

姓名 刘俊伶 学号 2022141461136

1 思路

1.1 实现墨西哥帽模型

墨西哥帽模型使用的函数是：

$$z(x, y) = A(1 - r^2)e^{-r^2}$$

$$\text{where}(r = \sqrt{x^2 + y^2}).$$

通过计算不同的 (x,y) 坐标下的值我们可以得出不同的y值，以接着把坐标输入进 *points*。

这里每一个点坐标以一个步长为单位。将其输入进入一个临时数组，js数组不能直接使用 `int\[\]\[\]` 这一点很不友好。

接着上色即可。

1.2 实现MVP矩阵

1.2.1V

实现mvp矩阵，主要是计算eye的位置，通过计算球坐标即可得出，公式为：

$$x = r \sin \theta \cos \varphi$$

$$y = r \sin \theta \sin \varphi$$

$$z = r \cos \theta$$

在实际使用时，y和z要互换一下位置：

$$x = r \sin \theta \cos \varphi$$

$$z = r \sin \theta \sin \varphi$$

$$y = r \cos \theta$$

接着使用 lookat就可以了，在后续难点中讲解到了up的计算。

1.2.2P

实现p很简单，就直接带入即可

```

if(isOrth){
    return ortho(left, right, bottom, ytop, near, far);
}else{
    return perspective(fov, aspect, near, far);
}

```

1.3 实现teapot模型

teapot是导入的模型。在导入时，webGL中要进行一般这样的操作：

- 准备obj文件
- `new XMLHttpRequest()`: 创建一个 `XMLHttpRequest` 对象，这个对象用于与服务器进行交互，允许在不重新加载页面的情况下发出 HTTP 请求。
- `xhr.open('GET', objFilePath, true)`: 这是用来初始化请求。`GET` 是请求类型，`objFilePath` 是要请求的 `.obj` 文件的路径 `true` 表示这是一个异步请求（不会阻塞页面的其他操作）。
- 接着 `xhr.send()`; 表示请求服务器异步处理（这里不进行异步处理，会导致模型无法成功加载。）
- 接下来就是按照格式 `split` 读取到的数据。包括读取顶点信息，面信息，添加颜色信息。接下来就是添加页面的js操作。

2 重点难点

本次实验难点在up值的处理上，原则上，up向量一般取的是 (0,1,0)。但是在计算n时，(at-eye) 可能与up共线，这个情况下会导致计算

$$corss(up, n)$$

这个叉乘时发生错误，并且在后续的计算中，up和n的叉乘最后计算出的vc坐标变成左手系，而不是右手系。导致画面发生偏转。

解决方法：

只需要在n向量和(0, 1, 0)或者(0, -1, 0)相等时进行up取反操作即可。但是这只是完成了不重合时的视角正确。依然还是会发生在临界位置的视角抖动。

此时，我们需要计算一个额外的向量来替代原来的up向量，这个向量我取的是

$$n - n_{last}$$

但是这样还不够，这样还不够，在逆时针旋转回来时会发生翻转。解决办法是，添加一个判断是否翻转的函数即可。如果翻转，那么返回的就是

$$-(n - n_{next})$$

这个向量。这样就完美解决了视角抖动的问题。

相关代码如下：

```

function formViewMatrix(){
    var radius = 2.0;
    const at = vec3(0.0, 0.0, 0.0);

    // 计算eye位置
    var eye = vec3(
        radius * Math.sin(radians(theta)) *
Math.cos(radians(phi)), //x
        radius * Math.sin(radians(phi)), //y
        radius * Math.cos(radians(phi)) *
Math.cos(radians(theta)) //z
    );
    // 计算观察方向n, 并归一化
    var n = normalize(subtract(at, eye));
    //n和up重合
    if (areVec3Equal(n, vec3(0.0, 1, 0.0)) || areVec3Equal(n,
vec3(0.0, -1.0, 0.0)) ){
        up = vec3(0,-up[1],0);
        var temp = normalize(subtract(n,lastn));
        lastn = n;
        return isPhiIncrease(phi)? lookAt(eye, at, vec3(-
temp[0],-temp[1], -temp[2])): lookAt(eye, at, temp);
    }
    isPhiIncrease(phi);
    lastn = n;
    // 返回视点变换矩阵
    return lookAt(eye, at, up);
}
//是否相等
function areVec3Equal(vecA, vecB, epsilon = 0.00001) {
    if (!vecA || !vecB) return false;
    return Math.abs(vecA[0] - vecB[0]) < epsilon &&
        Math.abs(vecA[1] - vecB[1]) < epsilon &&
        Math.abs(vecA[2] - vecB[2]) < epsilon;
}
//判断phi是否增加

```

```
function isPhiIncrease(phi){  
    if (phi>phi0){  
        phi0=phi;  
        return true;  
    }  
    else{  
        phi0=phi;  
        return false;  
    }  
}
```

3 心得

本次实验我收获颇多，我不仅学会了如何使用webGL构建一个完整的墨西哥帽模型，还学会了MVP矩阵的生成和应用，使得我可以对3d场景进行漫游。同时还学会了模型的读入，导入了一个teapot。同时还解决了up向量的计算问题，完成了没有视角抖动的场景漫游。