郭隆邦 2023-05-20

11. WebGPU顶点数据插值计算

本节课给大家介绍下，在WebGPU比较常用的一个知识点，就是顶点数据的插值计算。

结构体方式设置顶点着色器`main()`返回值

先学习一个与顶点着色器返回值相关的新语法。

前面课程常用的顶点着色器代码，返回值设置方式`@builtin(position) vec4<f32>`。

```js
@vertex
fn main(@location(0) pos: vec3<f32>) -> @builtin(position) vec4<f32> {
    return vec4<f32>(pos,1.0);
}
```

顶点着色器main函数返回值用结构体的方式设置。

执行`var out:Out;`通过结构体Out定义一个变量out，类似JavaScript语言中的类，通过new实例化一个对象。

```js
fn main(@location(0) pos: vec3<f32>) -> Out {
    var out:Out;//通过结构体生成一个变量
}
```

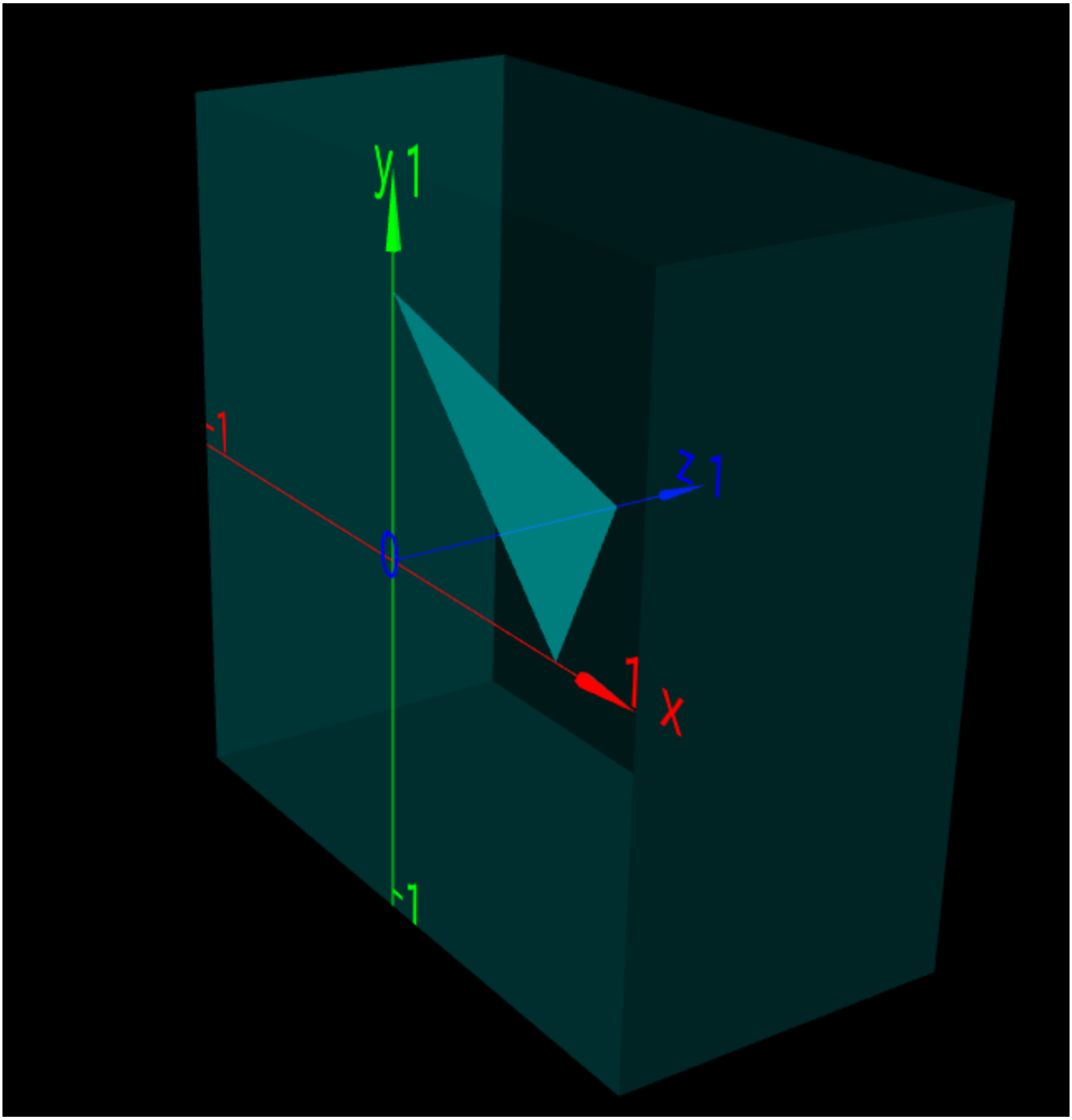
```js
struct Out{
    @builtin(position) position:vec4<f32>,
}
@vertex
fn main(@location(0) pos: vec3<f32>) -> Out {
    var out:Out;
    out.position = vec4<f32>(pos,1.0);
    return out;
}
```

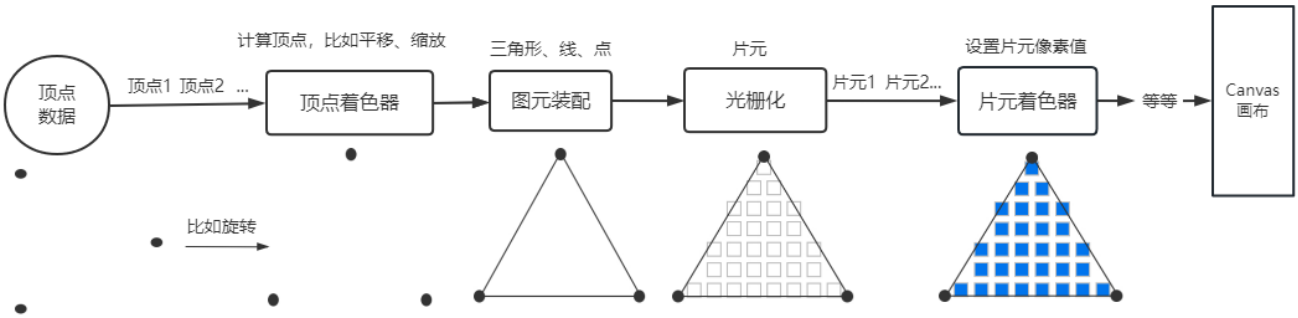
顶点位置数据插值计算

为了方便理解**顶点数据的插值计算**，以片元的坐标计算为例说明，具体就是获取WebGPU 3D标准设备坐标系下，每个片元对应的xyz坐标。

```js
// 在xyz轴上分别取一个点构成构成一个三角形
1.0, 0.0, 0.0,
0.0, 1.0, 0.0,
0.0, 0.0, 1.0,
```

在WebGPU 3D坐标系下，每个片元都有一个xyz坐标值。





1. 声明一个变量表示片元xyz坐标

顶点着色器返回的数据结构Out中，声明一个变量`vPosition:vec3<f32>`表示顶点的xyz坐标，然后使用`@location(0)`标记，注意这里的`@location(0)`与main函数参数中的`@location(0)`不是一回事，也不冲突。一般需要顶点着色器输出的变量，需要使用@location()标记，location的参数可以是0、1、2等，本节课比较简单，只有一个变量，标记为0即可。

```js
struct Out{
    @builtin(position) position:vec4<f32>,
    // 位置变量vPosition表示顶点位置坐标插值后的坐标
    // 通过location标记改变量，location的参数可以是0、1、2等
    // vPosition可以用来表示每个片元的坐标xyz
    @location(0) vPosition:vec3<f32>
}
@vertex
fn main(@location(0) pos: vec3<f32>) -> Out {
    ...
}
```

表示顶点位置数据的变量`pos`赋值给`out.vPosition`，这样顶点着色器功能单元默认就会对顶点进行插值计算，所谓插值计算，就是给pos对应的顶点数据，插入更多的顶点坐标，与片元一一对应，表示每个片元的xyz坐标值。

```js
fn main(@location(0) pos: vec3<f32>) -> Out {
    var out:Out;//通过结构体生成一个变量
    out.position = vec4<f32>(pos,1.0);
    out.vPosition = pos;//插值计算，生成的每个片元对应的xyz坐标
    return out;
}
```

2. 片元着色器获取插值后的片元坐标xyz

main参数通过`@location(0)`声明一个变量，和顶点着色器中vPosition变量关联起来。关联起来的原因就是，两个变量都是通过`@location(0)`标记的，`@location()`的参数都是0。

`vPosition.x`表示片元的x坐标，通过课程代码中三角形三个顶点的坐标可以判断`vPosition.x`的范围是0~1之间。

```js
@fragment
fn main(@location(0) vPosition:vec3<f32>) -> @location(0) vec4<f32> {
    // 根据x坐标设置片元颜色
```

```
    if(vPosition.x<0.5){
        return vec4<f32>(1.0, 0.0, 0.0, 1.0);
    }else{
        return vec4<f32>(0.0, 1.0, 0.0, 1.0);
    }
}
```

← 10. 片元深度值、深度缓冲区

12. 练习—顶点位置插值→