



WebGL

Eduardo Expósito Barrera- alu0101230382@ull.edu.es
Cristo García González - alu0101204512@ull.edu.es



Index

1. Introduction
 - a. Definition of WebGL
 - b. History
 - c. Real applications
 - d. Visualization of real examples
2. Internal operation
 - a. Structure of WebGL Application
 - b. GLSL-Hello World
 - c. Considerations
 - d. Differences between WebGL and OpenGL
 - e. What are Shaders?
3. Code examples
4. Conclusions
 - a. Summary
 - b. Questions

Introduction



- Definition

WebGL is an API implemented in JavaScript for rendering in 3D graphics within any web browser

Keynotes:

- Zero dependencies!!!

- Graphics acceleration

<https://get.webgl.org/>



Definition



Browser	WebGL
Firefox 43+	✓
Chrome 45+ (All Platforms)	✓
Safari 9+	✓
iOS Safari 8.4+	✓
Edge	✓
Internet Explorer 11	✓
Internet Explorer 10	✗
Internet Explorer 9	✗
Internet Explorer 8	✗
Internet Explorer 7	✗

A bit of history

The first steps of WebGL are in the experimental Canvas 3D, in

Mozilla. The first prototype was demonstrated in 2006 and by the end of 2007, Mozilla and Opera has made their own separate implementations.

In early 2009 Khronos Group started the WebGL Working Group.

Version 1.0 was released in March 2011



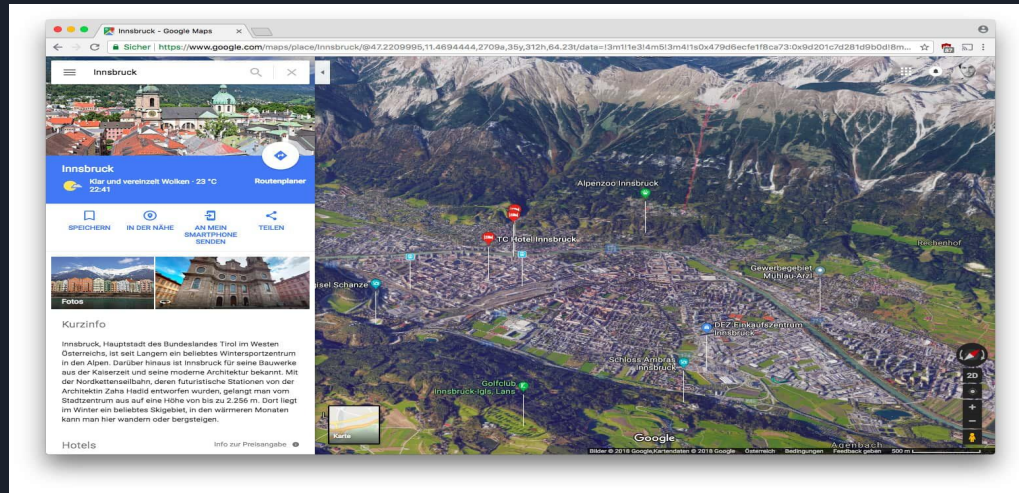
Vladimir
Vukićević

Some example usage



“Experience curiosity”

Some example usage



<http://madebyevan.com/webgl-water/>

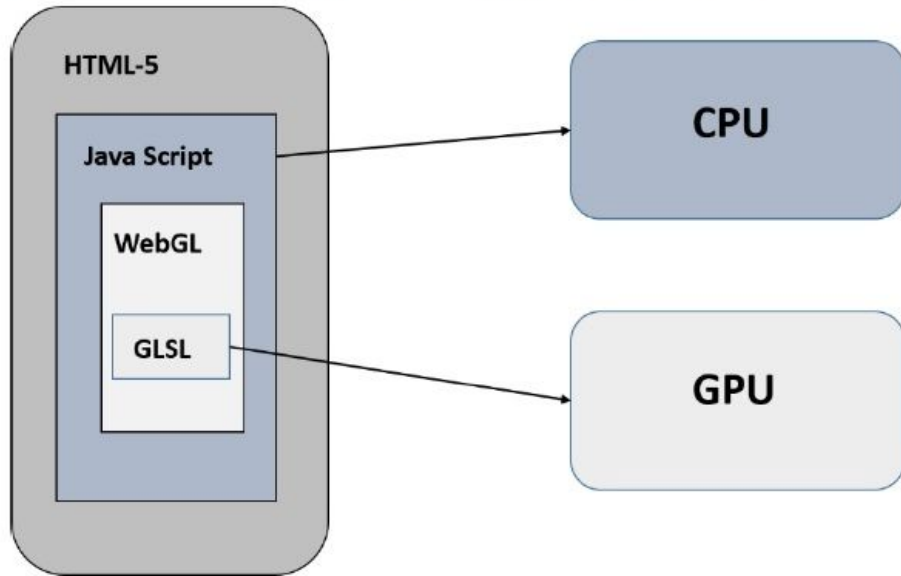
https://alteredqualia.com/three/examples/webgl_cars.html

<http://media.tojicode.com/q3bsp/>

Structure of WebGL Application

WebGL application code is a combination of JavaScript and OpenGL Shader Language.

- JavaScript is required to communicate with the CPU
- OpenGL Shader Language is required to communicate with the GPU.





GLSL-Hello World

We'll come back to this code...

```
#version 300 es

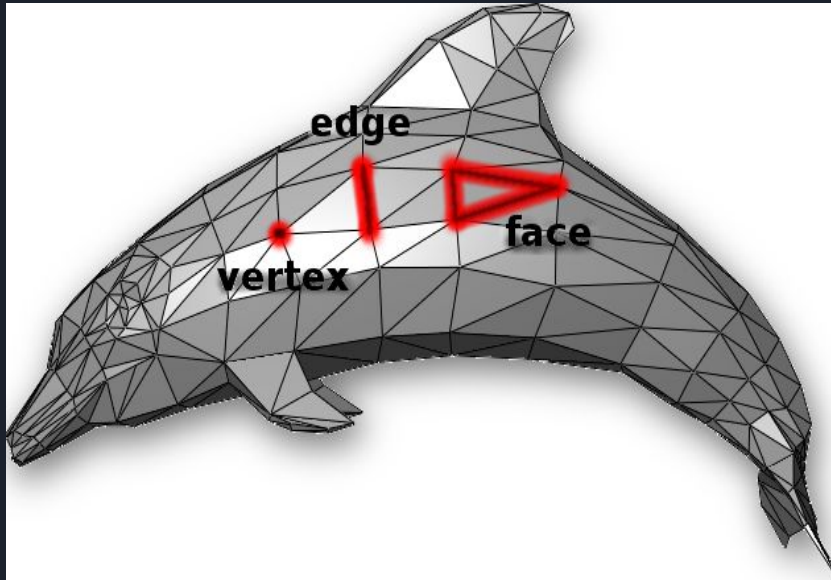
// an attribute is an input (in) to a vertex shader.
// It will receive data from a buffer
in vec4 a_position;

// all shaders have a main function
void main() {

    // gl_Position is a special variable a vertex shader
    // is responsible for setting
    gl_Position = a_position;
}
```

How should I approach WebGL? (Considerations)

- Vertex is the main structure



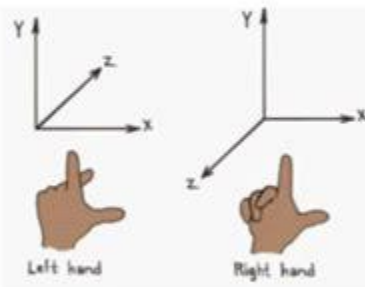
VERTEX	
Coordinates	(3.5, 2, -5)
Color	#ff0539
Normal	(0, 1, 0) ↑

How should I approach WebGL? (Considerations)

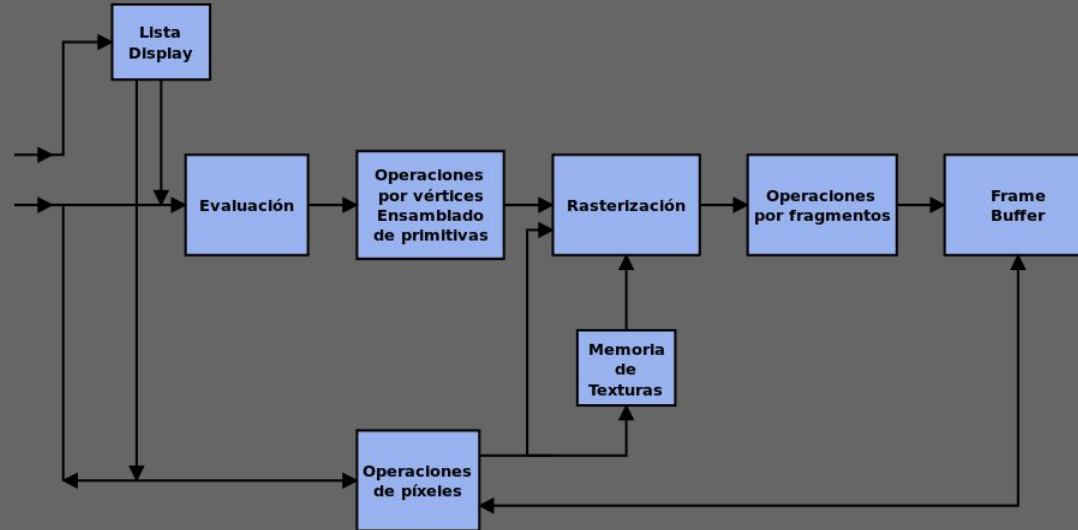
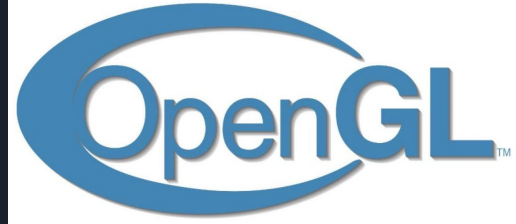
3D Coordinate Systems

- **DirectX**: usually uses Left-handed coordinate system
- **OpenGL**: usually uses Right-handed coordinate system

Left-handed versus Right-handed



OpenGL design and Process in the graphics pipeline





Differences between WebGL and OpenGL

The basis of Comparison	WebGL	OpenGL
Definition	It is designed for rendering 2D and 3D graphics	It is a cross-language and platform API to render 2D and 3D vector graphics
Application	It is mainly used to run in the browser for web applications	It is mainly used in desktop applications
Programmed	It is programmed in JavaScript programming	It is written in C language
Features	It has fewer features comparatively	It has many features to make the application or graphics more interactive
Pipeline	In WebGL, there is no fixed-function pipeline	In OpenGL, there is fixed function pipeline



Differences between WebGL and Canvas

WebGL is low level:

- WebGL is faster than Canvas (also due to GPU rendering).
- WebGL gives you more control in your code.
- Boilerplate

WebGL is newer:

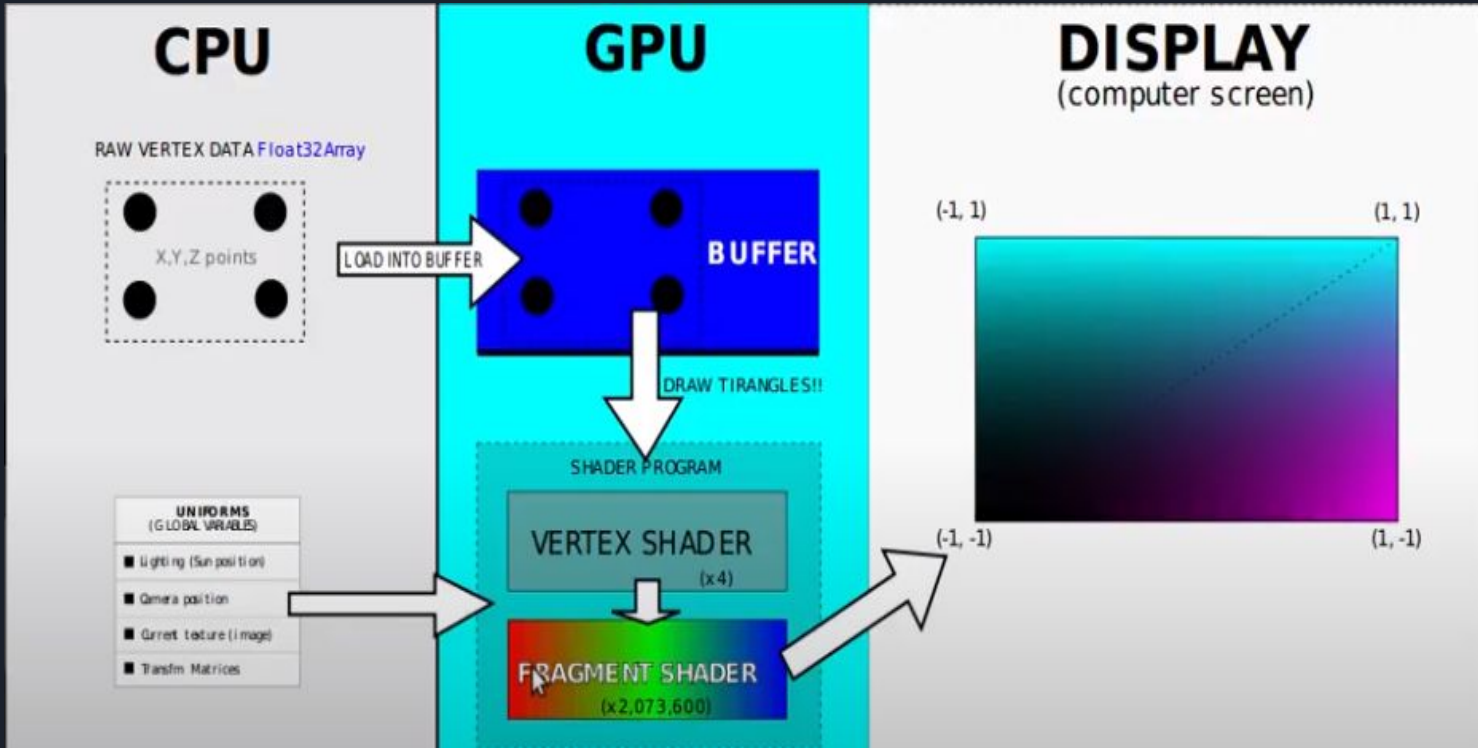
- Not so much used and robust
- Not as much as a standard as Canvas

3D Graphics

Web and mobile devices

Very hard to learn (Unless you have experience with OpenGL)

What are Shaders? How do they work?



Code examples

TO THE CODE!!!!





Summary

- Has no dependency and run in almost all the browsers (including mobile devices)
- Accelerated by Hardware
- It's low level (hard to use, boilerplate, fine grained control)
- It uses 2 languages Javascript for the CPU and GLSL for the GPU
- It's a subset, port, binding, etc... of OpenGL ES which in turn is a subset, port, binding, etc... of OpenGL
- It evolves from canvas, but that's it.
- Take a look at WebGL (<https://en.wikipedia.org/wiki/WebGPU>)



References

[WebGLRenderingContext.drawArrays()]

<https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext/drawArrays>

[WebGL-Shaders]

https://www.tutorialspoint.com/webgl/webgl_shaders.htm

[Structure of WebGL]

https://www.tutorialspoint.com/webgl/webgl_sample_application.htm

[Learn WebGL]

<https://www.youtube.com/watch?v=kju9OgYrUmU>

Gracias por su atención



Eduardo Expósito Barrera

[e-mail](#)



Cristo García González

[e-mail](#)