# 计算机图形学

# 课程报告

**作品名称：**

**报告日期： 2019.12.31**

**学生姓名： 梅洛瑜**

**学生学号： 71117408**

# 一、作品简介

本作品包含三个独立物体：bilibili小电视、小飞鸡、表情包方块

负责部分：模型设计、模型导入、物体的旋转和移动

# 二、设计思路

## 灵感来源

### Bilibili小电视

Bilibili小电视灵感来源于热门视频网站哔哩哔哩的logo

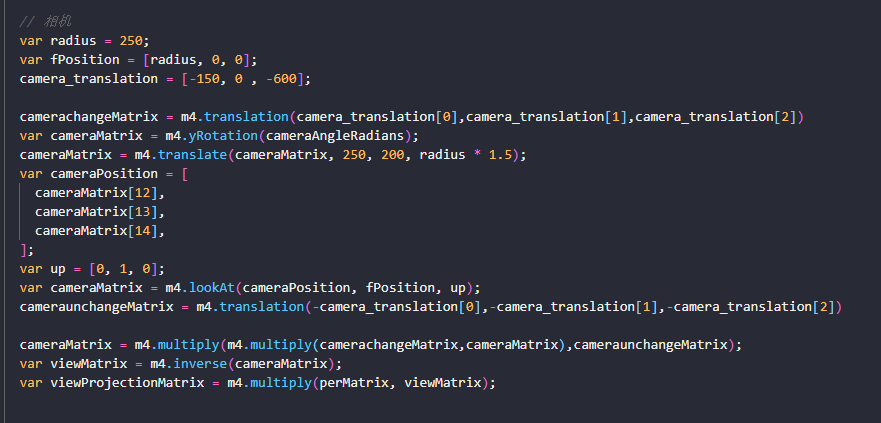


# 二、实现部分

## 场景漫游

### Bilibili小电视&小飞鸡

### 相机也是一种对象，把相机想象为一个对象，然后最后相机的视角就是对相机矩阵求逆, 乘到物体视角上



### 我们的地图中心我放在 （0,0,-600 ） 处， 所以我们在渲染时需要将物体的矩阵先做转置并移动到地图中心的位置，与相机结合渲染时还需知道相机的位置。于是我们设计了两种漫游方式，对应两个tick()状态。



## 物体移动和旋转

### 两种漫游方式



按钮背后原理是，设计了两个tick()函数，一个tick1()控制物体动+相机动，另一个tick2()控制物体运动。

### 物体移动

为了实现物体运动，设置了go()按钮，背后的逻辑是设定相应的运动方程，同时在渲染每一帧时进行计数，设定物体的上下左右边界以及移动速度，在物体达到规定边界时将速度方向反向。以上方程都是针对物体坐标进行更改，在渲染时仍要知道相机的相应位置，才能使得实际视野里的物体处在正确的位置。



实现对象绕指定轴旋转，需要利用对象的平移和绕坐标原点旋转。

由于对象中心不在坐标原点，首先平移坐标轴至对象中心，然后绕坐标轴进行旋转操作，旋转完成之后复原坐标轴

### 相机移动同时物体移动

相机也是**一种对象**，把相机想象为一个对象，然后最后相机的视角就是对相机**矩阵求逆**, 乘到物体视角上

我们的地图中心我放在 （0,0,-600 ） 处， 相机做旋转需要用到和物体旋转一样的，确定中心轴，所以用 changematrix 和unchangematrix 在做变换的前后使用，就可以改变中心位置。

此例子中做的交互，围绕着地毯中心旋转(0,0,-600)



## 纹理贴图

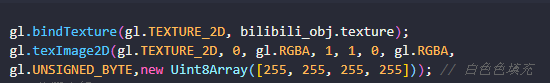
与之前的工作兼容

之前直接导入物体的顶点，并通过设置顶点color进行上色，尝试以后发现工作量较大。

在需要进行纹理处理时，如果逐顶点进行纹理设置可能不太现实。



用 color 来上色，工作量还比较大，如果全部改成纹理，比较麻烦，于是设置顶点与纹理两个矩阵，并在渲染时采用矩阵相乘的方式同时考虑顶点颜色以及纹理。如果想要单独使用顶点颜色则将纹理矩阵设为全1，同理，如果想要单独使用纹理矩阵则将顶点设为全1.



### 纹理图片处理

在一个物体上绘制多个纹理，由于资源以及IO次数考虑，使用一张图仅加载一次资源，把多个图像拼接成一个256\*1024的图形。在绘制物体的不同面时，首先设置相应面图像资源在拼接图像中所处的位置，然后通过图形坐标位置选择0.25 0.50 0.75 1来截取想要的图形。

# 四、实验中遇到的问题和解决方法

### 相机移动同时物体移动

一开始在写好相机运动的函数以后，不知道如何控制相机运动时视野的正确变换，后来我们想到将相机当作物体，相机朝向就是视野方向，有focus点以及相应的法相。

相机也是一种对象，把相机当作对象操作，事情一下子就简单了

html5 的帧动画api 是让动画看着像个样子的基础





## 纹理贴图

### 跨域访问问题

两种跨域访问：

1.SimpleHTTPServer

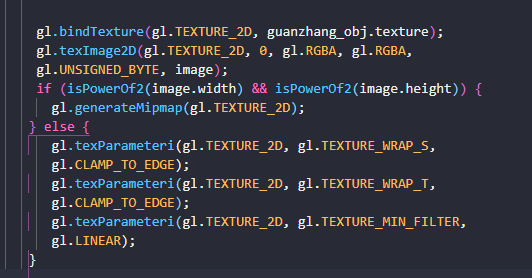
2.base64

由于采用HTTPServer对于小项目来说没有必要且客户端需要消耗更多资源来加载素材，我们未搭建SimpleHTTPServer，反之我们采用base64的方法，将对象及纹理资源转为base64格式并直接写入js文件中。

### 纹理图片处理

解决图片不是2的幂的方法

实际编程时，如果图像边长不是2的幂，则会报错并无法显示。解决此问题不需要对图像进行更改，在读入图片以后如果发现图片的边长不是2的幂则将边缘以相同rgb扩展包裹直到达到2的幂



# 五、个人总结

在短短的一学期课程内，老师的耐心教学自己的课外学习以及其他组的精彩作品，使我对图形学特别是实时图像渲染技术有了很深的印象。我在课堂上学习了从二维到三维图像渲染以及光源和纹理的设计。同时通过小组合作，了解了通过webgl编写实时图像渲染程序的整个过程。在设计场景漫游时，通过一点点尝试最终得出将相机视为物体进行计算和渲染的方法，以及更加高效的图片资源合并。在添加纹理效果时，解决了跨域访问问题以及顶点颜色和纹理的渲染问题，学到了可以通过SimpleHTTPServer以及直接将对象转为base64来解决跨域访问。同时，也掌握了js代码编写的许多小技巧。