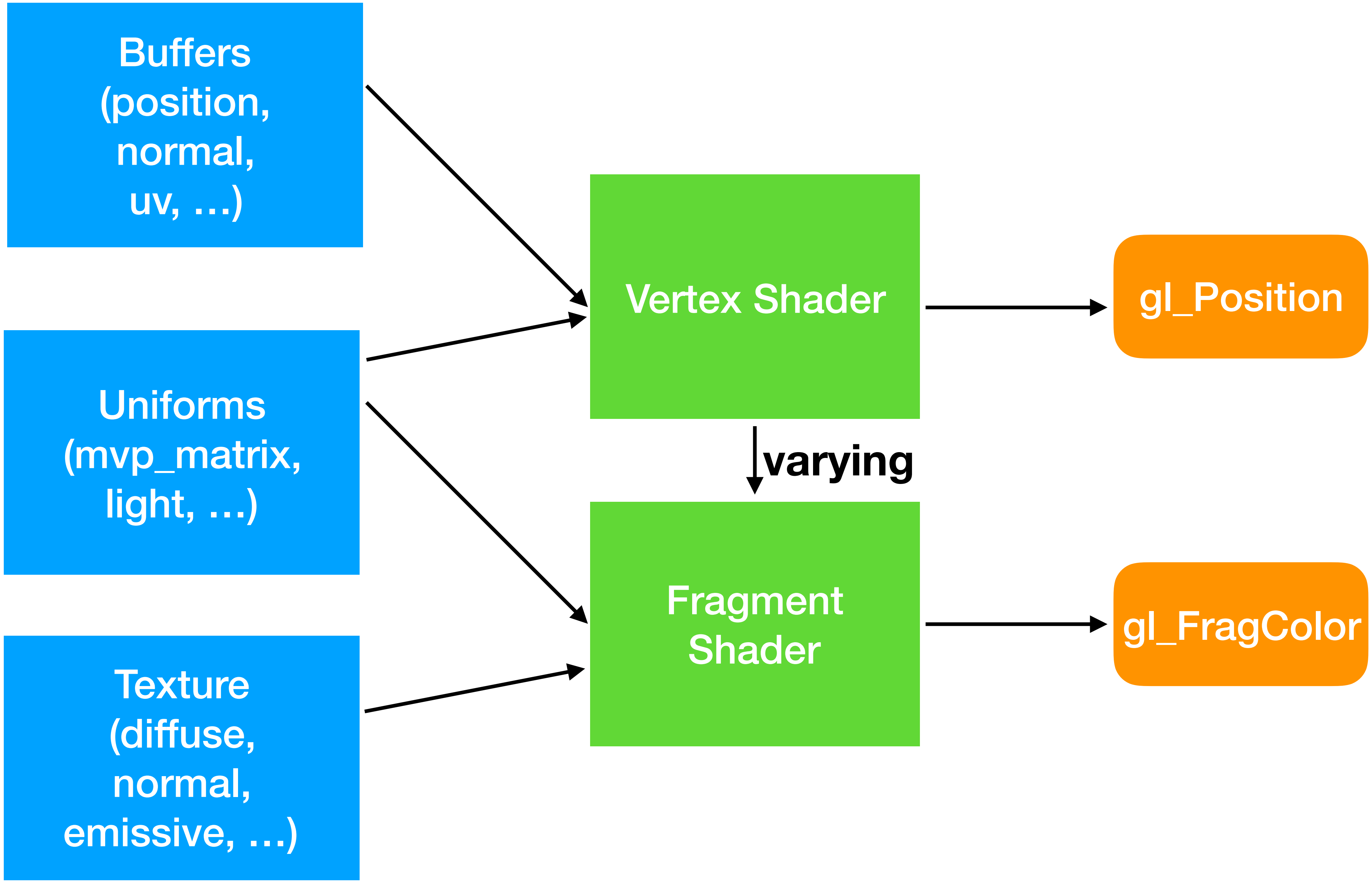
Reference

- WebGL 2.0 Specification
- GLSL_ES_Specification_3.00.pdf
- WebGLRenderingContext
- WebGL2RenderingContext
- Rendering algorithms implemented in raw WebGL 2
- Short and easy to understand samples demonstrating WebGL 2 features
- MDN WebGL tutorial

Books

- WebGL Programming Guide
- WebGL Beginner's Guide
- Professional WebGL Programming
- WebGL Game Development
- WebGL Insights (by Patrick Cozzi)
- WebGL 2.0 API Quick Reference Guide

Overview


Buffers (position, normal, uv, …)
Uniforms (mvp_matrix, light, …)
Texture (diffuse, normal, emissive, …)
Vertex Shader
varying
Fragment Shader
gl_Position
gl_FragColor

Setup

- var ctx = canvas.getContext(contextType, contextAttributes)
    - contextType: "2d", "webgl", "webgl2"
- High DPI
- resize
- viewport
- clearColor
- clear

Shader

- WebGLShader gl.createShader(type)
  - gl.VERTEX_SHADER
  - gl.FRAGMENT_SHADER
- void gl.shaderSource(WebGLShader shader, DOMString source)
- void gl.compileShader(WebGLShader shader)
- GLboolean gl.getShaderParameter(WebGLShader shader, gl.COMPILE_STATUS)
- DOMString gl.getShaderInfoLog(WebGLShader shader)
- void gl.deleteShader(WebGLShader shader)

GLSL

- First line should be: #version 300 es
- precision highp float;
- Types: bool, int, float, vec2, vec3, vec4, mat2, mat3, mat4
- Array: uniform mat4 u_jointMatrix[SKIN_JOINTS_COUNT];
- Struct: struct light { float intensity; vec3 pos; };
- Uniform block: uniform Material { vec4 u_ambient; vec4 u_specular; float u_shininess; };
- Control: void main(), for, while, do, break, continue, return, discard, if, else
- Sampler: sampler2D, sampler3D, sampler2DArray, samplerCube, sampler2DArrayShadow
- Storage Qualifiers: const, in, out, uniform
- Built-In Input & Output
    - Vertex Shader: in: int gl_VertexID, int gl_InstanceID, out: vec4 gl_Position, float gl_PointSize
    - Fragment Shader: in: vec4 gl_FragCoord, bool gl_FrontFacing, vec2 gl_PointCoord
- Built-In Function: sin, pow, abs, mix, step, dot, cross, reflect, texture

Program

- WebGLProgram gl.createProgram()
- void gl.attachShader(WebGLProgram program, WebGLShader shader)
- void gl.linkProgram(WebGLProgram program)
- GLboolean gl.getProgramParameter(WebGLProgram program, gl.LINK_STATUS)
- DOMString gl.getProgramInfoLog(program)
- void gl.useProgram(WebGLProgram program)
- void gl.deleteProgram(WebGLProgram program)

Draw

- void gl.drawArrays(mode, first, count)
    - mode: gl.POINTS, gl.LINE_STRIP, gl.LINE_LOOP, gl.LINES, gl.TRIANGLE_STRIP, gl.TRIANGLE_FAN, gl.TRIANGLES
- void gl.drawElements(GLenum mode, count, GLenum type, offset)
    - type: gl.UNSIGNED_BYTE, gl.UNSIGNED_SHORT, gl.UNSIGNED_INT
- void gl.vertexAttribDivisor(GLuint index, GLuint divisor)
- void gl.drawArraysInstanced(GLenum mode, GLint first, GLsizei count, GLsizei instanceCount)
- void gl.drawElementsInstanced(GLenum mode, GLsizei count, GLenum type, GLintptr offset, GLsizei instanceCount)
- WebGL instancing with ANGLE_instanced_arrays

Uniform

- GLint gl.getProgramParameter(WebGLProgram program, gl.ACTIVE_UNIFORMS)

- WebGLActiveInfo gl.getActiveUniform(WebGLProgram program, GLuint index)

- WebGLUniformLocation gl.getUniformLocation(WebGLProgram program, DOMString name)

- gl.uniform[1234][fi][v](WebGLUniformLocation location, value)

- gl.uniformMatrix[234]fv(WebGLUniformLocation location, GLboolean transpose, Float32Array value)

Attribute

- GLint gl.getProgramParameter(WebGLProgram program, gl.ACTIVE_ATTRIBUTES)
- WebGLActiveInfo gl.getActiveAttrib(WebGLProgram program, GLuint index)
- GLint gl.getAttribLocation(WebGLProgram program, DOMString name)
- void gl.bindAttribLocation(WebGLProgram program, GLuint index, DOMString name)
- void gl.vertexAttribDivisor(GLuint index, GLuint divisor)
- Attribute index
    - gl.bindAttribLocation
    - gl.getAttribLocation
    - layout(location = 0) in vec3 position;

Buffer

- WebGLBuffer gl.createBuffer()
- void gl.bindBuffer(GLenum target, WebGLBuffer buffer)
  - gl.ARRAY_BUFFER
  - gl.ELEMENT_ARRAY_BUFFER
  - gl.UNIFORM_BUFFER
- void gl.bufferData(GLenum target, ArrayBufferView srcData, GLenum usage)
  - gl.STATIC_DRAW
  - gl.DYNAMIC_DRAW
- void gl.bufferSubData(GLenum target, long offset, ArrayBufferView srcData)
- void gl.deleteBuffer(WebGLBuffer buffer)
- void gl.enableVertexAttribArray(GLuint index)
- void gl.disableVertexAttribArray(GLuint index)
- void gl.vertexAttribPointer(GLuint index, GLint size, GLenum type, GLboolean normalized, stride, offset)

VertexArrayObject

- WebGLVertexArrayObject gl.createVertexArray()
- void gl.bindVertexArray(WebGLVertexArrayObject vertexArray)
- void gl.deleteVertexArray(WebGLVertexArrayObject vertexArray)

Texture

- WebGLTexture gl.createTexture()
- void gl.activeTexture(texture)
    - gl.TEXTURE0 ~ 16
- void gl.bindTexture(GLenum target, WebGLTexture texture)
    - gl.TEXTURE_2D, gl.TEXTURE_CUBE_MAP, gl.TEXTURE_3D, gl.TEXTURE_2D_ARRAY
- void gl.deleteTexture(WebGLTexture texture)

Texture

- void gl.texParameter[fi](GLenum target, GLenum pname, GLfloat/GLint param)
    - gl.TEXTURE_MAG_FILTER
    - gl.TEXTURE_MIN_FILTER
    - gl.TEXTURE_WRAP_S
    - gl.TEXTURE_WRAP_T
    - ext.TEXTURE_MAX_ANISOTROPY_EXT
- void gl.pixelStorei(pname, param)
    - gl.UNPACK_FLIP_Y_WEBGL
    - gl.UNPACK_PREMULTIPLY_ALPHA_WEBGL

Texture

- void gl.texStorage2D(GLenum target, GLint levels, GLenum internalformat, GLsizei width, GLsizei height)
- void gl.texStorage3D(target, levels, internalformat, width, height, depth)
- void gl.texSubImage2D(target, level, xoffset, yoffset, format, type, pixels)
- void gl.texSubImage3D(target, level, xoffset, yoffset, zoffset, width, height, depth, format, type, pixels)
- void gl.texImage2D(target, level, internalformat, format, type, pixels)
- void gl.texImage3D(target, level, internalformat, width, height, depth, border, format, type, source)
- void gl.compressedTexImage2D(target, level, internalformat, width, height, border, ArrayBufferView? pixels)
- void gl.compressedTexSubImage2D(target, level, xoffset, yoffset, width, height, format, ArrayBufferView? pixels)
- void gl.compressedTexImage3D(target, level, internalformat, width, height, depth, border, ArrayBufferView srcData, optional srcOffset, optional srcLengthOverride)
- void gl.compressedTexSubImage3D(target, level, xoffset, yoffset, zoffset, width, height, depth, format, ArrayBufferView srcData, optional srcOffset, optional srcLengthOverride)
- void gl.generateMipmap(GLenum target)

Camera & Matrix

- glMatrix
- mat4.fromRotationTranslationScale(out, q, v, s)
- quat.fromEuler(out, x, y, z)
- mat4.perspective(out, fovy, aspect, near, far)
- mat4.lookAt(out, eye, center, up)
- Camera Math
- Understanding the View Matrix
- 3D Projection

Uniform buffer

- GLuint gl.getUniformBlockIndex(WebGLProgram program, DOMString uniformBlockName)
- void gl.uniformBlockBinding(program, uniformBlockIndex, uniformBlockBinding)
- void gl.bindBufferBase(Glenum target, GLuint index, WebGLBuffer buffer)
    - target: gl.TRANSFORM_FEEDBACK_BUFFER, gl.UNIFORM_BUFFER
- void gl.bindBufferRange(target, index, buffer, offset, size)
- void gl.bufferSubData(target, offset, ArrayBufferView srcData)
- void gl.bufferSubData(target, dstByteOffset, ArrayBufferView srcData, srcOffset, length)
- GLint gl.getProgramParameter(WebGLProgram program, gl.ACTIVE_UNIFORM_BLOCKS)
- Array<GLuint> gl.getUniformIndices(WebGLProgram program, Array<DOMString> uniformNames)
- any gl.getActiveUniforms(program, uniformIndices, pname)
- any gl.getActiveUniformBlockParameter(program, uniformBlockIndex, pname)
- DOMString gl.getActiveUniformBlockName(program, uniformBlockIndex)

Framebuffer

- WebGLFramebuffer gl.createFramebuffer()
- void gl.bindFramebuffer(GLenum target, WebGLFramebuffer framebuffer)
    - target: gl.FRAMEBUFFER = gl.DRAW_FRAMEBUFFER, gl.READ_FRAMEBUFFER
- void gl.deleteFramebuffer(WebGLFramebuffer framebuffer)
- void gl.framebufferTexture2D(GLenum target, GLenum attachment, GLenum textarget, WebGLTexture texture, GLint level)
    - attachment: gl.COLOR_ATTACHMENT0(~15), gl.DEPTH_ATTACHMENT, gl.DEPTH_STENCIL_ATTACHMENT, gl.STENCIL_ATTACHMENT
    - textarget: gl.TEXTURE_2D, gl.TEXTURE_CUBE_MAP_[POSITIVE|NEGATIVE]_[X|Y|Z]
- void gl.framebufferTextureLayer(GLenum target, attachment, texture, level, layer)
- void gl.clearBufferfv(GLenum buffer, drawbuffer, values, optional srcOffset)
- void gl.clearBufferfi(GLenum buffer, GLint drawbuffer, GLfloat depth, GLint stencil)
- void gl.blitFramebuffer(srcX0, srcY0, srcX1, srcY1, dstX0, dstY0, dstX1, dstY1, GLbitfield mask, filter)
- GLenum gl.checkFramebufferStatus(target)

Renderbuffer

- WebGLRenderbuffer gl.createRenderbuffer()
- void gl.bindRenderbuffer(gl.RENDERBUFFER, WebGLRenderbuffer renderbuffer)
- void gl.deleteRenderbuffer(WebGLRenderbuffer renderbuffer)
- void gl.renderbufferStorage(gl.RENDERBUFFER, GLenum internalFormat, width, height)
  - internalFormat: gl.DEPTH_COMPONENT16, gl.STENCIL_INDEX8, gl.DEPTH_STENCIL
- void gl.renderbufferStorageMultisample(gl.RENDERBUFFER, samples, internalFormat, width, height)
- void gl.framebufferRenderbuffer(GLenum target, GLenum attachment, gl.RENDERBUFFER, WebGLRenderbuffer renderbuffer)
  - attachment: gl.COLOR_ATTACHMENT0(~15), gl.DEPTH_ATTACHMENT, gl.DEPTH_STENCIL_ATTACHMENT, gl.STENCIL_ATTACHMENT

Transform Feedback

- void gl.transformFeedbackVaryings(program, [DOMString] varyings, bufferMode)
  - bufferMode: gl.INTERLEAVED_ATTRIBS, gl.SEPARATE_ATTRIBS
- WebGLTransformFeedback gl.createTransformFeedback()
- void gl.bindTransformFeedback(gl.TRANSFORM_FEEDBACK, transformFeedback)
- void gl.deleteTransformFeedback(transformFeedback)
- void gl.beginTransformFeedback(GLenum primitiveMode)
  - primitiveMode: gl.POINTS, gl.LINES, gl.TRIANGLES
- void gl.endTransformFeedback()

Global

- void gl.viewport(x, y, width, height)
- void gl.clear(mask)
  - gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT | gl.STENCIL_BUFFER_BIT
-

OffscreenCanvas

- https://developer.mozilla.org/en-US/docs/Web/API/OffscreenCanvas
-

WebAssembly

- https://developer.mozilla.org/en-US/docs/WebAssembly
- https://webassembly.org/
- https://github.com/webassembly
-

Web Worker

- https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API
-

Debug

- Spector.js
- Babylon.js Inspector

Performance

- Prefer WebGL 2.0 over WebGL 1.0 (3~7% speed-up, less GC)
- Prefer single interleaved VBO over multiple VBOs (GPU cache)
- Use VAO
- Use UBO (one for program, one for individual object)
- Use instanced draw
- Cache WebGL state, attribute & uniform location
- Sort by material (avoid changing program too often)
- Sort opaque object by z (near to far, avoid overdraw)
- View Frustum Culling
- Occlusion Culling
- Shader: prefer #if over if
- Use LOD

Tips

- HandlingContextLost
- Don’t check for null on creation
- Don’t put attributes on WebGL resource objects

Library

- Babylon.js
- Three.js
- PlayCanvas
- OSG.JS
- xeogl Scene.js
- CopperLicht

3D Model Format

- FBX
- Collada
- OBJ
- STL
- DAE
- GLTF (GL Transmission Format)

Online Models

- Sketchfab (OSG.JS)
- Clara.io (Three.js)
- 3D Warehouse
- TurboSquid
- Poly (Three.js)
- Remix 3D (Babylon.js)

Physics Engine

- cannon.js A lightweight 3D physics engine written in JavaScript
- ammo.js Direct port of the Bullet physics engine to JavaScript using Emscripten
- JigLibJS JigLib port for javascript
- Oimo.js It's a full javascript conversion of OimoPhysics
- Physijs Physics plugin for Three.js

3D GIS

- CesiumJS An open-source JavaScript library for world-class 3D globes and maps

- Mapbox GL JS JavaScript library for interactive, customizable vector maps on the web

- LumaGL KeplerGL DeckGL WebGL2 powered geospatial visualization layers

- tangram Real-Time WebGL Maps

- d3-geo-projection Extended geographic projections for d3-geo

Editor

- Blend4Web
- Verge3D
- Simlab Composer
- CopperCube
- nunuStudio
- Koru WebGL authoring and export software

WebXR

- WebXR
- WebXR Device API
- WebXR Samples
- WebVR Spec
- Mozilla Mixed Reality Blog
- A-Frame A web framework for building virtual reality experiences
- Exokit Native VR/AR/XR engine for JavaScript

Future

- WebGPU
  - Apple’s WebKit team proposed
  - Not a direct port of any existing native API
  - Based on concepts in Vulkan, Metal, and Direct3D 12
- W3C GPU for the Web Community Group
- nxt-standalone (ANGLE-like library for Chromium's next-generation Web graphics API prototype)