郭隆邦 2023-04-22

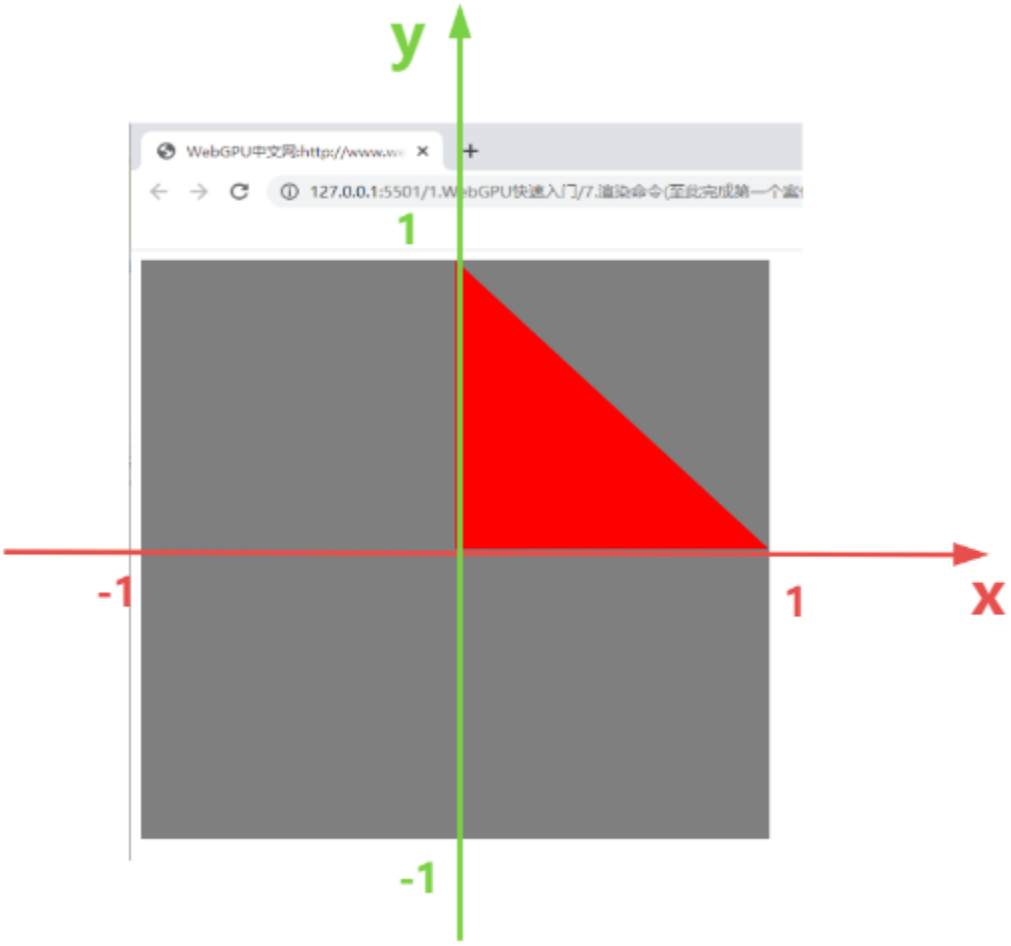
8. WebGPU 3D坐标系(投影)

经过前面7节课的讲解，完成一个最简单的WebGPU三角形小案例，本节课就在前面基础上，给大家讲解WebGPU的**3D坐标系**和**投影**。

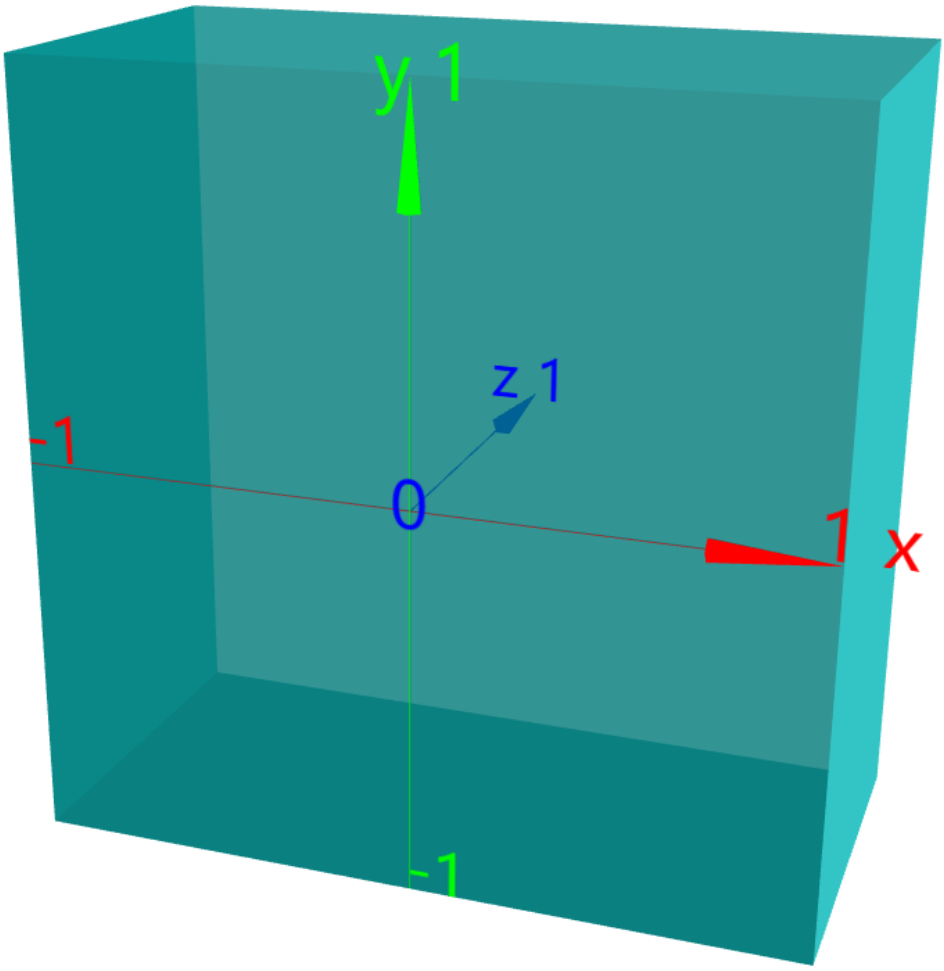
WebGPU标准设备坐标系

在1.3小节，创建顶点缓冲区的时候，简单介绍过WebGPU坐标系知识，咱们先回顾下。

WebGPU坐标系在Canvas画布上的**坐标原点**是Canvas画布的中间位置，**x轴**水平向**右**，**y轴**竖直向**上**，x和y的坐标范围都是[-1,1]，



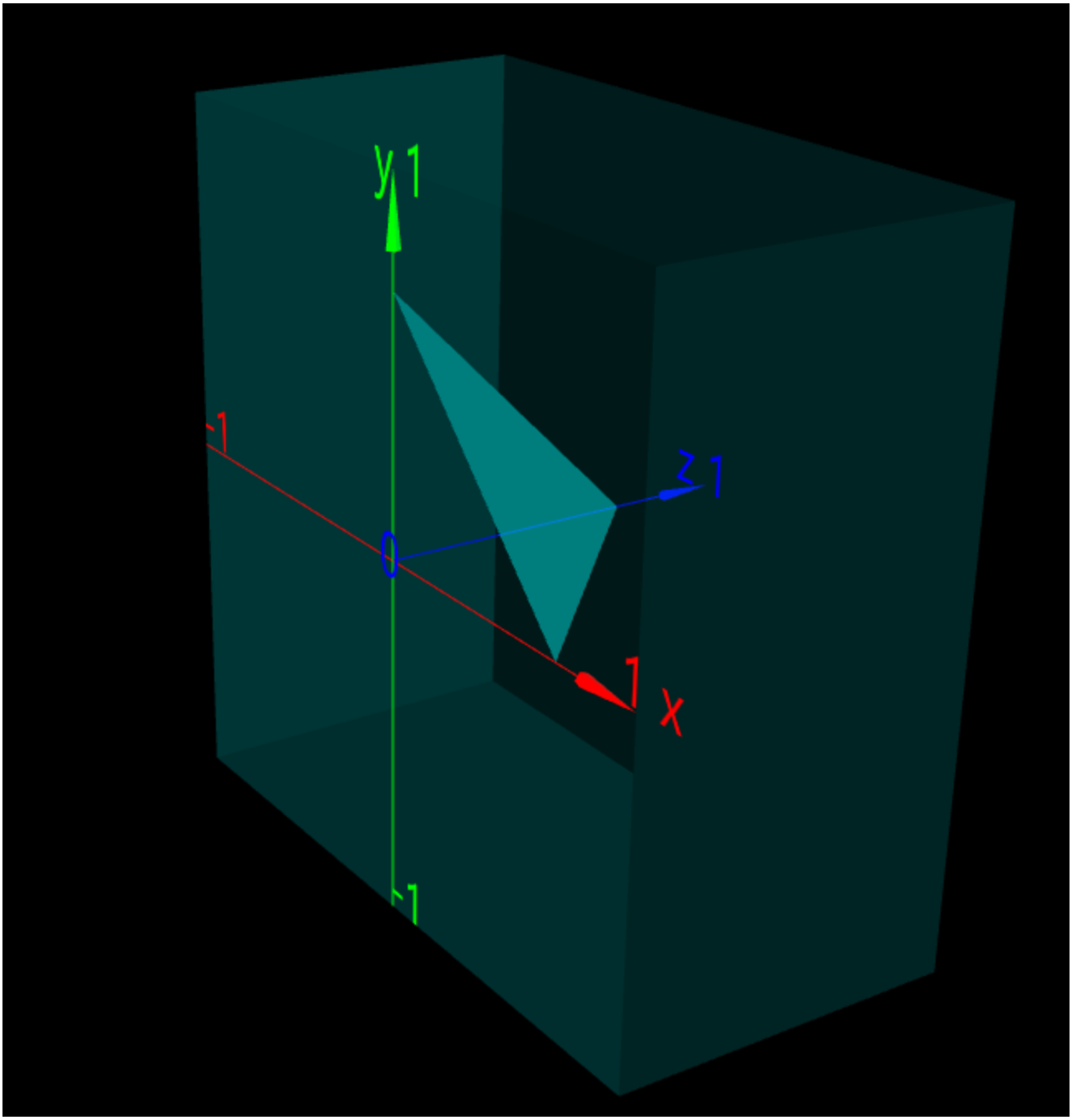
WebGPU坐标系**z轴**与Canvas画布垂直，朝向屏幕，z坐标的范围是[0,1]。



对于这种WebGPU坐标系，在图形学中，有个专门的名，就是标准化设备坐标系，对应英文名Normalized Device Coordinates，简称NDC，因为坐标范围是-1~1或0~1的相对值，你把NDC称为归一化设备坐标系也行。

WebGPU渲染规则(投影)

在x、y、z轴上各取一个点创建一个等边三角形。

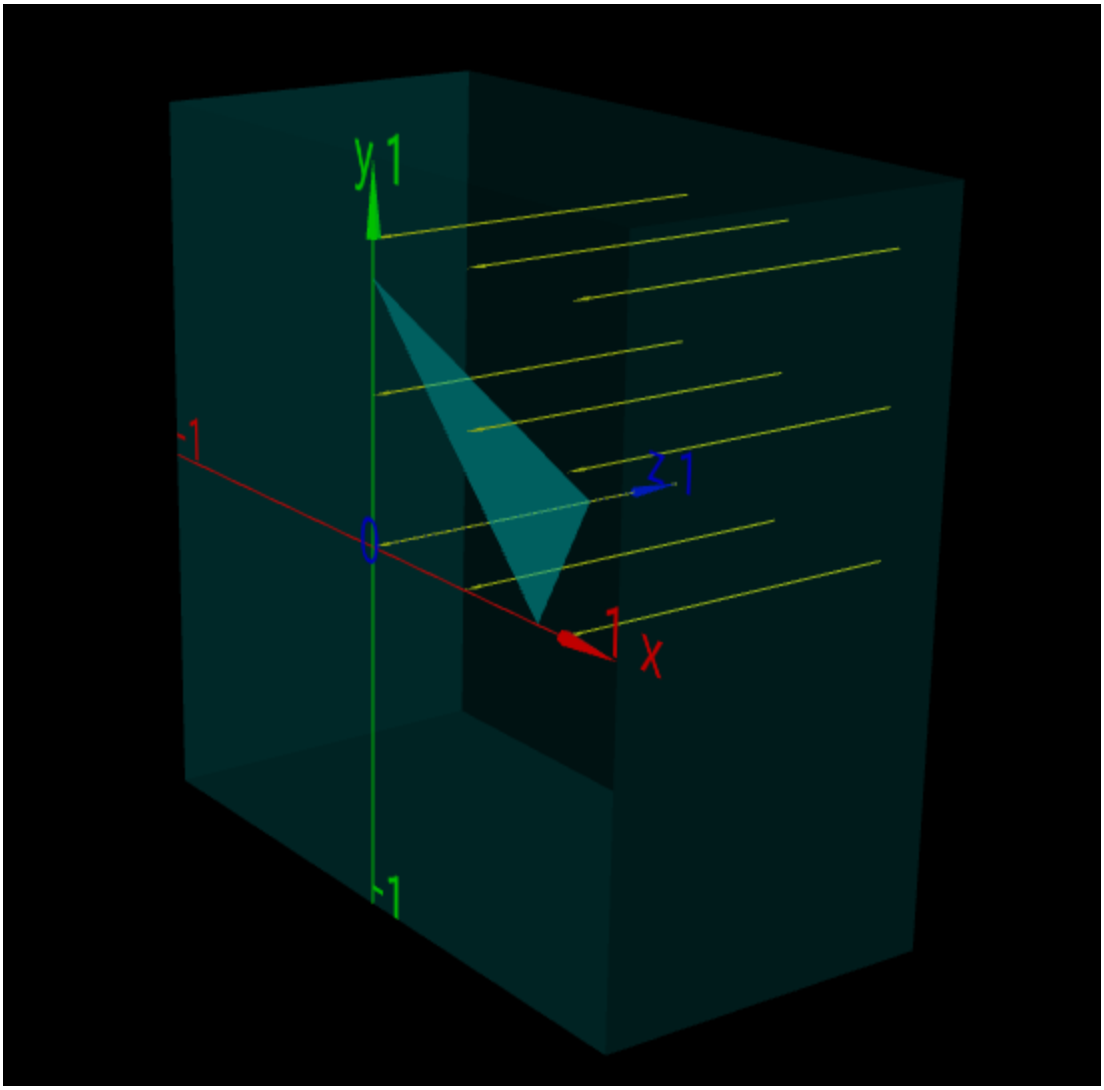



```js
const vertexArray = new Float32Array([
    1.0, 0.0, 0.0,
    0.0, 1.0, 0.0,
    0.0, 0.0, 1.0,
]);
```

那么默认情况下，WebGPU会如何渲染上面顶点坐标定义的三角形？

为了大家更好理解，我们假设在WebGPU的3D空间中，存在一束平行光线，沿着z轴照射到XOY平面上，这时候3D空间中的三角形会在XOY平面上产生投影，就像生活中，人在太阳光下，会地面上产生投影。

这时候，z轴上的任何顶点，投影后，其实都在坐标原点，这样上面一个等边三角形，三个点投影后，就是两个点在x和y轴，z轴上的点投影到坐标原点，这样三个点连接起来，渲染的投影结果就是一个直接三角形。



上面等边三角形顶点坐标和下面三餐性顶点坐标，在WebGPU默认情况下，投影效果其实一样的

```js
const vertexArray = new Float32Array([
    1.0, 0.0, 0.0,
    0.0, 1.0, 0.0,
    0.0, 0.0, 0.0,
]);
```

测试WebGPU 渲染范围

WebGPU坐标系x和y的坐标范围是[-1,1]，z坐标的范围是[0,1]。

WebGPU默认的渲染规律是，如果你的几何图形，超出xyz长方体空间范围的部分会被剪裁掉，不显示。

三个顶点坐标都没有超出范围，可以看到完整三角形，点1的x为z刚好和右侧canvas画布边缘重合，点2y为1，刚好和canvas画布的顶部边缘重合。

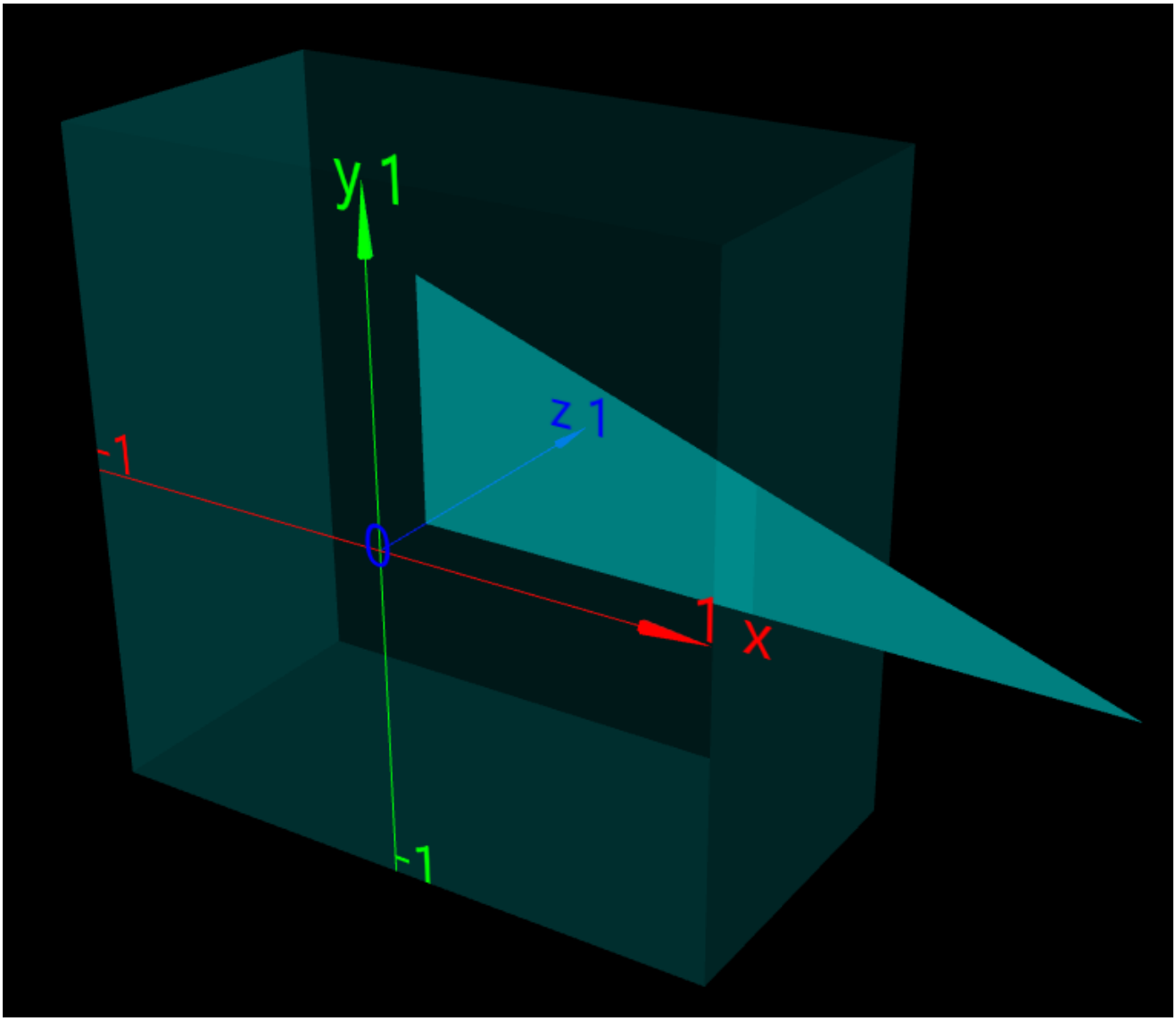
```js
const vertexArray = new Float32Array([
    1.0, 0.0, 0.0,
    0.0, 1.0, 0.0,
    0.0, 0.0, 0.0,
]);
```

下面三角形z坐标都是2.0，渲染的时候，在canvas画布上，你可以看不到三角形。

```js
const vertexArray = new Float32Array([
    1.0, 0.0, 2.0,
    0.0, 1.0, 2.0,
    0.0, 0.0, 2.0,
]);
```

顶点1的x坐标超出范围，三角形超出WebGPU渲染范围部分不显示，三角形渲染不完整

```js
const vertexArray = new Float32Array([
    2.0, 0.0, 0.2,
    0.0, 1.0, 0.2,
    0.0, 0.0, 0.2,
]);
```



← 7. 渲染命令(至此完成第一个案例)

9. 三角形拼接矩形→