

# Computação Gráfica

## WebGL

Prof. Gabriel M. Alves

2023-08-10

versão: 0dfd33-dirty

## WebGL

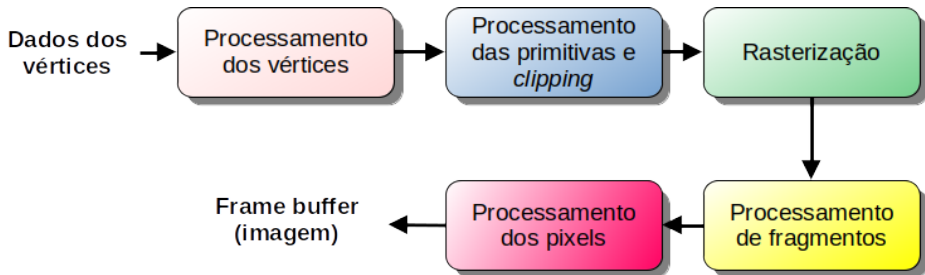
*Web Graphics Library* (WebGL) é uma API JavaScript para renderizar gráficos 2D e 3D interativos em qualquer navegador (*browser*) que seja compatível.

Fonte: <https://en.wikipedia.org/wiki/WebGL>

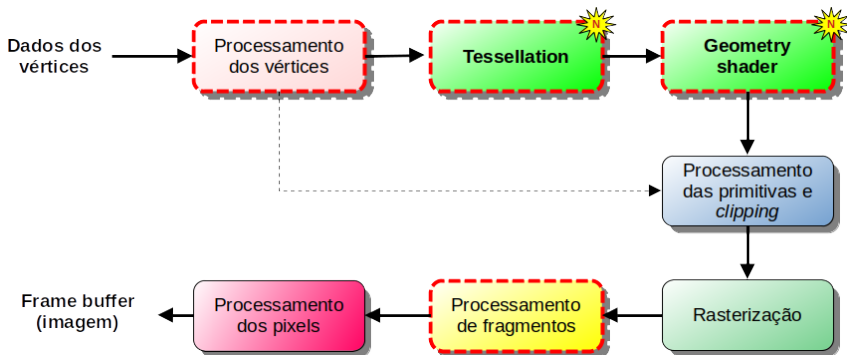


- WebGL é *cross-platform*, uma API 3D de baixo nível baseada na OpenGL ES.
- WebGL utiliza JavaScript como linguagem de programação para controlar a interação com os recursos gráficos.
- O resultado gerado pela WebGL é apresentado no elemento canvas do **HTML5**.
- OpenGL 2.0 introduziu o pipeline programável e manteve a compatibilidade com o *pipeline* fixo.
- WebGL é **obrigatório** o uso do *pipeline* programável.
- Um programa WebGL consiste de duas partes:
  - parte escrito em JavaScript
  - parte escrito em GLSL

- **Relembrando:** Visão simplificada do **pipeline fixo**

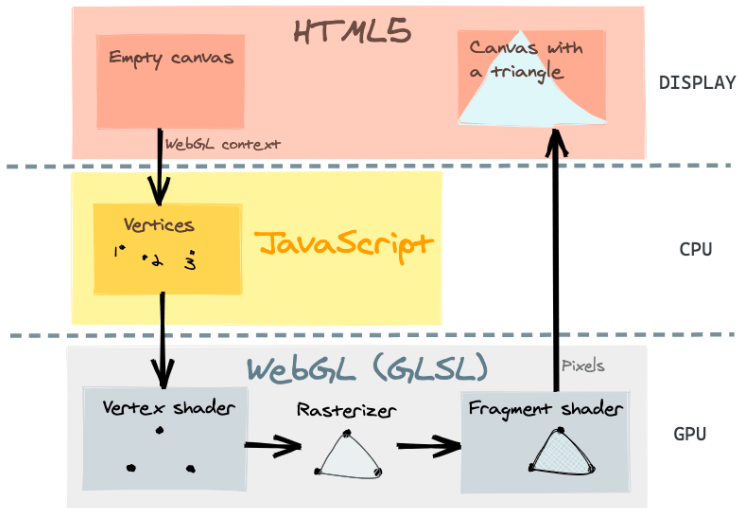


- **Relembrando:** Visão simplificada do **pipeline programável**



- processamento dos vértices (*vertex shader*):
  - é executado uma vez por vértice e processa os dados associados;
  - programador deve especificar as matrizes de transformação e não apenas as transformações como no pipeline fixo.
- processamento de fragmentos (*fragment shader*):
  - é executado uma vez por fragmento e configura ou descarta a cor dele.
  - calcula os valores das cores interpoladas por fragmento.
  - caso contenha textura, o cálculo da cor deve considerá-la.
- WebGL, atualmente, não suporta *tessellation shader* nem *geometry shader*.

# Estrutura de um programa WebGL



Fonte: <https://aralroca.com/blog/first-steps-in-webgl>

# Estrutura de um programa WebGL

- O navegador tem suporte a WebGL? <https://webglreport.com/>
- Inicialmente é necessário um **contexto gráfico WebGL**.
- Esse contexto gráfico é um objeto JavaScript:

---

```
1  const canvas = document.getElementById("glcanvas");
2  // contexto gráfico
3  const gl = canvas.getContext("webgl2");
4
5  if ( !gl ) {
6      alert("Navegador não suporta WebGL.");
7      return;
8  }
```

---



# Estrutura de um programa WebGL

- O contexto gráfico (variável *g*) é associado a um elemento HTML canvas representado pela variável *canvas*.
- Considera-se que há uma página HTML que inclui um elemento canvas:

---

```
1  <!-- marcação html anterior -->
2
3  <canvas width="800" height="640" id="glcanvas"></canvas>
4
5  <!-- restante do html -->
```

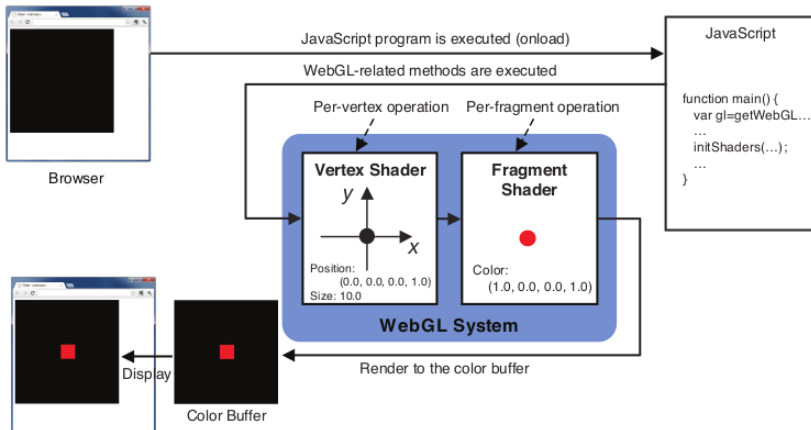
---

# Estrutura de um programa WebGL

- WebGL é executada na placa gráfica do computador em que o navegador está sendo executado.
- Portanto, além do código JavaScript, são necessários dois programas: *vertex shader* e *fragment shader*.
- *Shaders* são escritos na linguagem *OpenGL Shading Language* (GLSL).
- Cada *shader* possui sua própria função `main()`.
- Os dois *shaders* são compilados separadamente e então *linkados* para produzir o programa completo.
- O programa *shader* completo é executado na placa gráfica do computador.

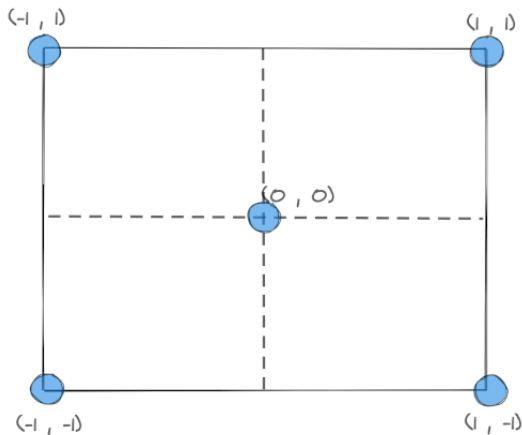
# Exemplo prático

- Desenhar um ponto na tela



# Exemplo prático

- Coordenadas



Fonte: <https://aralroca.com/blog/first-steps-in-webgl>

- WebGL2 Fundamentals. Disponível em:  
<https://webgl2fundamentals.org>
- Ginsburg, D.; Purnomo, B. OpenGL ES 3.0: Programming Guide. Second Edition. Addison-Wesley. 2014.
- Bailey, M.; Cunningham S. Graphics Shaders: Theory and Practice. Second Edition. CRC Press. 2012.
- Dave Shreiner, Graham Sellers, John M. Kessenich, Bill Licea-Kane. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, Version 4.3, Addison-Wesley. 2013.

- Dúvidas?
- Comentários?

## Contato

Gabriel Marcelino Alves  
gabriel.marcelino@ifsp.edu.br



*This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>*

