## WebGL入门

1.清空绘图区

清空绘图区是使用指定的背景颜色填充canvas，使用gl.clearColor设置背景色。gl.clearColor(red, green, blue, alpha)。openGL的颜色取值返回是0-1。

调用gl.clear()函数，用clearColor指定的背景色清空绘图区域。gl.clear(g.COLOR\_BUFFRE\_BIT)，清理绘图区域实际上在清理颜色缓冲区（color buffer），传递的gl.COLOR\_BUFFER\_BIT就是在告诉WebGL清理颜色缓冲区。

缓冲区还包括：  
gl.COLOR\_BUFFER\_BIT颜色缓冲区、gl.DEPTH\_BUFFER\_BIT深度缓冲区、gl.STENCIL\_BUFFER\_BIT模板缓冲区。清理函数分别为gl.clearColor(red,green,blue,alpha)、gl.clearDepth(depth)、gl.clearStencil(s)。

2.什么是着色器

WebGL有两种着色器：

顶点着色器(Vertex shader)：顶点着色器是用来描述顶点特性（如位置、尺寸等）的程序。顶点(vertext)是指二维或三维空间中的一个点，比如二维或三维图形的端点或交点。  
  片元着色器(Fragment shader)：进行片元处理过程如光照的程序。片元（fragment）是一个WebGL术语，你可以将其理解为像素（图像的单元）。

3.GLSE中的数据类型

 float:表示浮点数

 vec4:表示有四个浮点数组成的矢量

 mat4:4\*4矩阵（WebGL中矩阵是列主序的）

 必须注意的是，如果WebGL需要的参数是浮点类型，例如10.0。如果传递10会报错，因为10被认为是整数。

4.vec4函数

 我们在使用WebGL时，会给顶点着色器参数赋值，gl\_Position的数据类型为vec4，gl\_Position = vec4(0.0, 0.0, 0.0, 1.0)。但实际需要的位置坐标只有三个(x,y,z)值。幸好WebGL提供了vec4函数。由4个分量组成的矢量被称为齐次坐标，他能够提高三维数据的效率，在三维图形系统大量使用。如果最后一个分量设置为1.0，那么齐次坐标可以表示前三个分量为坐标值的那个点。所有当需要用齐次坐标表示顶点时，只需要将最后一个分量设置为1.0就可以了。

5.gl.drawArrays(mode, first, count)函数

 mode:指定绘制的方式，包括gl.POINTS、gl.LINES、gl.LINE\_STRIP、gl.TRIANGLES、gl.TRIANGLE\_STRIP、gl.TRIANGLE\_FAN。

 first:指定从哪个顶点开始绘制（整形数）  
   count:指定绘制需要用到多少个顶点（整形数）

6.attribute变量

 想要将位置信息从javascript程序中传给顶点着色器。有两种方式可做到：attribute变量和uniform变量。attribute变量传输时那些与顶点相关的数据，uniform变量传输的是哪些所有顶点相同(与顶点无关)的数据。例如attribute vec4 a\_Position, attribute变量a\_Position的类型为vec4。一般attribute变量都以a\_开头，而uniform变量以u\_开头。如何获取attribute变量的存储位置？  
   var a\_Position = gl.getAttribLocation(gl.program, 'a\_Position');  
    如何给顶点位置设置值？  
    gl.vertexAttrib3f(a\_Position, 0.0, 0.0, 0.0);

7.gl.getAttribLocation(program, name)函数

    program:指定包含顶点着色器和片元着色器的着色器程序对象  
    name:想要获取存储地址的attribute变量的名称  
    返回值：大于等于0，attribute变量的存储地址；小于0，指定的attribute变量不存在

8.gl.vertexAttrib4f(location, v0, v1, v2, v3)函数

  location:指定将要修改的attribute变量的存储位置  
    v0:attribute变量的第一个分量的值  
    v1:attribute变量的第二个分量的值  
    v2:attribute变量的第三个分量的值  
    v2:attribute变量的第四个分量的值  
    说明：gl.vertexAttrib3f有几个同族函数。gl.vertexAttrib1f(location, v0),gl.vertexAttrib2f(location, v0, v1),gl.vertexAttrib3f(location, v0, v1, v2)。

9.gl.getUniformLocation(program, name)

program:指定包含顶点着色器和片元着色器的着色器程序对象  
name:想要获取存储地址的uniform变量的名称

返回值：not-null，指定的uniform变量的位置；null，指定的uniform变量不存在，或者其命名以gl\_或者webgl\_前缀。

说明：getUniformLocation和getAttribLocation的区别在于为null和-1，如果变量不存在时。

10.gl.uniform4f(location, v0, v1, v2, v3)

向location位置处的变量赋值，参数和vertexAttrib4f函数参数相似。

11.gl.uniform4fv(location, [v0, v1, v2, v3])

向location位置处的变量赋值，参数是以数组的形式传入。

11.gl.uniformMatrix4fv(location, transpose, array)

将array表示的4\*4矩阵分配给由location指定的uniform变量。

参数：

    location:uniform变量的存储位置。

    Transpose：在WebGL中必须指定为false

    array:带传输的类型化数组，4\*4矩阵按列主序存储在其中

12.gl.FragCoord

该内置变量的第一个分量和第二个分量表示片元在<canvas>坐标系统中的坐标值

13.gl.drawingBufferWidth/gl.drawingBufferHeight

分别表示颜色缓冲区的宽度和高度

## WebGL绘制与变换

1.使用缓冲区对象向顶点着色器传入多个顶点的数据，需要遵循以下五个步骤：

    1.1 创建缓冲区对象（gl.createBuffer()）。

    1.2 绑定缓冲区对象（gl.bindBuffer()）。

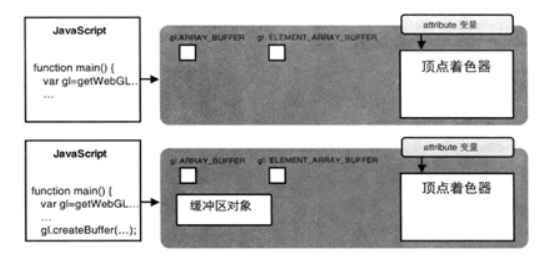
    1.3 将数据写入缓冲区对象（gl.bufferData()）。

    1.4 将缓冲区对象分配给一个attribute变量（gl.vertexAttribPointer()）。

1.5 开启attribute变量（gl.enableVertexAttribArray()）。

2.创建缓冲区对象（gl.createBuffer()）

    在使用WebGL时，需要调用gl.createBuffer()方法来创建缓冲区对象。下面的图上部分是执行前的状态，下部分是执行后的状态。



 对应createBuffer的是gl.deleteBuffer(buffer)函数，用来删除创建的缓冲区对象。buffer表示带删除的缓冲区对象。

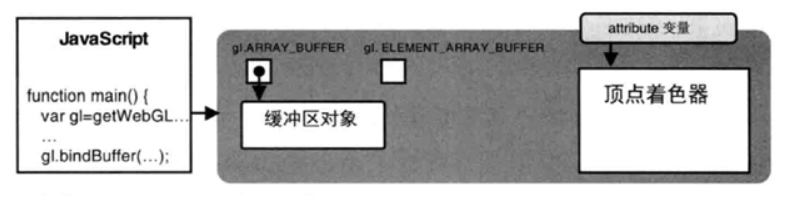
3.绑定缓冲区（gl.bindBuffer(target, buffer)）

    将缓冲区对象绑定到WebGl系统中已经存在的“目标”（target）上。参数：

    target:绑定的目标。可以使以下中的一个：gl.ARRAY\_BUFFER,表示缓冲区对象中包含了顶点的数据;gl.ELEMENT\_,表示缓冲区对象中包含了顶点的索引值。

    buffer：指定之前由gl.createBuffer()返回的带绑定的缓冲区对象。

    执行绑定后，WebGL系统内部状态发生变化。如图所示：



4.向缓冲区对象中写入数据（gl.bufferData(target, data, usage)）

开辟存储空间，想绑定在target上的缓冲区对象总写入数据data。

参数：

    target:gl.ARRAY\_BUFFER或gl.ELEMENT\_ARRAY\_BUFFER。

    data:写入缓冲区对象的数据。

usage: 表示程序将如何使用存储在缓冲区对象中的数据。参数值包括：gl.STATIC\_DRAW,只会向缓冲区对象中写入一次数据，但需要绘制很多次；gl.STREAM\_DRAW,只会向缓冲区对象中写入一次数据，然后绘制若干次;gl.DYNAMIC\_DRAW,会想缓冲区对象中多次写入数据，并绘制很多次。

5.类型化数组

WebGL使用的各种类型化数组包括：

    数组类型/每个元素所占用字节数/描述（C语言总的数据类型）

    Int8Array/1/8位整形数(signed char)

    UInt8Array/1/8位无符号整形数(unsigned char)

    Int16/2/16位整形数（signed short）

    UInt16Array/2/16位无符号整形数(unsigned short)

    Int32Array/4/32位整形数（signed int）

    UInt32Array/4/32位无符号整形数(unsigned int)

    Float32Array/4/单精度32位浮点数（float）

Float64Array/8/双精度64位浮点数（double）

6.类型化数组的方法、属性和常量

方法、属性和常量/描述

    get(index)/获取第index个元素值

    set(index, offset)/设置第index个元素的值为value

    set(array, offset)/从第offset个元素开始讲数组array中的值填充进去

    length/数组长度

BYTES\_PER\_ELEMENT/数组中每个元素所占字节数

7.将缓冲区对象分配给attribute变量（gl.vertexAttribPointer(location, size, type, normalized, stride, offset)）

   将绑定到gl.ARRAY\_BUFFER的缓冲区对象分配给有location指定的attribute变量。参数：

    location:指定带分配attribute变量的存储位置。

    size:指定缓冲区中每个顶点的分量个数（1到4）。若size比attribute变量书序的分量数小，确实分量将按照与gl.vertexAttrib[1234]f()相同的规则补全。

    type:使用一下类型之一来指定数据格式：gl.UNSIGNED\_BYTE,无符号字节，UInt8Array;gl.SHORT，短整形，Int16Array;gl.UNSIGNED\_SHORT,无符号短整形，Uint16Array;gl.INT,整形，Int32Array;gl.UNSIGNED\_INT, 无符号整形，Uint32Array;gl.FLOAT,浮点型，Float32Array。

    normalize:传入true或false，标明是否将非浮点型的数据归纳化到[0,1]或[-1,1]区间。

    stride:指定相邻两个顶点间的字节数，默认为0。

offset：指定缓冲区对象中的偏移量以字节为单位。如果是起始位置，则offset为0。

8.开启attribute变量（gl.enableVertexAttribArray(location)）

为了使定点着色器能够访问缓冲区内的数据，需要使用gl.enableVertexAttribArray()方法开启attribute变量。

9.将矩阵传递给对象缓冲区（gl.uniformMatrix4fv(location, transpore, array)

  将array表示的4\*4矩阵分配给有location指定的uniform变量。

参数：

    location:uniform变量的存储位置

    Transpose:在WebGL中必须指定为false

    array:带传输的类型化数组，4\*4矩阵按例主序存储在其中

获取渲染上下文

创建着色器（创建着色器对象🡪提供数据源🡪编译后生成着色器）

将着色器链接到一个program（着色程序）

从program中找到a\_position属性值所在位置

属性值需从缓冲中获取数据，故创建一个缓冲（gl.createBuffer()）

绑定位置信息缓冲(gl.bindBUffer())

通过绑定点向缓冲中存放数据（gl.bufferData()）