

Homework 02

- 姓名：郭灵
- 学号：3170105370
- 专业：计算机科学与技术
- 指导老师：张宏鑫

实现内容

- 使用显示列表进行绘制
- 实现地球，太阳，月亮，火星，木星
- 星球倾斜自转，即例如实现黄赤交角
- 星球公转，同时实现轨道平面存在交角
- 实现纹理映射，即星球表面绘制
- 实现以太阳为中心的光照模型
- 实现摄像机移动，镜头移动
 - **w**：摄像机上移
 - **s**：摄像机下移
 - **a**：摄像机右移
 - **d**：摄像机左移
 - **z**：摄像机前进
 - **c**：摄像机后退
 - **i**：镜头向上偏转
 - **k**：镜头向下偏转
 - **j**：镜头向左偏转
 - **l**：镜头向右偏转
- 实现暂停按键
 - 空格
 - **默认开始处于开始状态**

Detailed Computing Steps

每次绘制新的星球都使用 `glPushMatrix()` 和 `glPopMatrix()` 保存原始坐标，在新的坐标系内进行旋转移动，绘制完成后弹出原始的坐标系，因此，实际上的坐标变换可以分为两类

- 卫星和行星
- 太阳

行星、卫星

以火星的坐标变换为例：

- 使用 `OpenGL` 的默认坐标系，太阳系的平面基本位于 **X-Z** 平面
- 首先转动轨道平面产生夹角，此处令 **X-Y** 平面绕 **Z** 轴旋转，设原始坐标为 (x, y, z) ，转动夹角参数 θ ，对应的旋转矩阵 S_{orbit} ：

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

- 接下来平移星球至转动轨道，该轨道位于 **X-Z** 平面，设对应的半径为 R ，当前距离起始位置的旋转角度为 α ，对应的平移矩阵 T 为：

$$\begin{bmatrix} 1 & 0 & 0 & R\sin\alpha \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & R\cos\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

- 接下来旋转星球，使得地轴偏移垂直方向（由于粘贴了纹理，考虑到地轴并非一定垂直于公转平面，因此有必要修改地轴指向），原地轴平行于 **Y** 轴，需要与 **Y** 轴产生夹角，此处选择绕 **Z** 轴进行旋转，转动地轴偏角为 β ，对应的旋转矩阵 S_{self} 为：

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

- 接下来星球需要绕地轴进行自转，此处 `glPushMatrix()` 和 `glPopMatrix()` 在新的坐标系内操作，减轻计算量；新的坐标系内，只需令星球绕地轴旋转即可，对应的旋转矩阵：

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

- 完成该矩阵后调用显示列表进行绘制
- 一个特例是月球作为地球的卫星，在地球计算完成(4)后，**pop**矩阵后，做一次(3)旋转的逆运算（即反向旋转）后（其原因是修正月球的公转轴向，否则会绕地不规则运转）。继续调用 `glPushMatrix()` 和 `glPopMatrix()` 在新的坐标系内操作，重复(1)(2)矩阵的计算，进行正常绘制。

太阳

- 只需要进行(4)矩阵，做自旋操作即可

运行截图



