Intro to WebGL: The 3D Web is Here

Nop Jiarathanakul
Autodesk, Inc.

Learning Objectives

• Learn about WebGL—what it is, how it works, and some of the history
• Discover the powers and capabilities of what WebGL can do
• Take a tour through some of the best state-of-the-art WebGL content today
• Learn about a few tools and options for building your own WebGL content

Description

Until now, rich 3D experience has always been limited to desktop applications and a few high-end mobile games. With WebGL, for the first time we have a truly cross-platform graphics API that unlocks the power of hardware-accelerated 3D rendering right on the web. Join us as we learn about what WebGL is, how it works, and how you can use it. We will also show a showcase of some of the most impressive WebGL demonstrations out there. You wouldn’t believe it's the web! Knowledge of web development is recommended but not required. We will mainly cover high-level concepts, run through some demonstrations, and show small snippets of code.

Your AU Expert

Nop Jiarathanakul loves making things with 3D graphics. He graduated from University of Pennsylvania with a bachelor’s degree and a master’s degree in computer graphics, and spent the early years of his young career in games and animation studios like Electronic Arts, Inc., (EA Games) and DreamWorks. Nop came to Autodesk, Inc., for his passion in WebGL, a technology that enables him to combine both his love for the web and 3D, and bring the powers of desktop graphics to the web. He has been using WebGL since its early days in 2010, and has been keeping up with the community ever since. Nop is now working on the Web Viewer Team as a graphics engineer.
What is WebGL?

WebGL (for Web Graphics Library) is a Javascript API for interacting with the GPU for hardware accelerated rendering within the web browser without the use of plug-ins. It is an open web standard supported by all major browser vendors (Chrome, Firefox, Safari, Internet Explorer, and more.)

With WebGL, browsers expose access to low level graphics API for high performance rendering, which was before only accessible to native applications. The WebGL API closely conforms to OpenGL ES 2.0, another open standard that has been vital to the development of computer graphics since 1992.

Timeline

2009 – Khronos started the WebGL Working Group
2010 – First version of Three.js released
2011 – Chrome and Firefox supported WebGL by default
2011 – Safari and Opera support for WebGL but disabled
2014 – Internet Explorer finally bought into WebGL, after giving up on Silverlight
2013 – Android (Chrome Browser)
2014 – iOS 8
2015 – WebGL became the first truly cross-platform 3D API!

How to Use WebGL?

Many people often think of WebGL as a graphics library that is ready to use for building applications. It is a common misconception. Instead, WebGL is a very low level API. It doesn’t have any higher level concepts of 3D graphics, like transforms, lights, or materials. It also doesn’t have any understanding of file formats, and thus WebGL by itself cannot simply open an OBJ or a COLLADA file. WebGL is simply a thin Javascript wrapper around the native OpenGL driver, and all it does is interface with the hardware, like copying byte buffers and issuing commands.

For this reason, using WebGL is hard! A lot of things have to be done before you can build an application on top of it, which is the reason why many people develop engines, pieces of code that understand concepts of 3D graphics and how to draw them. By far the most popular and long-standing of them all is Three.js. But if you don’t know how to code, there have been a recent proliferation of online viewers and editors that lets you harness the power of WebGL without writing a line of code. These alternatives will be covered in the later section.

There are many topics involved in the field of computer graphics and building 3D applications. WebGL is merely concerned with the implementation aspect. This aspect is called the Graphics Pipeline, which is the series of operations a computer performs to produce images on a screen.
Topics in Computer Graphics

- Linear Algebra
  - The mathematical foundation required for describing objects in 3D
  - Examples: points, lines, vectors, matrices, transforms
- Computer Graphics
  - The theoretical study of producing and manipulating 3D visualizations
  - Main disciplines: Rendering, Animation, Simulation
- Graphics Pipeline
  - The implementation of computer graphics
  - How modern computers draw and produce images
  - This is WebGL

Graphics Pipeline

1. Submit data to the GPU
   - Vertices, transforms, textures, etc
2. Vertex Shader
   - Processes and transforms vertices
3. Triangle Assembly
   - Connect vertices to form triangles
4. Rasterization
   - Breaks triangles into pixel fragments
5. Fragment Shader
   - Outputs color
6. Fragment Tests
   - Test whether a pixel should draw to the frame buffer
7. Blending
   - Output pixels are blended and drawn onto the final frame buffer

Shaders

- A program executing on the GPU for producing rendering effects
- Compiled into machine code for graphics hardware
- Highly parallel: Single Instruction Multiple Data (SIMD)
- Can be used for general computing like physics simulations and computer vision
- Vertex Shaders act on vertices
- Fragment Shaders act on pixels
Coding WebGL

Sample Code: [https://github.com/nopjia/webgl-starter](https://github.com/nopjia/webgl-starter)
Live Demo: [http://nopjia.github.io/webgl-starter](http://nopjia.github.io/webgl-starter)

Basic “Hello World” Steps
1. Create a <canvas> element
2. Initialize WebGL context
3. Setup buffers
4. Setup shaders
5. Setup matrices
6. Draw

Real World Examples

Viewers
Art Station + Marmoset Viewer
- Premier art showcase platform with the most stunning WebGL viewer engine today
- [https://www.artstation.com/artwork?marmoset=true](https://www.artstation.com/artwork?marmoset=true)
- Demo Picks:
  - Ares, by Baj Singh – [https://www.artstation.com/artwork/DAqG0](https://www.artstation.com/artwork/DAqG0)
  - Handgun, by Ethan Hiley – [https://www.artstation.com/artwork/3qK8J](https://www.artstation.com/artwork/3qK8J)
  - Darth Vader, Yosuke Ishikawa – [https://www.artstation.com/embed/1400589](https://www.artstation.com/embed/1400589)

Game Engines
PlayCanvas
- Pure WebGL game engine and editor – [https://playcanvas.com/](https://playcanvas.com/)
- Starlord – [http://playcanv.as/b/FQbBsJTd](http://playcanv.as/b/FQbBsJTd)
- BMW i8 – [http://playcanv.as/p/RqJJ9oU9](http://playcanv.as/p/RqJJ9oU9)

Advertising
Porsche Black Edition

Smart Water
- [http://www.drinksmartwater.com/#/explore-the-sky](http://www.drinksmartwater.com/#/explore-the-sky)

Sennheiser Orpheus

E-Commerce
Penny Skateboards Customizer

Mixamo – marketplace for 3D animations
- [https://www.mixamo.com/store/#/](https://www.mixamo.com/store/#/)
Visualization
Google’s 100,000 Stars
- http://stars.chromeexperiments.com/
Visualizing OpenFlight Data
- http://stewd.io/airborne/

Simulation
WebGL Water, by Evan Wallace – one of the first and still today most well-done WebGL demos
- http://madebyevan.com/webgl-water/
WebGL Fluid, by George Corney
Vortex Spheres, by David Li
- http://david.li/vortexspheres/
Particle Dream, by Nop Jiartathanakul
- http://iamnop.com/particles

Art
Jellyfish – one of the very first WebGL demos
- http://akirodic.com/p/jellyfish/
Matt Deslauriers
- http://mattdesl.svbtle.com/some-javascript-sketches
Yi Wen – Codevember
- http://yiwenl.github.io/Codevember/
Enough, 30-minute interactive picture book by Isaac Cohen
- http://cabbi.bo/enough/

Chrome Experiments
- https://500.chromeexperiments.com/ – Celebrating 500 Chrome Experiments
ShaderToy – GLSL sandbox with impressive demos and active community
- https://www.shadertoy.com/

Interactive Web
Steven Wittens – legendary graphics programmer
- http://acko.net/
Valentin Marmonier
- http://vaalentin.github.io/2015/
Ways to Use WebGL

Viewers

Sketchfab
- Popular, simple, and easy to use 3D Viewer
  - https://sketchfab.com/

Marmoset
- High-end 3D viewer with stunning quality, exported from their desktop renderer
  - http://www.marmoset.co/toolbag/gallery

Clara.io
- Free online 3D modeling and viewing
  - https://clara.io/

Autodesk LMV
- Online 3D viewer for 50+ formats, specialized for large CAD models with hundreds of parts
  - http://lmv.rocks/

Game Engines

PlayCanvas
- A powerful and full-featured game editor and engine, all on the web with WebGL
  - https://playcanvas.com/

Goo Create
- A WebGL editor and engine, targeted for interactive web experiences
  - http://goocreate.com/

Unity and Unreal Engine
- The two most popular game engines support exporting to WebGL
- They leverage Emscripten to cross-compile their engines to Javascript

Presentation

Cl3ver
- Online platform to create interactive 3D virtual product presentations
  - https://www.cl3ver.com/

Verold
- Online visual editor for turning 3D content into interactive presentations
  - https://www.verold.com/

Vizor.io
- Node-based editor for creating VR experiences, based on Three.js
  - http://vizor.io/

Javascript Libraries

Three.js
- The number one most popular and successful open-source WebGL library
  - http://threejs.org/
Babylon.js
- The second most popular open-source WebGL library
- [http://www.babylonjs.com/](http://www.babylonjs.com/)

### Resources

#### Online Lessons

Interactive 3D Graphics
- Free Udacity course on fundamentals of 3D Graphics, using Three.js as a coding tool
- Taught by Eric Haines of Autodesk!

WebGL Fundamentals
- Learn modern WebGL API and concepts, from the ground up
- [http://webglfundamentals.org/](http://webglfundamentals.org/)

Learning WebGL
- An older WebGL tutorial
- [http://learningwebgl.com/blog/?page_id=1217](http://learningwebgl.com/blog/?page_id=1217)

#### Links

Three.js
- Full of examples, useful links, and of course, source code
- [http://threejs.org/](http://threejs.org/)

Codeflow
- Personal blog of veteran graphics programmer Florian Boesch
- Well written with many good resources on WebGL and general 3D graphics
- [http://codeflow.org/tags/webgl.html](http://codeflow.org/tags/webgl.html)

TojiCode
- Personal blog of Chrome developer Brandon Jones
- Latest updates on Chrome, WebGL, and WebVR

WebGL Stats
- Gathers data on global browser WebGL capabilities
- [http://webglstats.com/](http://webglstats.com/)

WebGL Report
- Reports your browser’s WebGL capabilities
- [http://webglreport.com/](http://webglreport.com/)

MSDN WebGL Reference
- By far the best WebGL API reference out there. Way to go Microsoft!