

🎯 6. gl-matrix生成顶点着色器的矩阵

你可以把本节课内容作为前面三小节的练习题，尝试先自己写下，再看视频。

- gl-matrix生成缩放、平移等模型矩阵
- uniform传递矩阵数据
- 顶点着色器使用矩阵变换顶点

引入gl-matrix库

根据你自己文件gl-matrix库的地址设置路径。

gl-matrix库使用的基本语法，可以参考[2.3小节](#)。

```
// 引入gl-matrix库
import * as glMatrix from '../..gl-matrix/index.js'
```

js

glMatrix生成顶点着色器缩放矩阵

```
// 传递着色器对应uniform数据
const mat4Array = glMatrix.mat4.create();
//缩放变换
glMatrix.mat4.scale(mat4Array, mat4Array, [0.5, 0.5, 1]);
// 在GPU显存上创建一个uniform数据缓冲区
const mat4Buffer = device.createBuffer({
  size: mat4Array.byteLength,
  usage: GPUBufferUsage.UNIFORM | GPUBufferUsage.COPY_DST
});
// mat4Array里面矩阵数据写入uniform缓冲区mat4Buffer
device.queue.writeBuffer(mat4Buffer, 0, mat4Array);
```

js

glMatrix生成顶点着色器平移矩阵

js

```
// 传递着色器对应uniform数据
const mat4Array = glMatrix.mat4.create();
//生成平移变换矩阵
glMatrix.mat4.translate(mat4Array, mat4Array, [-1, -1, 0]);
// 在GPU显存上创建一个uniform数据缓冲区
const mat4Buffer = device.createBuffer({
    size: mat4Array.byteLength,
    usage: GPUBufferUsage.UNIFORM | GPUBufferUsage.COPY_DST
});
// mat4Array里面矩阵数据写入uniform缓冲区mat4Buffer
device.queue.writeBuffer(mat4Buffer, 0, mat4Array);
```

模型矩阵(先平移、后缩放)

js

```
// 传递着色器对应uniform数据
const modelMatrix = glMatrix.mat4.create();
//后发生缩放变换，先乘
glMatrix.mat4.scale(modelMatrix, modelMatrix, [0.5, 0.5, 1]);
//先发生平移变换，后乘
glMatrix.mat4.translate(modelMatrix, modelMatrix, [-1, -1, 0]);
// 在GPU显存上创建一个uniform数据缓冲区
const modelMatrixBuffer = device.createBuffer({
    size: modelMatrix.byteLength,
    usage: GPUBufferUsage.UNIFORM | GPUBufferUsage.COPY_DST
});
device.queue.writeBuffer(modelMatrixBuffer, 0, modelMatrix);
```

模型矩阵(先缩放、后平移)

js

```
// 传递着色器对应uniform数据
const modelMatrix = glMatrix.mat4.create();
//后发生平移变换，先乘
glMatrix.mat4.translate(modelMatrix, modelMatrix, [-1, -1, 0]);
//先发生缩放变换，后乘
glMatrix.mat4.scale(modelMatrix, modelMatrix, [0.5, 0.5, 1])
// 在GPU显存上创建一个uniform数据缓冲区
const modelMatrixBuffer = device.createBuffer({
    size: modelMatrix.byteLength,
    usage: GPUBufferUsage.UNIFORM | GPUBufferUsage.COPY_DST
});
```

```
device.queue.writeBuffer(modelMatrixBuffer, 0, modelMatrix);
```

← [5. WebGPU传递uniform数据](#)

[7. WebGPU动画\(uniform旋转矩阵\)](#) →