

# webgl基础学习

---

## 简介

WebGL (Web Graphics Library) 是一个 Web 标准 JavaScript API, 通过 HTML5 的 canvas 元素进行暴露, 无需使用插件, 即可在浏览器中渲染高性能的交互式 3D 和 2D 图形。目前是由非营利 Khronos Group 设计和维护。

WebGL是一种3D绘图协议, 这种绘图技术标准允许把JavaScript和OpenGL ES 2.0结合在一起, 通过增加OpenGL ES 2.0的一个JavaScript绑定, WebGL可以为HTML5 Canvas提供硬件3D加速渲染, 这样Web开发人员就可以借助系统显卡来在浏览器里更流畅地展示3D场景和模型了。

webgl其实就是openGL2.0的网页版本, 里面缩减了一部分的内容。

### 优势:

#### 1. 高性能

WebGL 性能高的原因是它使用到了 GPU。GPU 和 CPU 针对的是两种不同的应用场景, 大家可以把 CPU 想象为一个切图专家, 而 GPU 是一群初级切图仔, 现在有一大堆非常简单的页面, 大街上随便抓个人都能切。那么对于这个任务不用想就知道一群初级切图仔更快, 切图专家当然厉害, 但是也奈不了对面人多。所以对于大量简单计算 GPU 的执行速度是远大于 CPU 的。

### 不足:

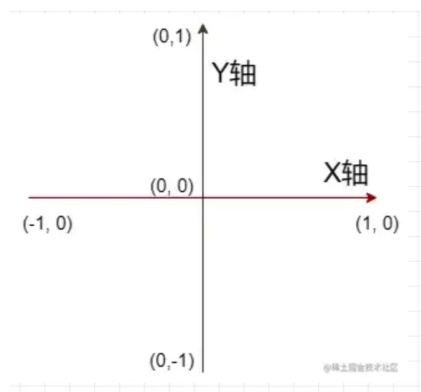
1. 开发成本高**巨高**, 需要掌握一定的数学知识, 尤其是线性代码
2. 硬件要求比较高, 尤其是GPU

## webgl坐标系

webgl的每个方向都有可以使用的值得区间, 超出该区间的图像不会绘制。这些值与canvas的尺寸没有关系, 无论canvas为多大webgl都为  $[-1, +1]$

x轴最左边为-1, 最右边为1

y轴最下边为-1, 最上边为1



## 初识webgl

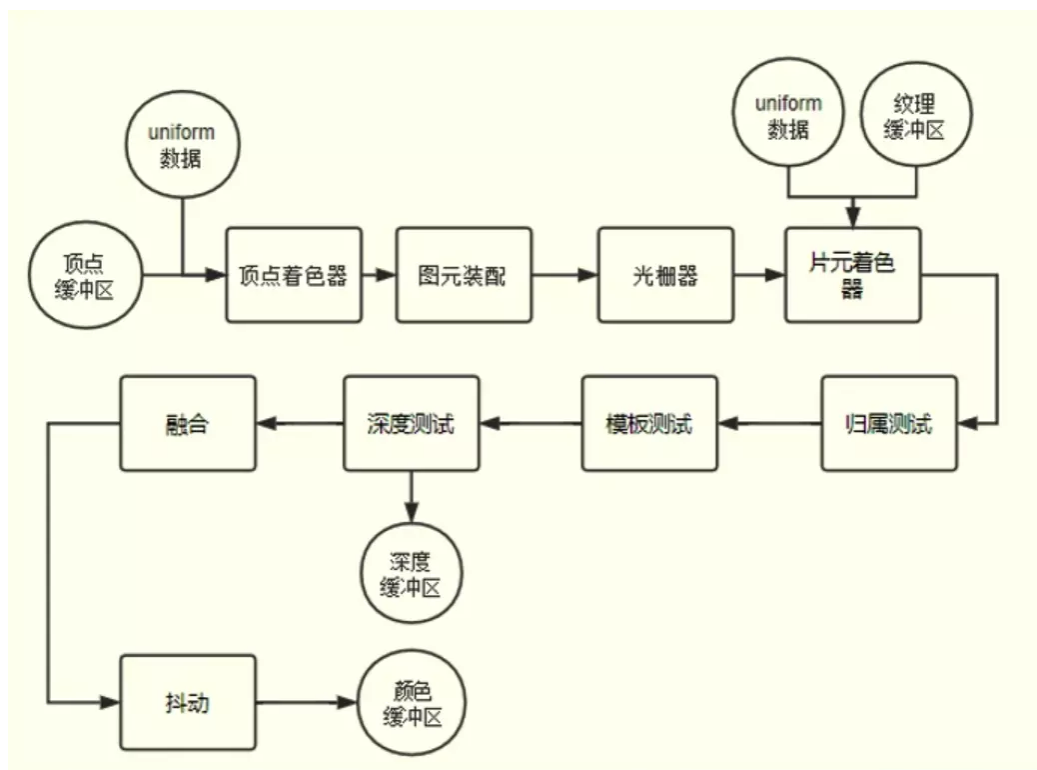
### webgl绘制一个简单的三角形

### 常见的缓存区

1. 顶点缓存区(webgl.ARRAY\_BUFFER)
2. 索引缓存区(webgl.ELEMENT\_ARRAY\_BUFFER)
  - a. 使用drawElements绘制
3. 纹理缓存区
4. 帧缓存
5. 深度缓冲区
6. 颜色缓冲区
7. 模板缓冲区

缓冲区的类型包括 静态的 (webgl.STATIC\_DRAW)、动态的 (webgl.DYNAMIC\_DRAW)、偶尔变(webgl.STATIC\_DRAW)

### 渲染管线



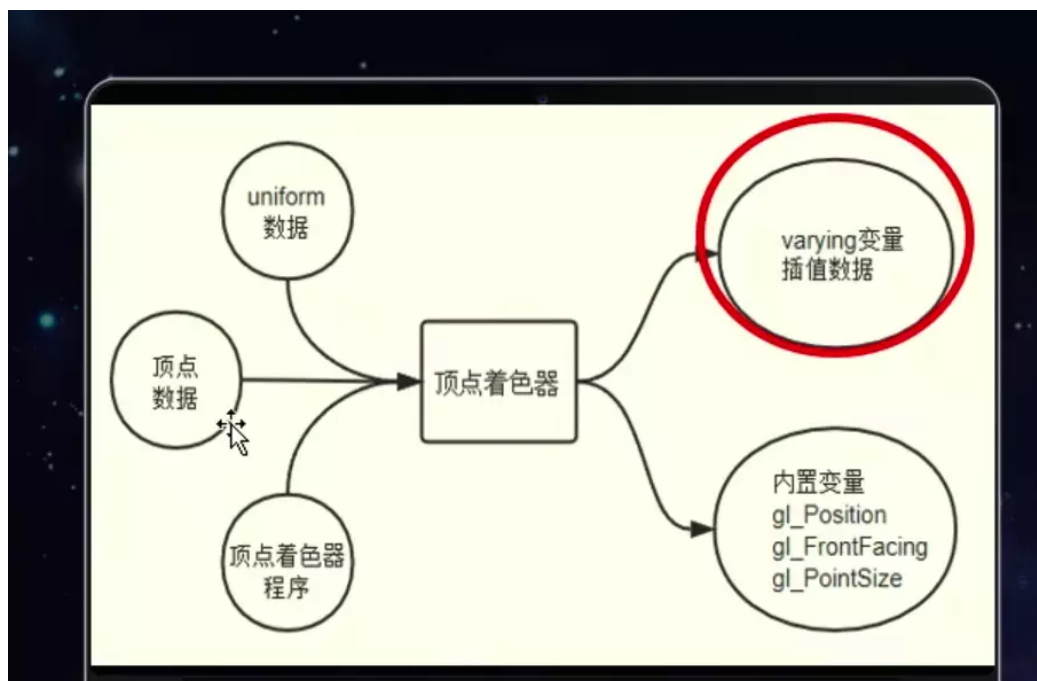
渲染管线就像一条流水线，由一些列具有特定功能的数据电路组成，下一个功能单元处理上一个功能单元生成的数据，逐级处理数据

顶点着色器和片元着色器是可编程的功能单元，拥有更大的自主性，还有光栅器，深度测试等不可编程的功能单元，CPU会通过webgl API和GPU通信，传递着色器程序和数据，GPU执行的着色器可以通过useProgram方法切换，传递数据就是把CPU主存中的数据传送给GPU的显存中

## webgl关键词

### 顶点着色器

顶点着色器是GPU渲染管线上一个可以执行着色器语言的功能单元，具体执行的就是顶点着色器程序webgl顶点着色器程序在js中以字符串的形式存在，通过编译处理后传递给顶点着色器执行。顶点着色器主要是作用就是执行顶点着色器程序对顶点进行变换计算，比如顶点坐标执行进行旋转、平移等矩阵变化。变换后新的顶点坐标然后赋值给内置变量gl\_Position, 作为顶点着色器的输出，图元配置和光栅化环节的输入。

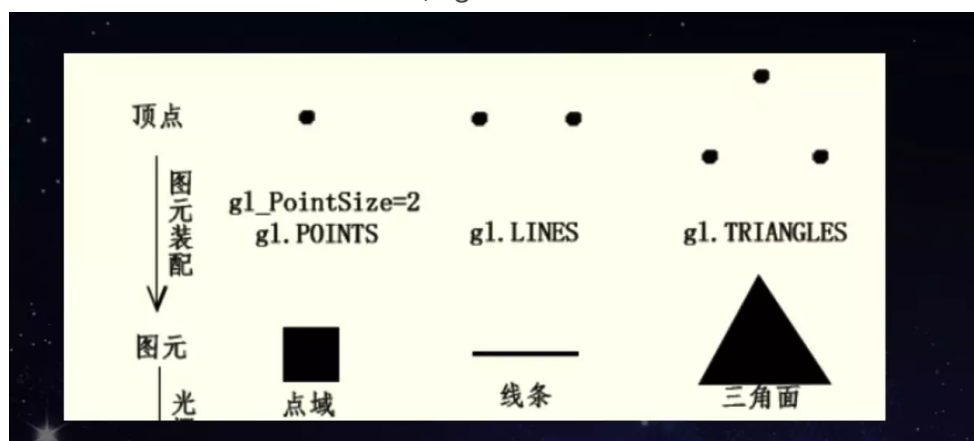


## 存储限定符

- uniform 静态的数据
- attribute 一般是传递的顶点的数据
- varying vs的输出 ps的输入

## 图元装配

顶点变换后的操作是图元配件，就是绘制函数drawArrays() 或 drawElements()第一个参数绘制模式mode控制顶点如何装配为图元，gl.LINES的定义是把两个顶点配置成一个线条图元，gl.TRIANGLES定义的是三个顶点装配为一个三角面图元，gl.POINTS定义的是一个点域图元

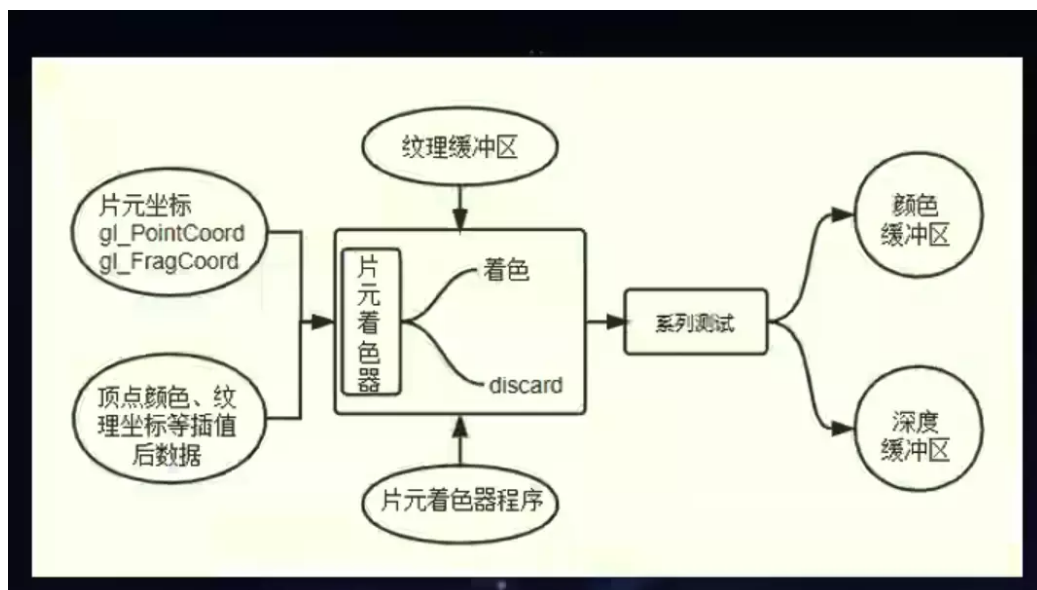


## 光栅化



## 片元着色器

片元着色器和顶点着色器一样是GPU渲染管线上一个可以执行着色器的功能单元，顶点着色器处理的是逐级顶点处理顶点数据，片元着色器是逐片元处理片元数据，通过给内置变量gl\_FragColor复制可以给每一个片元进行着色，值可以是一个确定的GRBA值，可以是一个和片元位置相关的值，也可以是插值后的顶点颜色。除了给片元进行着色之外通过discard还可以实现哪些片元可以被丢弃，被丢弃的片元就不会出现在帧缓存区，自然不会显示在canvas画布上



## 常见的矩阵

1. 模型矩阵
2. 观察矩阵
3. 投影矩阵
4. 视口矩阵

## 投影

投影就是把空间中的物体投射到某个平面的影像，这样就把3D场景转换为为了2D平面图像。

投影的方式一般分为正交投影、透视投影两种，透视投影可以产生近大远小的效果，就和人类观察世界的方式类似；而正交投影是一种平行投影，类似用一束平行光把物体的影像垂直地投射到地面上。

## 纹理

常见的纹理属性名称

1. TEXTURE\_WRAP\_S: 纹理包装，纹理坐标是0->1.x方向超过时渲染模式
2. TEXTURE\_WRAP\_T: 纹理包装，纹理坐标是0->1.y方向超过时渲染模式
3. TEXTURE\_MAG\_FILTER:
4. TEXTURE\_MIN\_FILTER:

常见的纹理属性值

5. NEAREST: 最近点成采样算法，效率好，质量差
6. LINEAR: 线性采样，效率差质量好
7. CLAMP\_TO\_EDGE: 边缘引用，边的最后一个点的颜色
8. REPEAT: 重复平铺
9. MIRRORED\_REPEAT: 上下，左右均翻转重复平铺

## 参考

api文档: <https://registry.khronos.org/webgl/specs/latest/2.0/>

优秀文章: <https://juejin.cn/post/7146385487039725582>

矩阵框架概念: <https://zhuanlan.zhihu.com/p/473031788>

矩阵框架库: <https://cdn.jsdelivr.net/npm/gl-matrix@3.0.0/gl-matrix.js>