目录

[1. 主体 1](#_Toc16253627)

[2. 阴影： 2](#_Toc16253628)

[2.1 延伸：锯齿 4](#_Toc16253629)

[3. 裁剪 6](#_Toc16253630)

[4. 改用重心权重—光栅化 8](#_Toc16253631)

[5. phong实现 9](#_Toc16253632)

背面剔除？放在哪

不加线段或多边形裁剪

选项：

投影：透视投影 正交投影 改投影矩阵

光栅化： DDA Bresenham 改scanline到pixel

CVV中各裁剪法

渲染器：

TransformMat

FrameBuffer

DepthBuffer

Camera:

ModelViewTransformMat

Position

LookAt

Up

## 主体

1. 准备阶段

渲染单个模型

draw(model, texture)

模型：3点一三角形的点数组point\*

纹理：图片

1. 几何阶段

draw\_primitive()

frustumcull() ：

点转换(perspective-normalization matrix透视规范化变换矩阵)到CVV(x=+-1,y=+-1,z=+-1)

剔除

视口变换

分矩形

生成多条scanline

光栅化（DDA）: rasterize() 得到像素点

drawpixel()

深度测试-DepthBuffer

窗口剔除

1. 着色阶段

在drawpixel()中加入对应纹理点的color

1. 屏幕初始化
2. 渲染器初始化
3. 逐图元渲染。device\_draw\_primitive
4. 模型变换，投影变换
5. 视见体（齐次裁剪空间）剔除--frustumcull
6. 保留w（），然后使用w进行透视除法
7. 视口变换新得到的点
8. 分梯形：根据y坐标排序（上到下123），再用根据3在12的线的左端还是右端，来形成两种分法
9. 对梯形进行扫描线遍历：
   1. 根据y，得到扫描线左右两点trapezoid\_edge\_interp
   2. 计算得到初始点v，步长（每个像素的纹理和颜色变化），线的宽度
   3. 绘制扫描线device\_draw\_scanline(device, &scanline);

保存的深度值可以用来做深度测试了

## 阴影：

shadowmap：

ShadowMap是比较流行的实时阴影实现方案

平行光源

为每个平行光源配置

方向

视场大小（关乎深度图大小）

摄像机（正投影）

MV+P(正交) T

深度纹理/深度信息图 Depth

非平行光

…

透视投影

…

渲染物体时，T\*点到 光源空间，如果深度小于，则为阴影

像素级别处理：

像素反推世界坐标》光源空间坐标》

还要计算视口变换

xyz乘w能够变大cvv逆转回去

矩阵的逆—伴随矩阵

为什么w能表示深度值，不用z

可以少声明一个变量？

裁剪体内z ：0~1 是从0看向1

。。。fb用了外部缓存。。都怪之前不看fb

求逆矩阵：1.伴随法 A⁻¹=A\*/|A| ( |A|为矩阵A的行列式的值，A\*为矩阵A的伴随矩阵 )

2.LU分解法

调大深度差，减少浮点数的影响

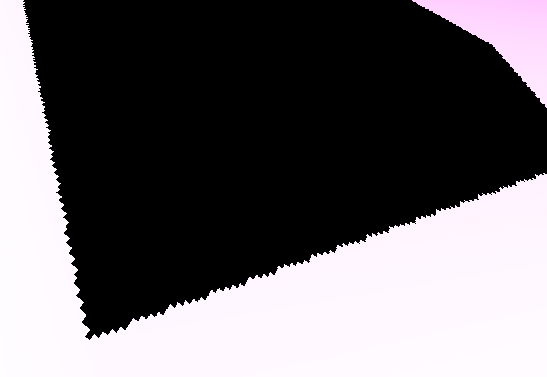
对阴影贴图模糊

背面产生的阴影怎么办？？

从光源处看阴影会产生问题

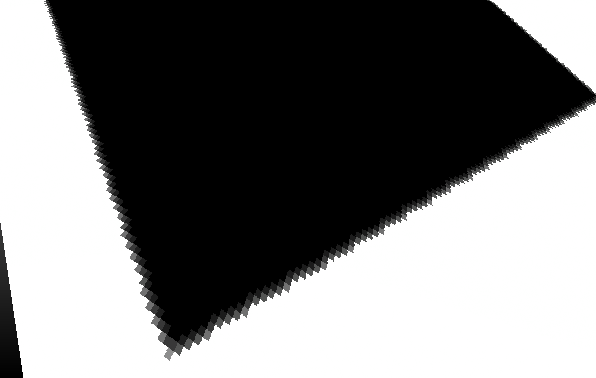
shadowmap在工业上的使用

### 延伸：锯齿



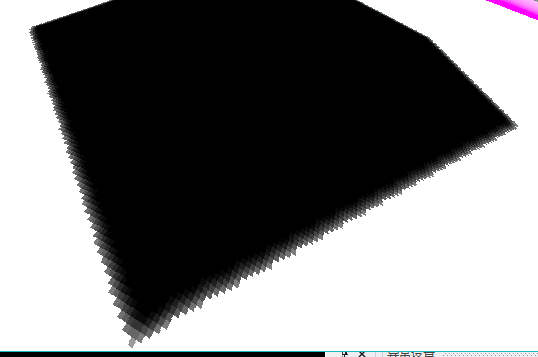
使用PCF滤波

每次都要遍历周围八个点

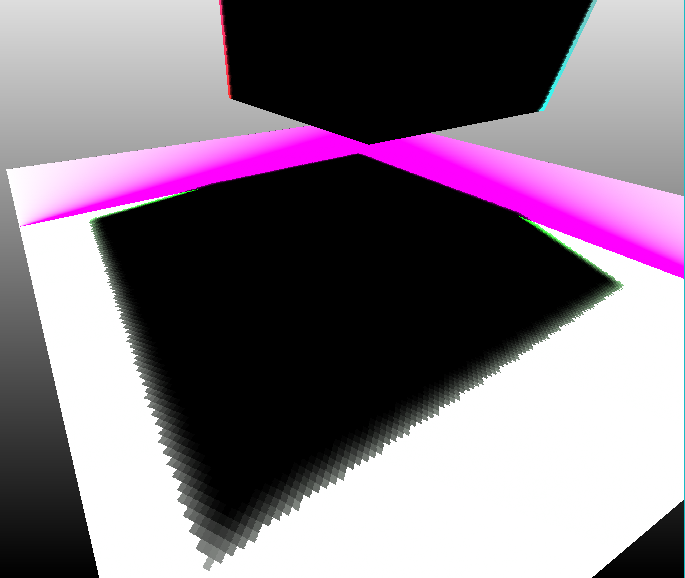


效果不是很明显？

遍历25个点



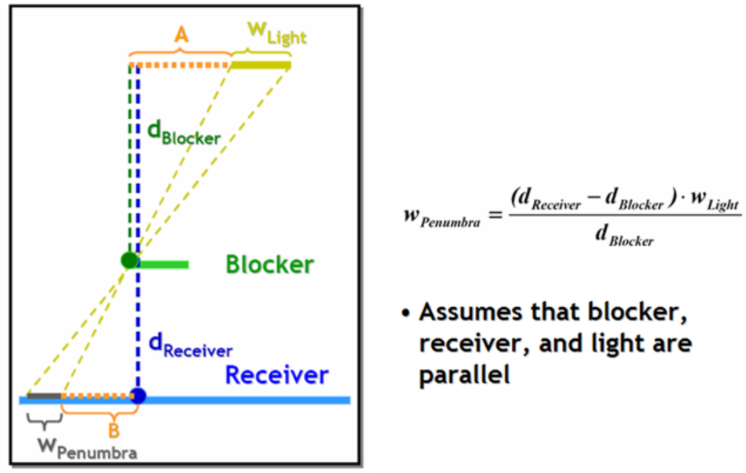
49个片元



PCSS

动态计算半影大小

相似三角形



如何判断是blocker？ 近一些就是？

Wlight是怎么定义的？

背面阴影

用光源位置点乘就好了？

设置rendertarget

## 裁剪

不仅要插值位置，还要颜色。所以是对vertex\_t操作

用哪个面来与直线相交？需要先进行透视除法吗？不然不能归到同一视锥体下

所以不会齐次裁剪空间进行裁剪

片元着色器可以裁剪但是非常卡

cvv面和三角形边的相交计算时，会通过暂时的4d变3d过程得到cvv中的值，这个只是暂时的，只是为了得到3d空间中的值计算用！！！

归一化放在裁剪完后是为了减少对不需要显示的面片的顶点进行透视除法

3内

s1 ==0 && s2==0 && s3==0 都在cvv内

3外：

s1 && s2 && s3 !=0 在同一个面的一边，抛弃

s1 != 0 && s2 != 0 && s3 != 0 都不在cvv内，但为了加快，先直接抛弃

1内，2外

2内，1外

https://www.cs.drexel.edu/~david/Classes/CS430/Lectures/L-14\_Color.6.pdf

1&4 = 0

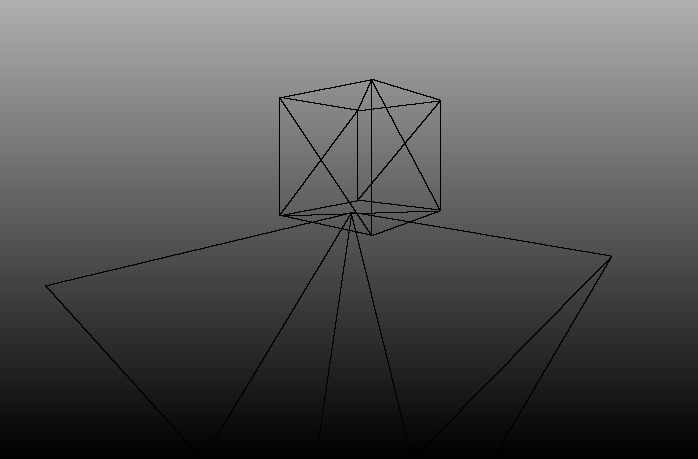
1&&4 = 1

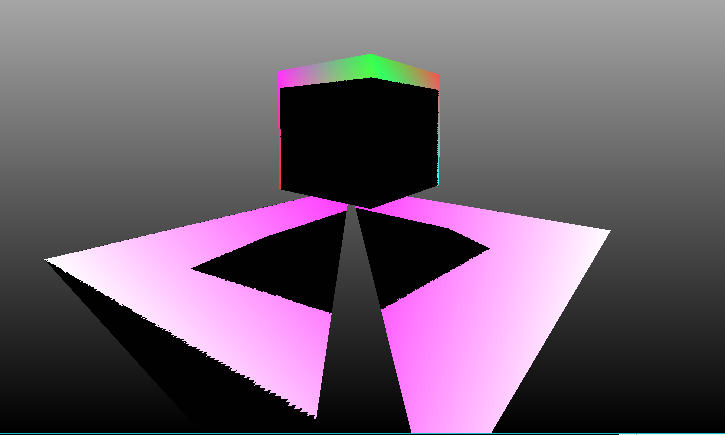
为什么会爆？？ 栈溢出

离远了会变形

裁剪的点有错的颜色or阴影？

为什么会根据距离远近变色





1.深度图未设最大值

精度问题？

裁剪新的点的

裁剪放到着色器！！！

为了方便插值其它值

## 改用重心权重—光栅化

能够更方便对各种值进行插值

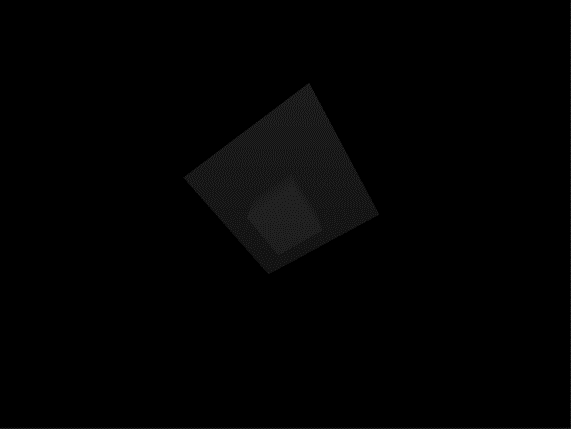
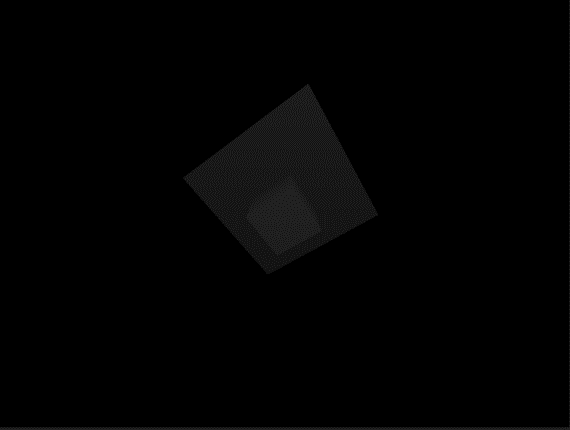
不再需要用视口变换，用权重就能求回顶点坐标

shader的坐标变换没有问题

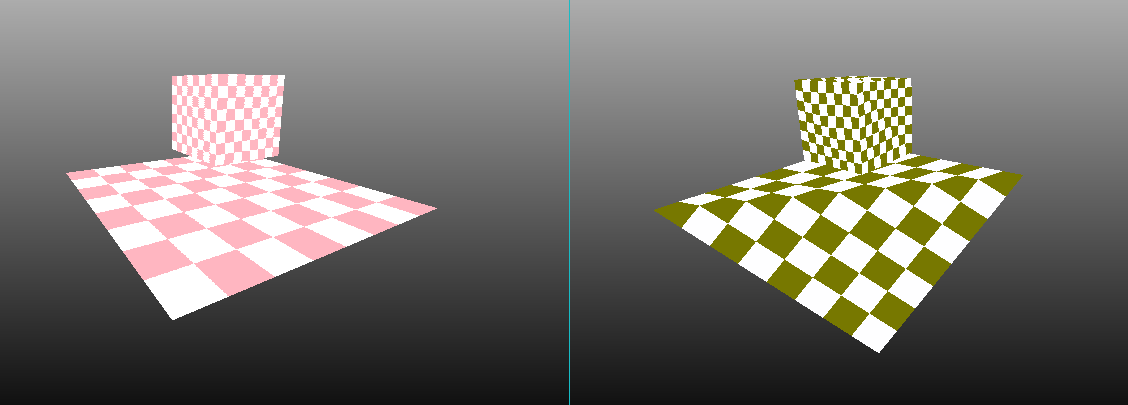
打印图片

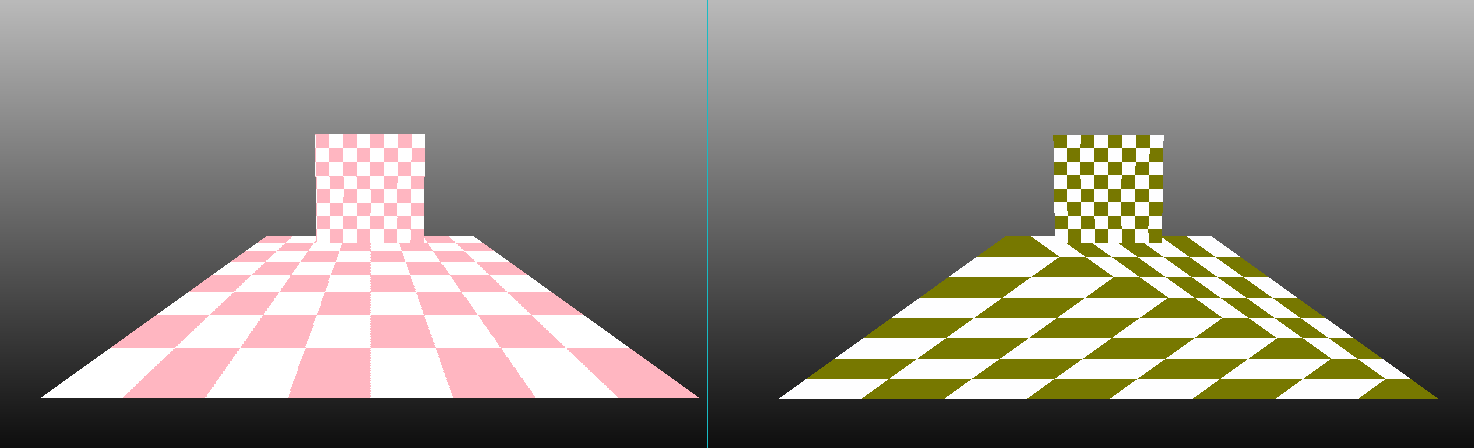
影子的变化错误！！！注意浮点数？？

应该是shader中的坐标变化有问题



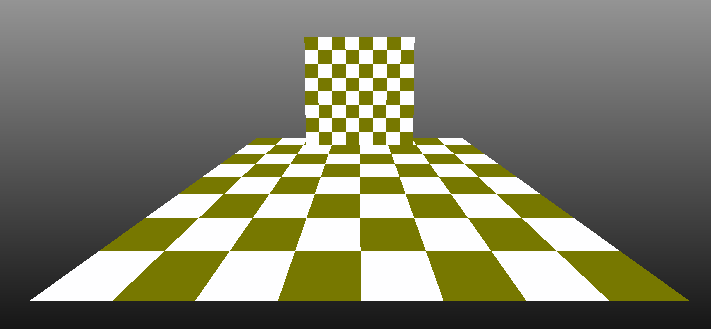
插值有问题！！！！



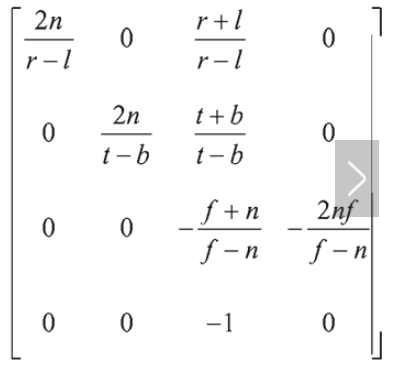


加入透视矫正：

屏幕空间的 x y 与 世界中的x y z， u，v等呈非线性关系



为什么除的是w 不是z！！！因为透视投影之后的z不能使用，是要用原来世界坐标下的z。

而对顶点(x, y ,z, w) 使用完透视投影矩阵后，w’ = -z，所以可以用w来作为 原来的z

## phong实现

自定义模型：

obj格式

f 顶点索引/uv点索引/法线索引

shader.N = (shader.M.inverse()).transpose();

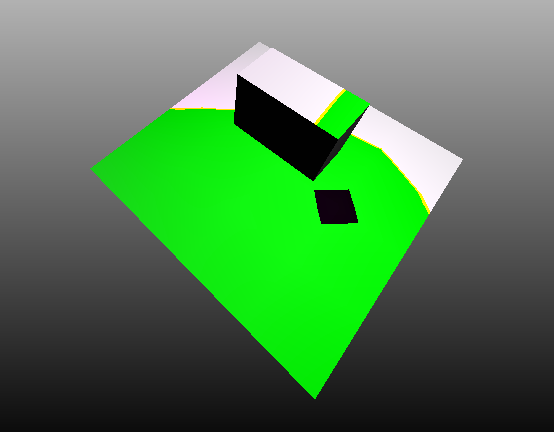
??? 逆的转置？？有什么用。法线什么时候用过M了？？。没看懂

自己写

动不了了！！？？只有纹理shader能显示

深度纹理没问题

spec出问题。。。。原来是向量单位化的时候，除的是模长，应该是模长的平方



unity中

ambient = 外面设置

ambient = UNITY\_LIGHTMODEL\_AMBIENT.xyz;

diffuse = 光的颜色 \* 漫反射的颜色 \* 率

fixed3 diffuse = \_LightColor0.rgb \* \_Diffuse.rgb \* max(0, dot(worldNormal, worldLightDir));

spec = 光的颜色 \* 镜面反射的颜色 \* 率

fixed3 specular = \_LightColor0.rgb \* \_Specular.rgb \* pow(max(0, dot(worldNormal, halfDir)), \_Gloss);

漫反射的颜色，镜面反射的颜色 在shader中可调

反射大致正确，应该是diffuse太大了

侧面的法线错了。。。。没错，只是光线放在了正上方所以没侧面。

光线的位置与相机的分离问题

光总在相机右边了

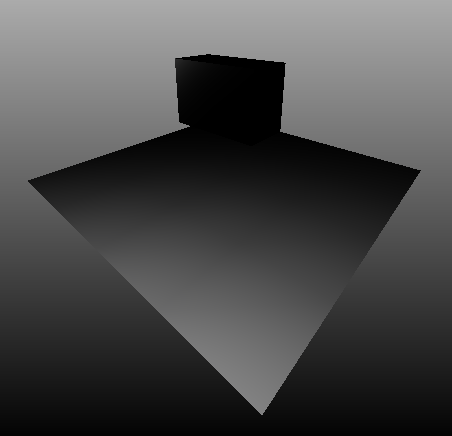
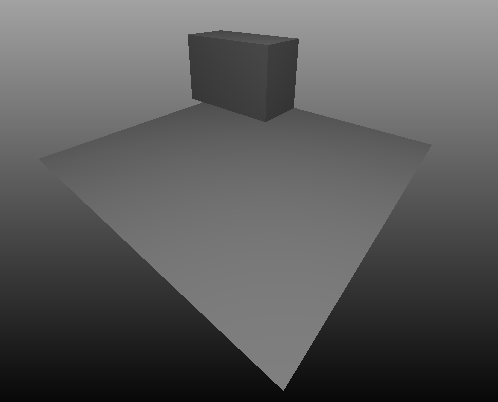
太亮的地方变黑了！spec变负数了？？？

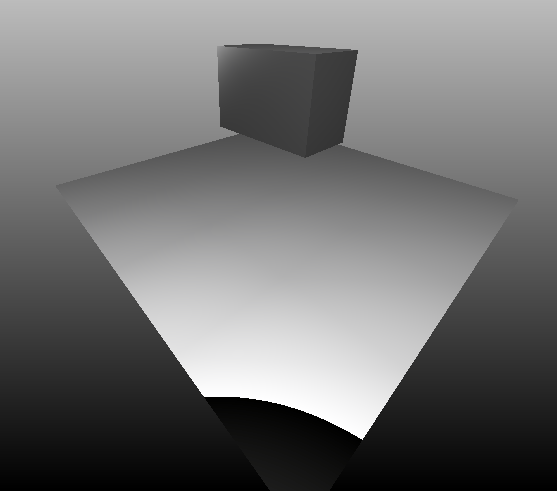
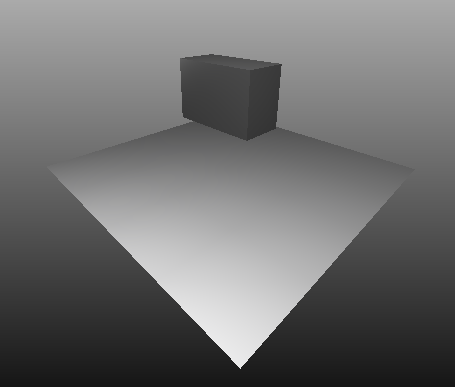
ambient+diffuse正常

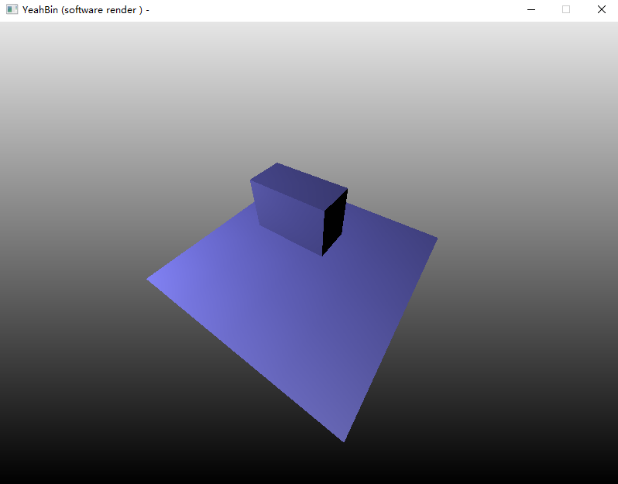
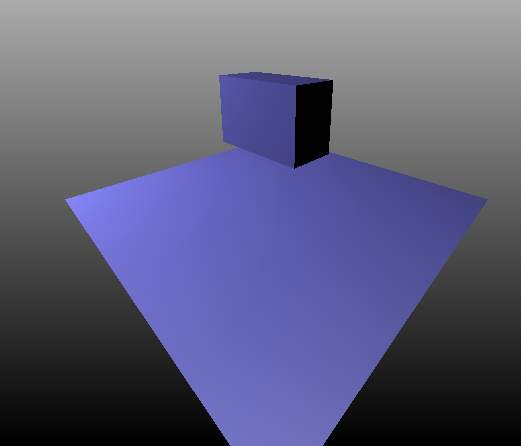
spec正常

！！因为超过1的时候，前24位未必都是1.

clamp一下片元着色器的结果







高光跟随太明显。

立方体的侧面打不上高光？？

todo：(多一点上传)

切换数组

light初始化

整理matrix

将光颜色， 强度 放入class Light

phong模型中 点光源的光线衰减

纹理颜色

M放进mesh中

mesh与model分离