郭隆邦 2023-10-17

2. 着色器GLSL ES语言

学习原生WebGL，除了前面说的JavaScript语言之外，你还需要学习一门新的语言就是着色器语言GLSL ES。

平时你接触的JavaScript、C、java等语言是在CPU上执行，对于着色器语言GLSL ES是在显卡GPU上执行。

如何学习

着色器语言GLSL ES语法虽然类似Typescript、C等语言，但是GLSL主要在GPU上执行，有自身的特殊性，只有结合WebGL案例学习GLSL ES语法，才能更好的理解，所以这节课只介绍GLSL ES部分语法。

所以你本节课不用写任何代码，只要跟着视频过一遍即可，后面还会结合WebGL案例具体介绍GLSL ES的语法。

GLSL ES基础数据类型

着色器语言GLSL的基本数据类型和C语言一样具有常见的整型数`int`、浮点数`float`和布尔值`bool`类型数据。

关键字	数据类型	值
bool	布尔值	布尔变量值为true或false
int	整型数	值为整数，比如0,1,2,3...
float	单精度浮点数	浮点数用小数点表示，比如0.6,3.14,2.8

这三个关键字的用法，下面就会给大家展示怎么用于声明变量。

声明变量

声明变量，并赋值

```js
// 整型变量
int count = 10;
// 浮点数变量
float num = 10.0;
// 声明一个布尔值变量
bool lightBool = true;
```

通过上面变量声明，你也可以看出，着色器语言声明变量和TypeScript一样需要注明数据类型，但是JavaScript不用注明变量数据类型。

```js
// JavaScript语言声明变量，不用设置数据类型
let count = 10;
```

先声明变量，后赋值或改变。

```js
float c;
c = 100.0;
```

改变变量的值

```js
float count = 10.0;
count = 20.0;
```

注意: 注意变量的数据类型和值要对应

错误赋值方式

```js
float num = 1;
```

正确赋值方式

```js
float num = 1.0;
```

变量简单运算

```js
// 32位浮点数相加
float a = 2.0;
float b = 4.0;
float c = a+b;
```

```js
//整数相加
int a = 2;
int b = 4;
int c = a+b;
```

两个变量进行运算，需要保持一样数据类型，否则报错。

声明一个常量 `const`

着色器语言和C语言、javascript语言一样可以通过关键字 `const` 声明一个常量。

```js
// 着色器语言定义一个整形常量
const int count = 10;
// 定义一个浮点数常量10.0
const float count = 10.0;
```

着色器语言和其它语言一样，声明一个变量，可以重新赋值，如果通过关键字const声明一个常量，顾名思义是常量，在代码中是不可以更改的。

```js
const int count = 10;
// 错误写法
count= 20;
```

着色器语言GLSL ES声明函数

函数计算后,如果需要返回的值，通过关键字 `return` 返回，不过注意声明函数时候，函数名称前需要声明 `return` 返回值的数据类型。

```js
// 两个参数是浮点数，相加后返回值自然也是浮点数
float add(float x,float y){
  return x + y
```

```
}
```

声明一个无返回值函数,函数前面用`void`关键字即可。

```js
void main(){
  float x = 10.0;
}
```

if语句

着色器语言GLSL中关于if语句、for语句的使用，和javascript语言、C语言中的if语句、for语句执行逻辑规则基本一致，这里默认你已经有一定的编程基础，也就不做过多讲解，只是简单说明一下。

单独使用`if`

```js
float x = 10.0
if(x > 100.0){
  x = 100.0;
}
```

`if-else`形式

```js
float x = 10.0
if(x > 100.0){
  x = 100.0;
} else {
  x = x + 1.0;
}
```

`if-else if-else if-...else`形式

```js
float x = 10.0
if(x<10.0){

}else if (x>=10.0 && x<20.0) {

}else if (x>=20.0 && x<30.0) {
```

```
}else {

}
```

for循环语句

和你平时写JavaScript的for循环语句基本相似，只是注意注明变量i的数据类型即可。

```js
for (int i = 0; i < 20; i++) {
  ...
}
```

`continue` 和 `break` 关键字

着色器语言 `continue` 和 `break` 关键字和JavaScript语言习惯也是相似的。

`break` 表示终止for循环执行

```js
for (int i = 0; i < 20; i++) {
  ...
  if(i==15){
    break;//直接终止循环执行，i=16、17等后面的循环不再执行
  }
  ...
}
```

`continue` 表示直接跳到for循环的下一个循环

```js
for (int i = 0; i < 20; i++) {
  ...
  if(i==15){
    continue;//进行下次循环，执行i=16对应的循环
  }
  ...
}
```

向量表示颜色

在GLSL ES中，向量可以表示多种数据，也能进行多种数学运算，咱们这里先不讲解那么多，说些简单的。

`vec3`、`vec4`关键字和`int`、`float`一样也是用来表示数据的类型，`vec3`表示三维向量、`vec4`表示四维向量，`vec3`和`vec4`的每个分量都是浮点数`float`。

```js
// 四维向量有四个分量，可以用来表示颜色的R、G、B、A
vec4 color = vec4(1.0, 0.0, 0.0, 1.0);//红色不透明
```

关键字	数据类型
vec2	二维向量，具有xy两个分量，分量是浮点数
vec3	三维向量，具有xyz三个分量，分量是浮点数
vec4	四维向量，具有xyzw四个分量，分量是浮点数
ivec2	二维向量，分量是整型数
ivec3	三维向量，分量是整型数
ivec4	四维向量，分量是整型数
bvec2	二维向量，分量是布尔值bool
bvec3	三维向量，分量是布尔值bool
bvec4	四维向量，分量是布尔值bool

向量表示顶点位置坐标

三维向量`vec3`表示变量pos具有三个分量，可以用来表示顶点的xyz坐标。

```js
vec3 pos = vec3(1.0, 2.0, 3.0);
```

用四维向量`vec4`表示齐次坐标，所谓齐次坐标，就是在GLSL ES中表示一个顶点坐标的的时候，增加一个分量，1.0表示。

```js
vec4 pos = vec4(1.0, 2.0, 3.0,1.0);
```

一个三维向量转化为四维向量

```js
vec3 pos = vec3(1.0, 2.0, 3.0);
vec4 newPos = vec4(pos,1.0);
```

一个二维向量转化为四维向量

```js
vec2 pos = vec2(1.0, 2.0);
vec4 newPos = vec4(pos, 3.0,1.0);
```

内置变量

不管是JavaScript语言，还是着色器语言GLSL ES，你想使用一个变量，都需要先声明。

```js
float a = 2.0;
float b = 4.0;
float c = a+b;
```

所谓**内置变量**就是着色器语言GLSL ES默认提供的变量，不需要声明，就可以使用。GLSL ES内置变量很多，下面介绍几个下节课会用到的。

- `gl_PointSize`：点渲染像素大小，数据类型浮点数`float`
- `gl_Position`：顶点坐标，数据类型四维向量`vec4`
- `gl_FragColor`：像素颜色，数据类型四维向量`vec4`

```js
// 赋值浮点数
gl_PointSize = 20.0;
```

vec4前面三个参数表示xyz坐标，第四个参数按照GLSL ES语法习惯需要设置为1.0

```js
// 赋值四维向量，表示xyz坐标是原点
gl_Position = vec4(0.0,0.0,0.0,1.0);
```

vec4前面三个参数是颜色RGB值，第四个参数是透明度值

```js
// 赋值四维向量，表示红色不透明
gl_FragColor = vec4(1.0,0.0,0.0,1.0);
```

GLSL ES代码注释

GLSL ES代码注释和JavaScript语言的习惯一样。

- 单行注释符号 `//`
- 快级注释符号 `/* */`

GLSL ES语句结尾分号

在JavaScript中，代码语句结尾的分号可以省略，但是 GLSL ES中分号不能省略。

```js
float a = 2.0;//正常
```

分号省略，会报错

```js
float a = 2.0//分号省略，会报错
```

← 1. WebGL学前说明

3. 第一个WebGL案例→