2020年图形学作业:

作业 1

1. What is the resolution of the image? What is the aspect ratio of the image?

答：~~1) resolution(分辨率): 屏幕或窗口上，水平和垂直方向的单位长度（英寸）的像素点数目，如 1920x1080 是指水平 1920 个像素/英寸，垂直 1080 个像素/英寸。~~

2) aspect ratio(纵横比): 以单位长度或像素数目衡量，图像的宽与高之比，如一个图像的宽是 3 英寸，高是 2 英寸，则 aspect ratio=3/2=1.5；一个显示器的分辨率是 1280x1024，则 aspect ratio=1280/1024=1.25。

2. Movies are generally produced on 35mm film that has a resolution of approximately 2000x3000 pixels. What implication does this resolution have for producing animated images for television as compared with film?

答：要达到稳定的画面质量，需要 24 帧/秒以上的刷新率。对于 television，分辨率为 2000x3000 的真彩色，则需要帧缓存大小 2000x3000x3=18MB/帧，意味着每秒需要处理 24 次以上的 18MB 大小的数据，即 41.67 毫秒就要处理 18MB 的数据，需要强大的显示处理能力。

电影只要显示胶片 24 帧/秒即可，不需要帧刷新。

3. 查找图形卡有哪些主要品牌？简述其中一种的主要指标有哪些？

N 卡和 A 卡。

基本指标有：计算单位的核的数量；每个核的频率；显存大小和位宽；显卡功率和 GPU 最高温度；支持的显示器最高分辨率和个数；支持的图形库；数据线接口等。

图形渲染指标：光照加速计算、纹理计算，HDR 效果等

Nvidia：GEFORCE RTX 3090 等

ARM：AMD Radeon™ RX 6900 XT 等

4.简述 OpenGL、OpenGL ES 和 WebGL 的相同和不同的特性。

作业 2

1. 第 1 章作业第 1.8 题

答：防止闪烁，如果每秒 72 帧，则每帧需要 13.89 毫秒，即帧缓存需要在不到 14 毫秒处理一帧画面，即 14 毫秒内处理 1280x1024x3=3.9MB 的显示数据。每帧有 1280x1024 个像素，所以每个像素的传输时间是 1/1280x1024x72=10.59ns(纳秒=1000us=1000x1000ms=1000x1000x1000s)

对于 480x640 的 60 帧/秒的隔行扫描，1/480x640x30=108.5ns

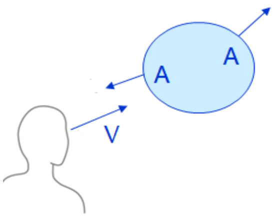
2. Installation OpenGL Lib in visual studio 2010+, then writing a green triangle program with OpenGL Shader or WebGL with JavaScript three.js.

3. Program the 2D Sierpinski Gasket—see §2.8, §2.9 for 2D

作业 3

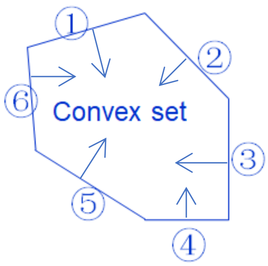
1. 给定视线方向 V 和物体 A 点的法向量，如何判断 A 点是否能被看到?

答：设 A 点的法向量为 N，则，当 V•N>0 时，A 点不可见；当 V•N<=0 时，A 点可见（注意：不能用 A 作为法向量）



第1题

2.如何判断一个多面体是一个凸多面体?



第2题

答：对每一个多面体的面 i，如果其他面都在面 i 的前面，则多面体是凸多面体。

定义多面体 Si（i=1,···,n），Si 的法向量 N 向着多面体内部，则判断算法如下:

for(each Si)

for(each Sj)

if(Si <> Sj) 且 (Sj 不在 Si 的前面) 则多面体是非凸多面体，结束

双重循环结束，则多面体是凸多面体，结束

方法二：用点到面的距离判断：对多面体的所有三角面，求出剩余顶点到该三角面的距离，如果这些距离 d 符号一致，表示这些顶点都在该面的一边，否则不符合凸多面体定义，就是凹多面体

作业 4

1. write down the matrices for rotation about x and y axes

答:

rotation about y :

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

rotation about z:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Page 161. #4.1

答：证明

1. 旋转和等比例缩放：RS(a,a,a)=aR=S(a,a,a)R
2. 绕同一轴的连续旋转：Rx(θ)Rx(φ)=Rx(θ + φ)
3. 连续平移：T1(x1,y1,z1)T2(x2,y2,z2)=T(x1+x2,y1+y2,z1+z2)=T2T1

3. Page 161. #4.23

答：向量 u 和 v 不平行，则向量 n=uXv，同时垂直于 u 和 v 向量
再令 v'=nXu，则 u,n,v'三个向量正交，构成一个正交坐标系。v'在 u 和 v 所确定的平面内。

作业 5

1. 飞机移动的位置由滚转角、俯视角和偏航角以及与物体的距离确定。根据这些参数给出一个观察矩阵？

答：

1. 假设飞机在世界坐标系 OXYZ 的 A(Ax,Ay,Az)点，要通过滚转 roll(θ r)、俯视 pitch(θ p)和偏航 yaw(θ y)，飞到 B(Bx,By,Bz)点，则系列变换的两个步骤为:
   1．将飞机变换到眼睛坐标系;
   2．在眼睛坐标系，飞机从 A 点飞到 B 点。

其中步骤 1 要做：1）坐标原点平移到 A 点 T1(-Ax,-Ay,-Az)；2）坐标轴旋转，使得飞机的前后方向是 Z 轴正方向(眼睛坐标系的 N 轴)，上下方向是 Y 轴正方向(眼睛坐标系的 V 轴)，世界坐标系到眼睛坐标系的变换矩阵为 M1= R T1; 3) B 点变换到眼睛坐标系的 B'(B'x,B'y,B'z)

步骤 2 要做：1）在眼睛坐标系，飞机分别相对于 N，U 和 V 轴做滚转 roll、俯视 pitch 和偏航 yaw，设滚转角为 θ r、俯视角为 θ p 和偏航角为 θ y，则相对眼睛坐标系原点到预定方向 B'的变换为 Rz(θ r)Rx(θ p)Ry(θ y)，旋转的顺序可自己确定；2）从 A 点飞到 B 点，即眼睛坐标系的原点到 B'点，做平移变换 T2(B'x,B'y,B'z)。

以下是两个步骤的变换矩阵：

步骤 1：

1) T1 为世界坐标系原点平移到 A 点的变换矩阵，A(Ax,Ay,Az)为世界坐标系下 A 点坐标：

$$T_1=\begin{bmatrix}1 & 0 & 0 & -A_x\\ 0 & 1 & 0 & -A_y\\ 0 & 0 & 1 & -A_z\\ 0 & 0 & 0 & 1\end{bmatrix}$$

2) M1 为飞机所在的世界坐标系到眼睛坐标系的变换矩阵，设飞机在世界坐标系的前后方向的向量为 $n(n_x,n_y,n_z)$：则飞机的机翼方向为 $u(u_x,u_y,u_z)=(0,1,0)\times(n_x,n_y,n_z) = (n_z ,0,n_x)$；上下方向为 $v(v_x,v_y,v_z)=n\times u = (n_x,n_y,n_z)\times(n_z ,0,n_x) = (-n_xn_y, n_x^2+n_z^2,-n_yn_z)$

$$M_1=RT_1=\begin{bmatrix}u_x & u_y & u_z & 0\\ v_x & v_y & v_z & 0\\ n_x & n_y & n_z & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & -A_x\\ 0 & 1 & 0 & -A_y\\ 0 & 0 & 1 & -A_z\\ 0 & 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}u_x & u_y & u_z & -u\cdot A\\ v_x & v_y & v_z & -u\cdot A\\ n_x & n_y & n_z & -u\cdot A\\ 0 & 0 & 0 & 1\end{bmatrix}$$

3) 将 B 点从世界坐标系变换到眼睛坐标系的 B‘点的变换：

$$B' = \begin{bmatrix}B'_x\\ B'_y\\ B'_z\\ 1\end{bmatrix} = M_1\begin{bmatrix}B_x\\ B_y\\ B_z\\ 1\end{bmatrix}$$

步骤 2：

1) 在眼睛坐标系中，飞机通过旋转矩阵将机头对着 B‘点方向。因为飞机的滚转 Rz(θr)、俯视 Rx(θp)和偏航 Ry(θy)旋转都是相对于坐标轴的，所以它们不能连续旋转。若飞机没有滚转，机头对着 B‘点方向的变换是 H_1 到 H_2 的变换(在 XZ 平面投影)，即：

$H_x = h\cos\alpha\sin\theta$

$H_y = h\sin\alpha$

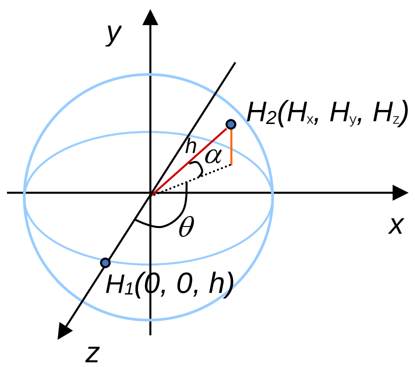
$H_z = - h\cos\alpha\cos\theta$

（其中θ是偏航角θy）

变换矩阵 R 为：

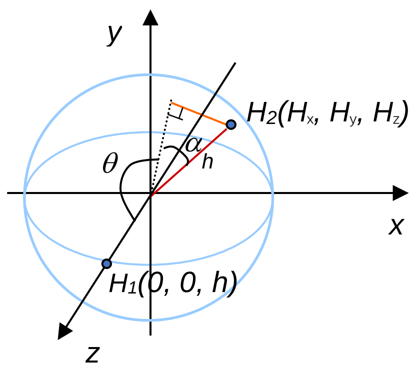
$$R=\begin{bmatrix}1 & 0 & \cos\alpha\sin\theta y & 0\\ 0 & 1 & \sin\alpha & 0\\ 0 & 0 & -\cos\alpha\cos\theta y & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

即：$H_2 = R\,H_1$



或 H_2 点向 YZ 平面投影的 H_1 到 H_2 的变换

其中θ是俯仰角θp

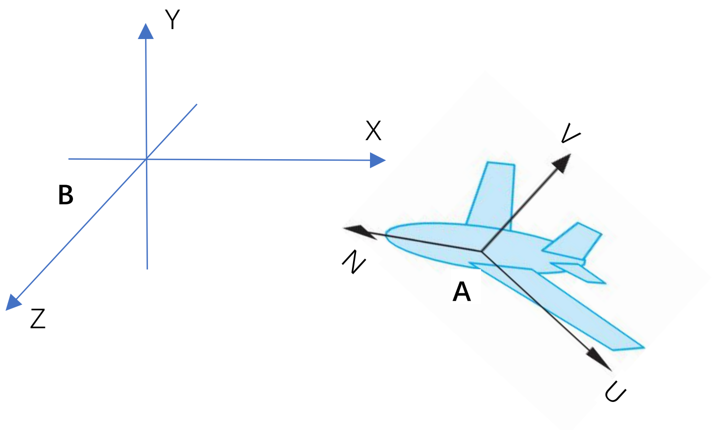


2）T2 为飞机在眼睛坐标系原点平移到 B'点的变换矩阵：

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & B'_x \\ 0 & 1 & 0 & B'_y \\ 0 & 0 & 1 & B'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

所以，飞机从 A 点到 B 点的变换矩阵 M 为：

$$M = T_2\, R\, M_1$$



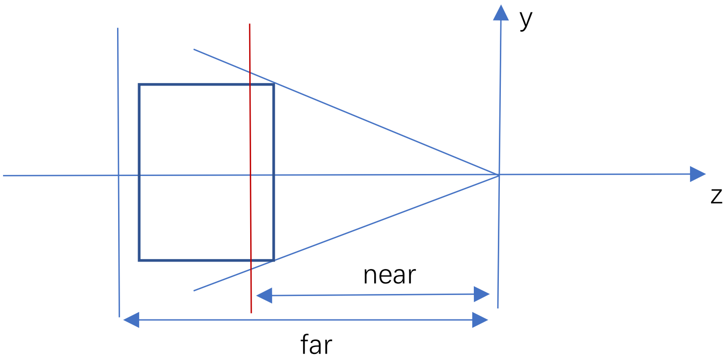
（要点：滚转 roll、俯视 pitch 和偏航 yaw 必须是绕坐标轴旋转，如果说明飞机的初始位置是在世界坐标系的原点，向前的方向为 Z 轴正方向，向上的方向为 Y 轴方向，则变换可以是 $T_2 R$）

2. 在进行视见变换时，眼睛空间的作用是什么？

答：显示在窗口的场景是视点的一个投影图，一般地，视点的方向是垂直于投影面的。在世界坐标系中，一个任意的观察表示场景中任意的投影面，这种投影变换是复杂的。眼睛坐标系是以视点方法作为 z 轴，投影面垂直于 z 轴，投影矩阵变得简单了。所以，视见变换是确定观察位置和方向，并将三维场景投影在投影平面上。眼睛空间的作用是**为了方便投影变换**，以及**简化视见体的裁剪操作**

3. 当我们从远处观察一个封闭房间的内部时，给出透视视见体的定义。

答：假设封闭房间为正方体，眼睛坐标系的 z 轴垂直于其中的 2 个墙面，一个为近墙面，一个为远墙面，则相对于视点的位置，**近平面要定义在比近墙面远的位置**，这样，近墙面就被视见体裁剪掉，可以在墙外观察到房间里



思考题：

场景中移动一个物体，既可以用 translate 变换作用在物体上实现，也可以用 Lookat 改变观察位置实现。即两种实现方法是互逆的。请问两种等价的实现方法，其实现时的参数有什么对应关系？translate 变换的参数为(tx, ty, tz)，Lookat 的参数为（eye_x, eye_y, eye_z, at_x, at_y, at_z, up_x, up_y, up_z）

答：初始状态，假设视点在世界坐标系的原点，向 Z 轴负方向看，物体在 P 点，平移后的物体在 P’ 点

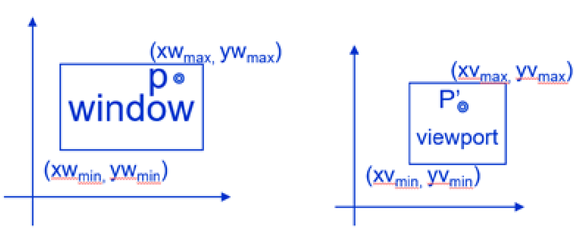
设 T= $\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 是物体的平移矩阵

则 T^{-1} 是坐标系的平移矩阵，即视点在新坐标系的原点，向新坐标系 Z 轴负方向看

与平移物体产生同样观察效果的 Lookat 定义为：原来的观察点 V_{eye}(eye_x,eye_y,eye_z)做 T^{-1} 平移变换，同时，观察方向 V_{at}(at_x,at_y,at_z)做 T^{-1} 平移变换，向上的向量(up_x,up_y,up_z)不变。

作业 6

1. 我们有一个变换，是从场景 2D 窗口映射到视见窗，其中缩放系数为(Sx，Sy)，讨论当 Sx≠Sy 时，视见窗的无变形的变换矩阵是什么。



答：因为 M= T_2 S T_1

$$T_1 = \begin{pmatrix} 1 & 0 & -xw_{min} \\ 0 & 1 & -yw_{min} \\ 0 & 0 & 1 \end{pmatrix} \qquad T_2 = \begin{pmatrix} 1 & 0 & xv_{min} \\ 0 & 1 & yv_{min} \\ 0 & 0 & 1 \end{pmatrix}$$

$$S = \begin{pmatrix} S_X & 0 & 0 \\ 0 & S_Y & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad 其中: S_X = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$S_y = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

若要变换无变形，则缩放系数要满足 Sx=Sy

当 Sx≠Sy 时，

1）若 Sx ＞ Sy，则取 Sy，即可以通过增加 XWmax 或减小 XWmin，使得 Sx 变小，因而不会丢失已选取的场景

2）若 Sx ＜ Sy，则取 Sx，即可以通过增加 YWmax 或减小 YWmin，使得 Sy 变小，因而不会丢失已选取的场景

（要点：选取相等的缩放系数 Sx 和 Sy。改变的窗口不能丢失原场景）

2. #5.16

作业 7

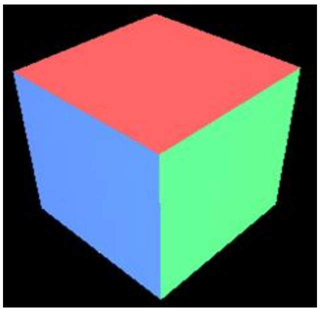
1. 画一个材质是白色的 cube，再定义三个点光源，分别是红色光、绿色光、蓝色光，光源位置在 cube 的三个可见面的前方。用 fragment shader 编程实验：

1） 只定义漫反射光

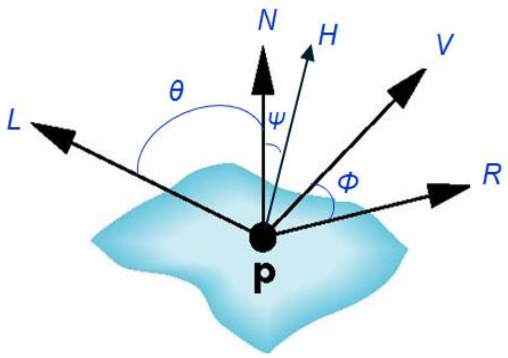
2） 只定义漫反射光和镜面反射光

3） 定义漫反射光、镜面反射光，以及环境光

比较三种情况的结果，并讨论原因。



作业 2. 教材#6.7 题。



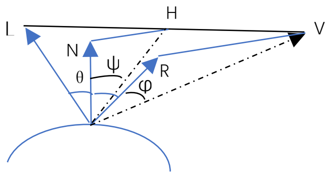
答：当 L,N,V,R 共面时，

因为<LN=<NR 且<LH=<HV，

所以<LV=2θ-ϕ=2(θ－ψ)

ϕ=2ψ

不共面时，L,N,R 是共面的，V 不在平面上，有ϕ<2ψ



作业 8

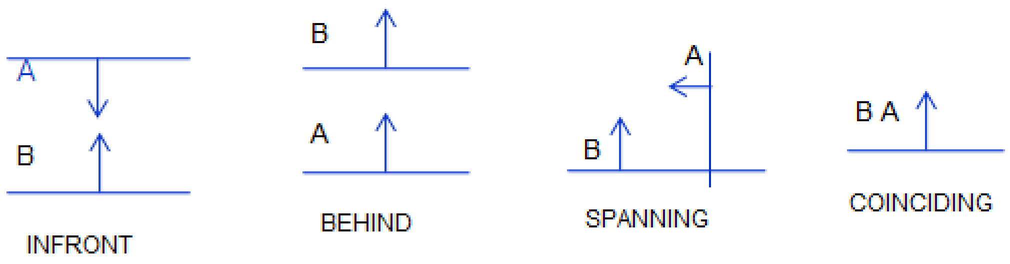
第七章练习 #7.2 题（在电影和电视中，汽车和四轮马车的轮子通常看起来沿错误的方向旋转，是什么原因导致这样的效果？能否修正这样的问题？）

答：这是时间域走样问题。当轮子旋转时，轮子转动的速度比显示器刷新的速度快时（Nyquist 速率，即超过了最小采样频率）的情况。例如，显示刷新速度为 30 帧/秒，即 0.033 秒显示一帧。轮子有 16 根辐条，两根之间是 22.5 度。设轮子顺时针旋转，当轮子的辐条在两帧之间（0.033 秒之内）转过了 360 度，感觉轮子没有转动。若辐条在两帧之间转角大于 15*22.5=337.5 度，即反时针转角小于 22.5 度，感觉是轮子反时针旋转。

一般 30 帧/秒，可以感觉是连续的画面，这是因为人眼的视觉残留

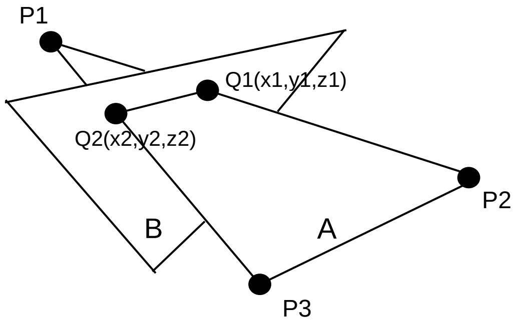
作业 9

1. 如何判断空间上两个三角形的位置关系？位置关系包括：三角形 A 在三角形 B 的前方；三角形 A 在三角形 B 的后方；两个三角形相交且有交线或交点；两个三角形共面。4 种情况



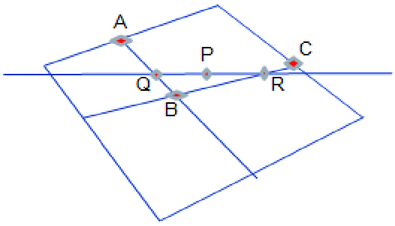
答：思想：判断三角形 A 的三个顶点 p1,p2,p3 与三角形 B 的位置关系：

1） 如果 Pi(i=1,2,3)都在三角形 B 的前方（即 B 的法向量方向），则 A 在 B 的前方
2） 如果 Pi(i=1,2,3)都在三角形 B 的后方（即 B 的法向量反方向），则 A 在 B 的后方
3） 如果 Pi(i=1,2,3)都在三角形 B 的平面上，则 A 与 B 共面
4） 如果不是以上三种情况，当 Pi(i=1,2,3)之一在三角形 B 上，则 A 与 B 有交点 Pi
当 Pi(i=1,2,3)之二在三角形 B 上，则 A 与 B 有交线 L=PiPj
当 Pi(i=1,2,3)分别在三角形 B 的前后方，则 A 与 B 可能有交线 L=Q1Q2，Q1 是 P1P2 与 B 平面的交点，Q2 是 P1P3 与 B 平面的交点（直线与平面求交点，见课件），再判断 Q1 和 Q2 是否在三角形 B 内部



（注：1. 这道题用到了课件上讲到的点与面的关系，线段与平面求交点，判断点是否在三角形面上等知识。关键是四种情况的讨论，对第 4 种情况可能有交点、交线或不相交。2.两个面的位置关系与视点无关，与两个面的法向量方向有关。3.可以用点到面的距离的符号判断点与面的关系，用点到面的距离判断共面的关系）

2. How to compute the interpolation illumination Ip2 by previous illumination Ip1 and increment.



$I_Q = (1-u) \times I_A + u \times I_B \qquad 0 \leq u \leq 1,\ u = AQ/AB$

$I_R = (1-w) \times I_B + w \times I_C \qquad 0 \leq w \leq 1,\ w = BR/BC$

$I_P = (1-t) \times I_Q + t \times I_R \qquad 0 \leq t \leq 1,\ t = QP/QR$

Increment Computing:

$I_{P2} = (1-t_2) \times I_Q + t_2 \times I_R$

$I_{P1} = (1-t_1) \times I_Q + t_1 \times I_R$

答：

$$I_{P2} = I_{P1} + (I_R - I_Q)(t_2 - t_1) = I_{P1} + \triangle I_{RQ} \times \triangle t$$