****

**计算机图形作业（二）报告**



**学 院 软件学院**

**专 业 软件工程**

**学 号 3017218135**

**姓 名 全康连**

目 录

目录

**摘要3**

项目名称3

操作简介3

**主体介绍3**

程序简介3

算法3

实现细节4

**结果讨论12**

结果12

讨论13

**收获和建议15**

课程收获15

项目开发收获16

课程建议16

一．摘要

1. 项目名称：Assignment1---Planet rotates around the sun
2. 操作简介
3. 自行设置行星的围绕太阳旋转的轨道，行星的大小和纹理
4. 添加键盘控制，例如右箭头控制行星绕x轴旋转的速度

二．主体介绍

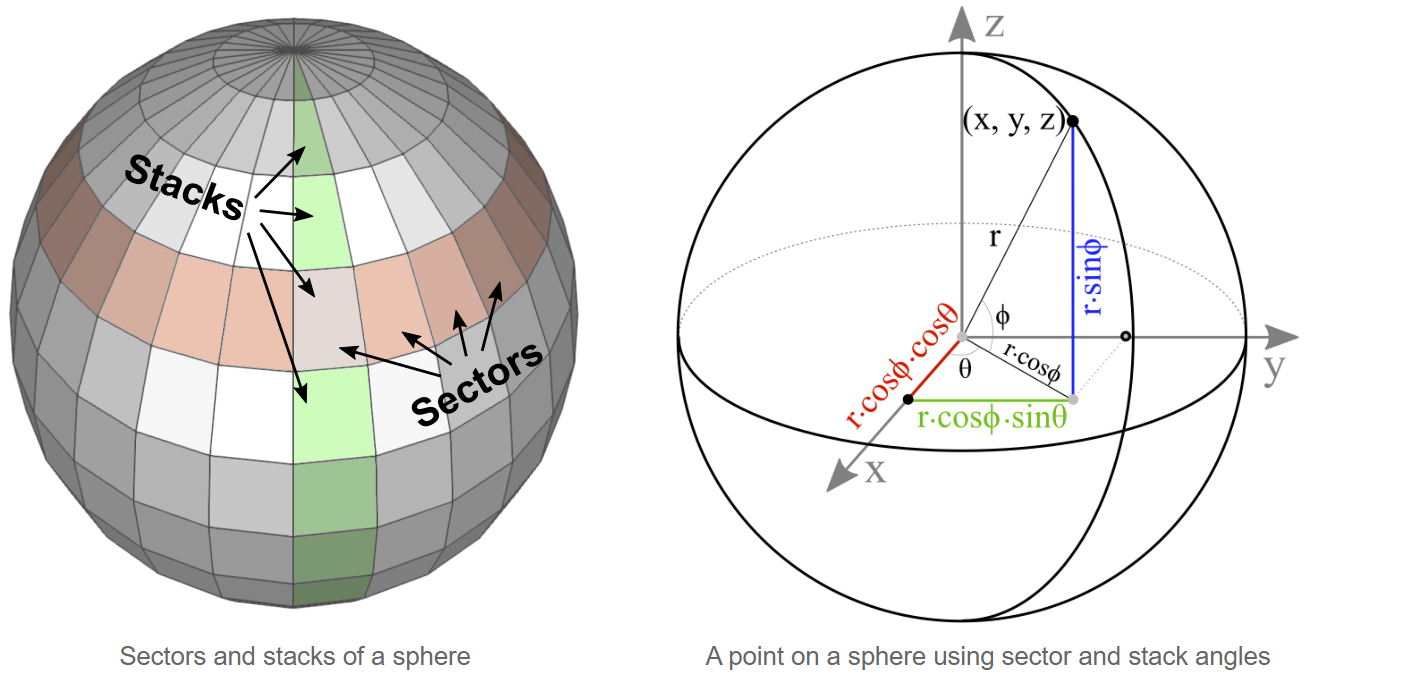
1. 程序简介

该程序完成了银河系八大行星围绕太阳旋转的场景设计，每个行星有其自身的轨道，大小和纹理样式；用户可通过键盘控制行星公转，自转的速度，也可将场景复原至最初样子，并恢复行星旋转

1. 算法
2. 球体绘画

程序设计场景中主要物体为3D球体，每一个行星需要进行各自的纹理贴图。球体的绘画算法在绘画球体的过程中将组成球体的顶点坐标，法向量，纹理坐标一并算出并存储，便于之后的光照添加和纹理贴图。

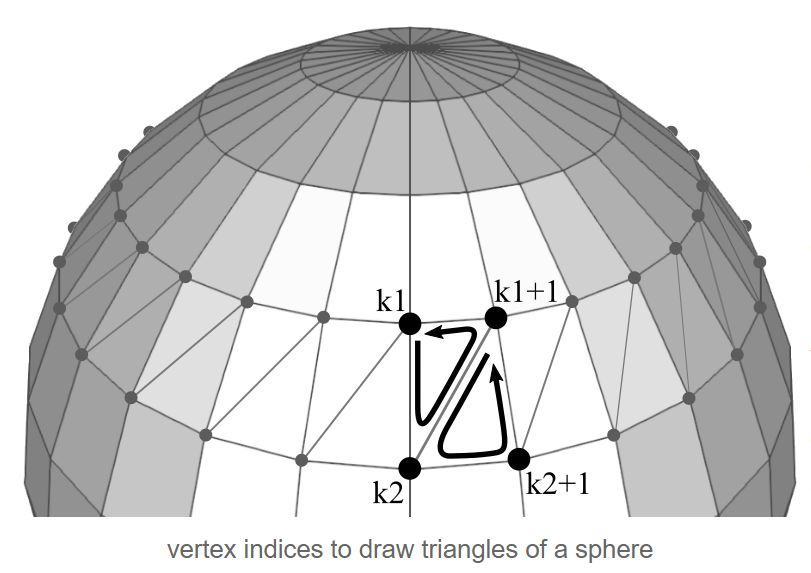
球体的三维表示为, 画球算法将球体按照经度和纬度进行等分操作，这样球面被分成多个四角面片；



其中任意一个点在球上的位置可以这样计算

经度(sector)的角度范围是0~360，纬度(stack)的角度范围是-90~90，所以画点时的前进角度可以计算如下，其中stackCount为纬度划分块数，sectorCount为经度划分块数

该算法算出的顶点为组成四角面片的四个顶点，但是openGL需要三角面片才能绘制球体，因此我们需要将每个四角面片分为两个三角面片(极点面片除外)，存储组成每个三角面片顶点的位置信息。

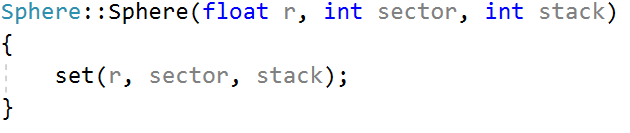


如图所示，当前stackCount为k1，则该组成该四角面片的三角形顶点索引为

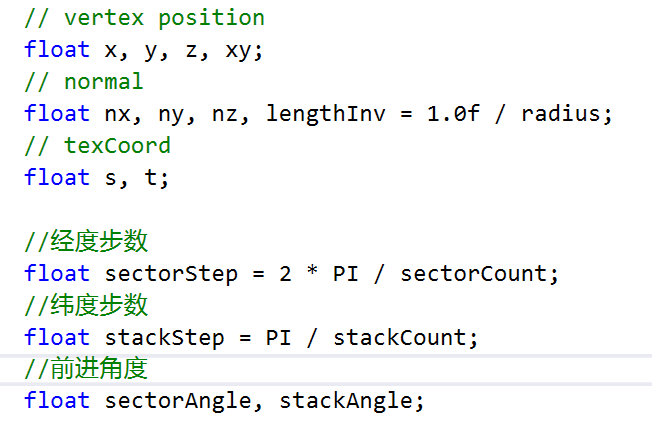
1. 实现细节
2. 球体绘画

程序自定义了Sphere.h和Sphere.cpp，在该文件中实现了球体的顶点位置计算，法向量计算和纹理坐标的计算

其中Sphere类构造方法如下，传入参数需要绘制圆的半径，经纬度等分块数



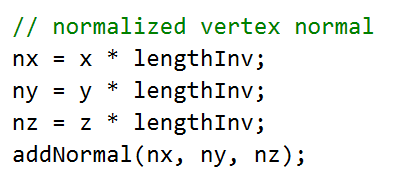
根据球体绘制算法，对顶点位置信息，法向量，和纹理坐标进行计算并将其存入顶点数据数组



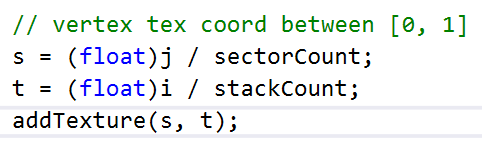
顶点位置信息



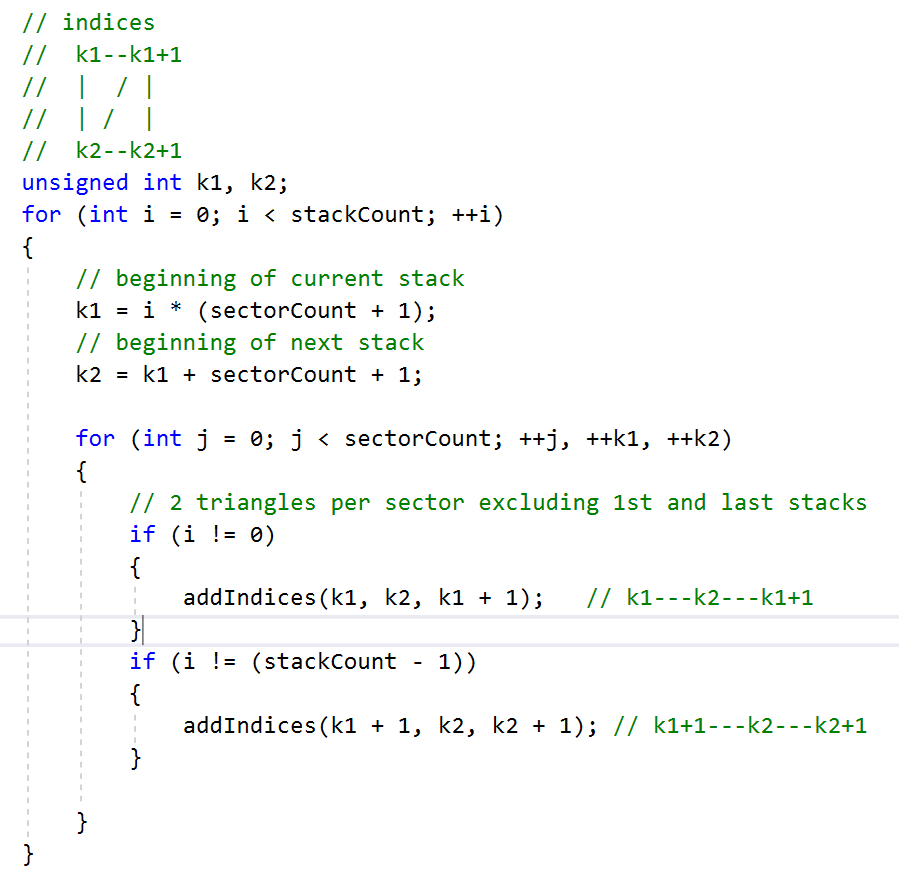
顶点法向量



纹理坐标



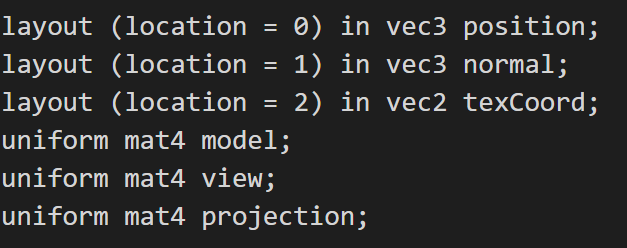
根据球体绘画算法，计算三角面片的顶点索引



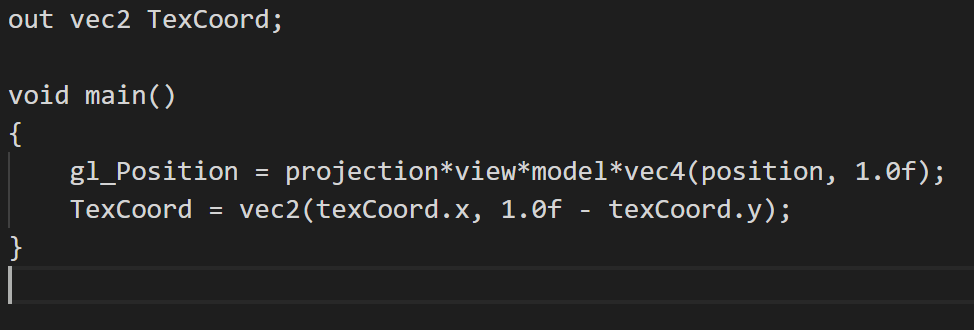
1. 定义main3.vert.glsl, main3.frag.glsl文件

Shader着色器中需要对物体进行纹理贴图，本程序并未对物体添加光照信息，

vertex着色器定义输入如下，定义顶点的位置position, 法向量normal，纹理坐标texcoord，模型变换矩阵model，模型视图矩阵view，投影变换矩阵projection



我们需要将纹理坐标输出到fragment着色器，因此定义输出如下，注意在处理纹理坐标时，程序引入了SOIL库对图片进行采样处理，该库使用y轴顶部作为原点，所以需要将纹理坐标的y轴数据进行倒转



1. 纹理贴图

程序对行星的表面样式进行了纹理贴图，需要进行一系列的设置步骤：

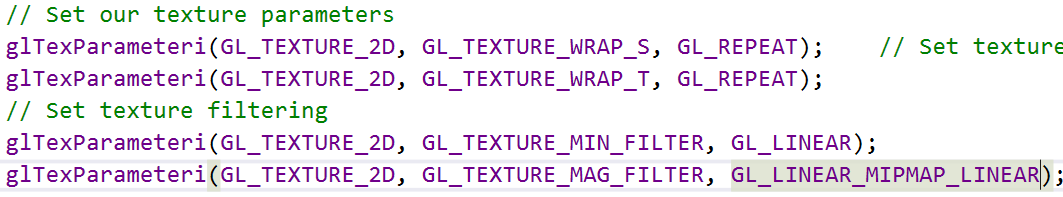
生成纹理：使用glGenTextures()函数生成纹理，需要指定纹理个数



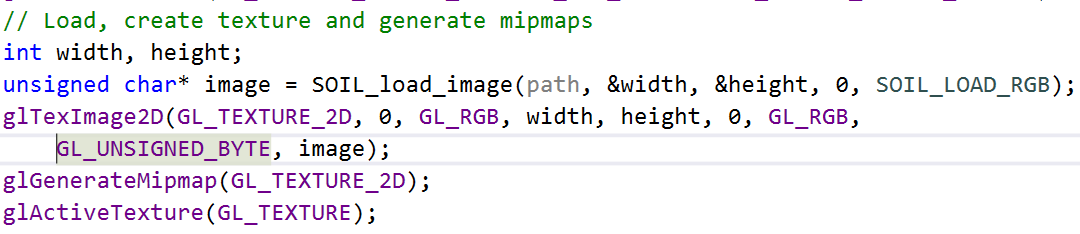
绑定纹理：使用glBindTexture()函数对纹理进行绑定



设置纹理参数：使用glTexParameteri()进行坐标轴环绕设置，纹理过滤设置；其中坐标轴环绕设置为重复纹理图样，纹理放大时设置为线性插值，缩小时设置为mipmap线性插值



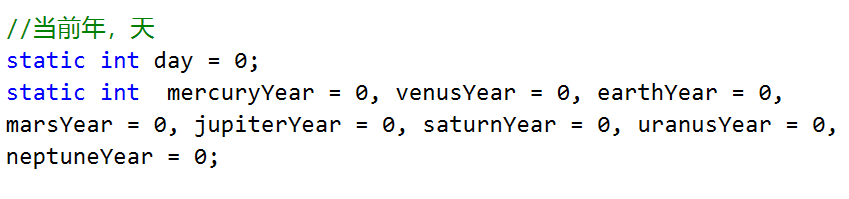
加载图片生成纹理采样：使用soil\_load\_image()加载图片；glTexImage2D()采样纹理；glGenerateMipmap()生成纹理的mipmap



1. 设置行星位置和环绕矩阵

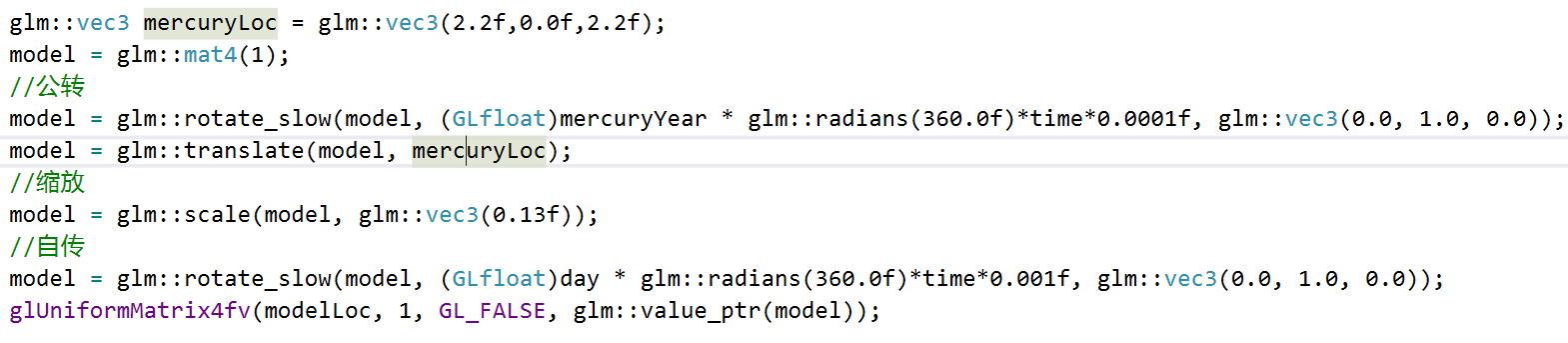
程序设计了八大行星水星，金星，地球，火星，木星，土星，天王星，海王星以太阳为中心进行公转，每个行星进行自传的场景；这需要对每个行星的位置，旋转速度进行设置

程序定义了每个行星的旋转年份和天数，通过键盘控制行星的公转和自传速度；当用户操作键盘指定按键时，行星的年份和月份会改变，并通过模型变换矩阵model改变当前行星的公转和自传速度



通过设置模型变换矩阵model（将局部坐标变换到世界坐标）来达到行星先围绕太阳旋转再自传的效果；八大行星按照顺序绘制

水星如下，



其中参数意义如下：

mercuryLoc: 水星的坐标

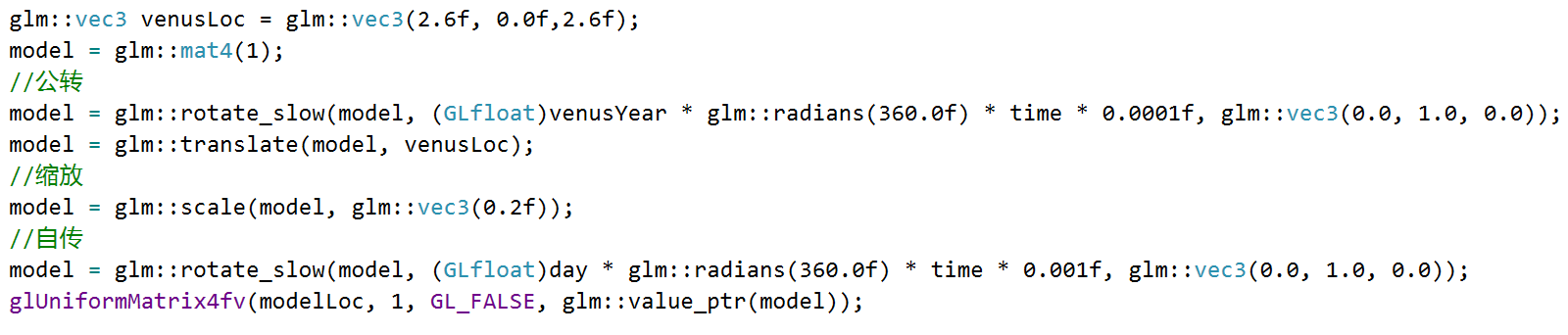
第一次rotate\_slow: 先对水星金星旋转后，再使用定义的水星坐标对其进行平移，则可以达到围绕太阳公转的效果；

Scale: 对水星的大小进行缩放，与太阳形成明显对比

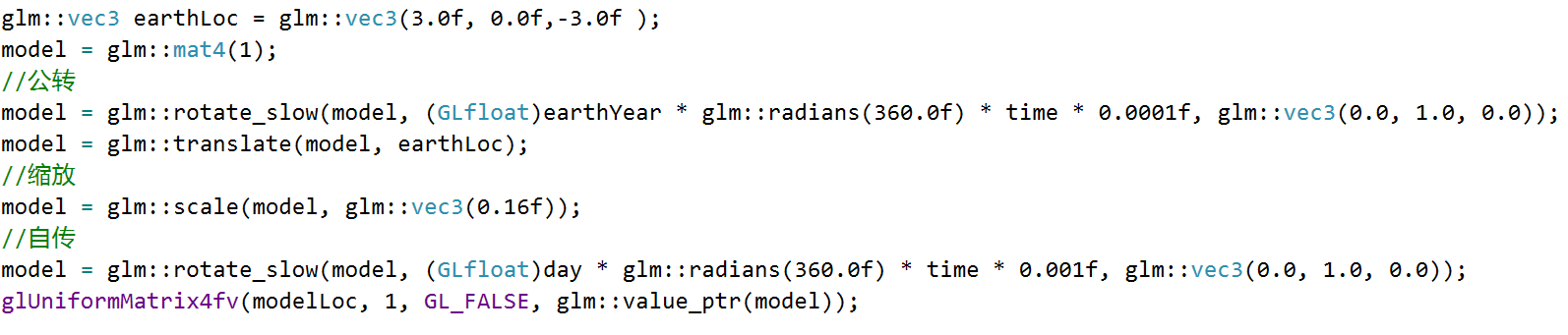
第二次rotate\_slow: 在以上的基础上对水星金星旋转操作，使得到达水星自转的效果

剩余星球的操作和水星的操作步骤相同，但每个星球的坐标和缩放比例不相同，公转和自转的速度也有所不同，通过年月设置

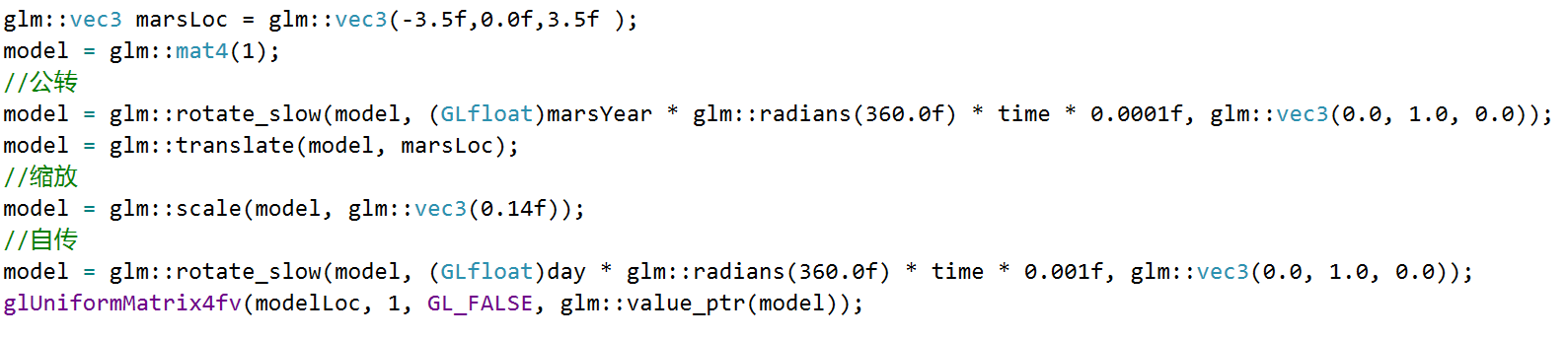
金星如下：



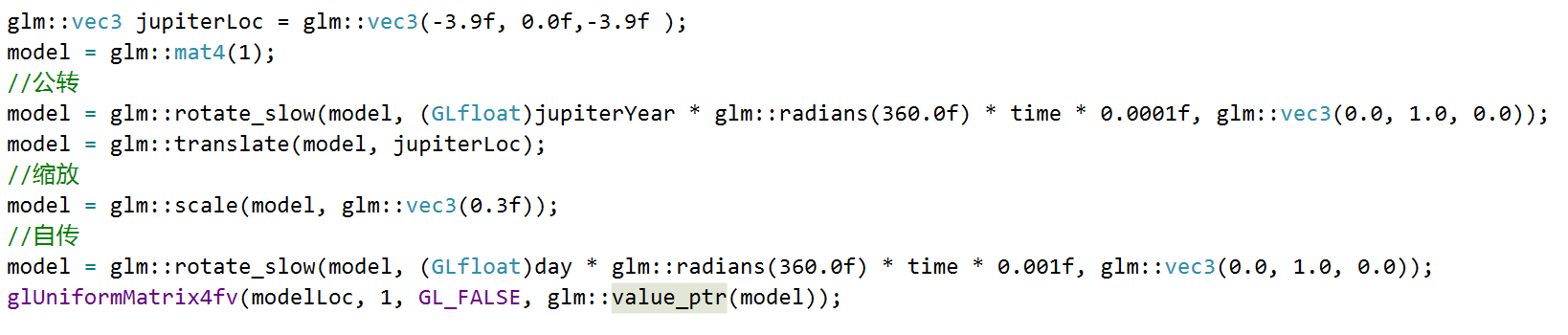
地球如下：



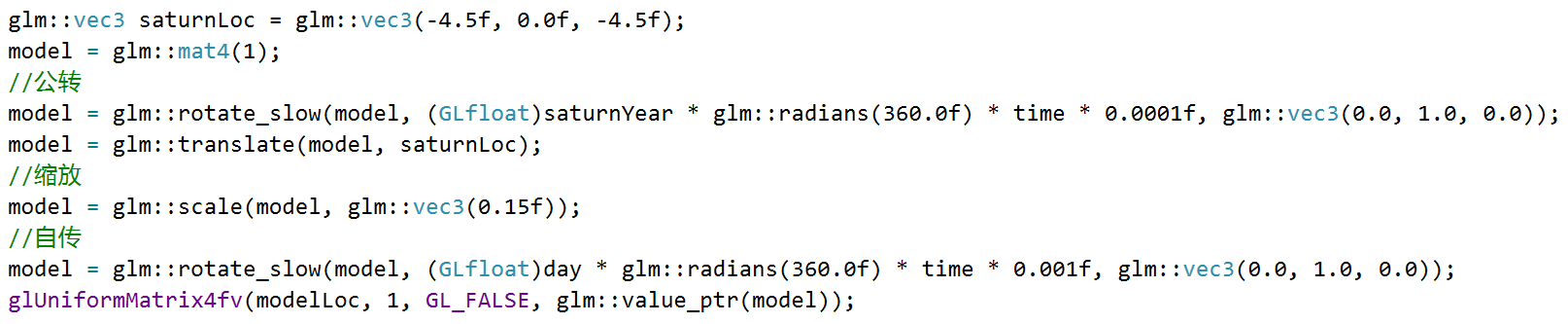
火星如下：



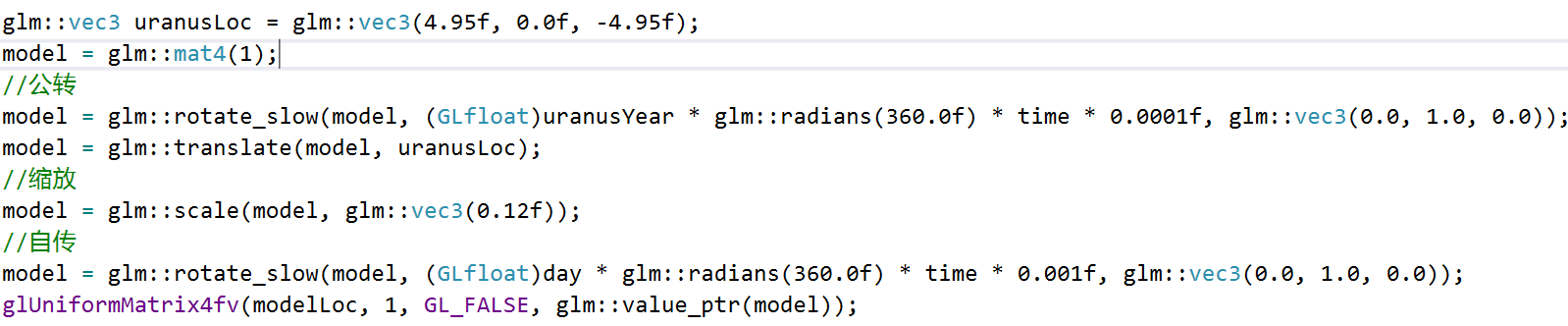
木星如下：



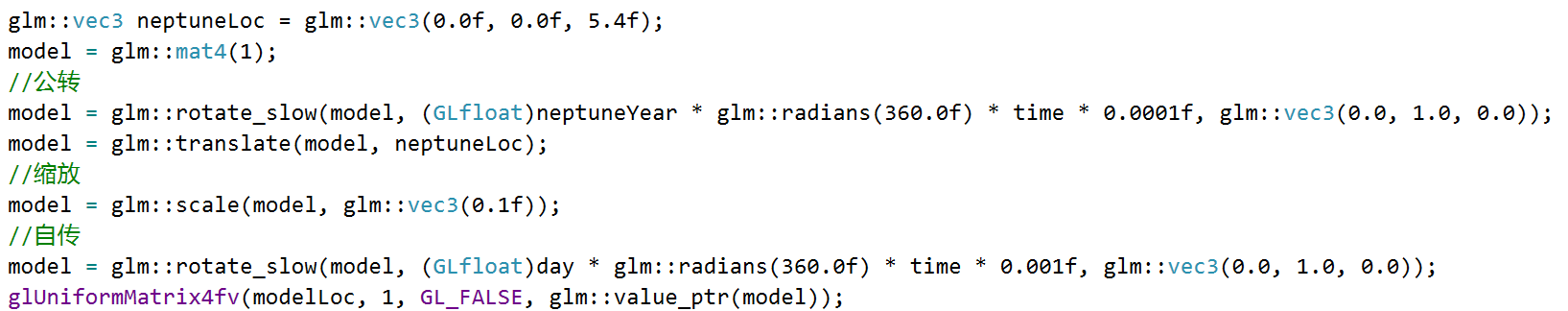
土星如下：



天王星如下：



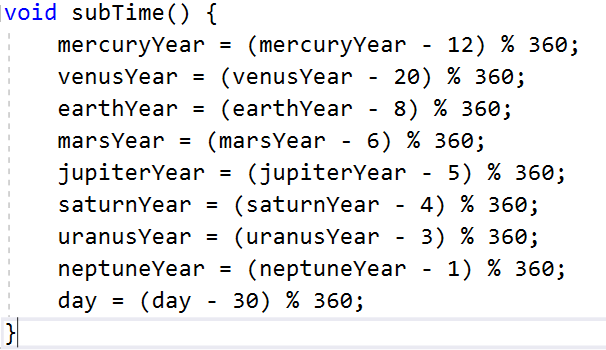
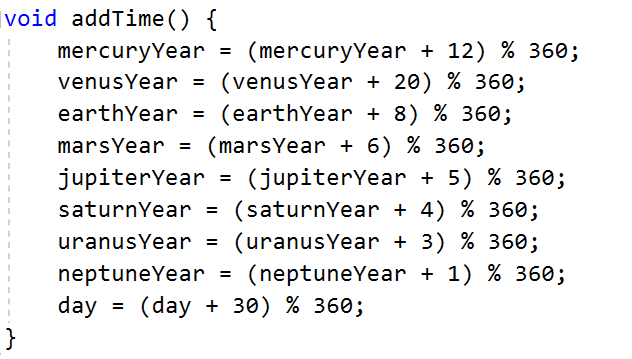
海王星如下：



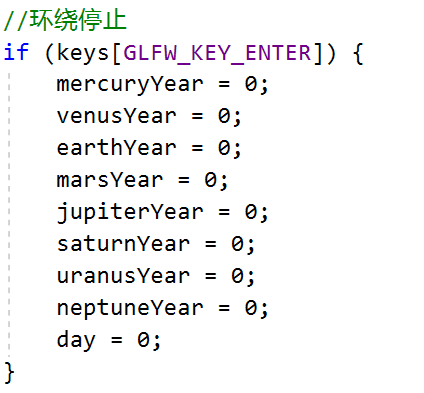
1. 添加键盘控制

本程序添加了用户可以通过指定按键来控制场景中行星环绕的速度功能。

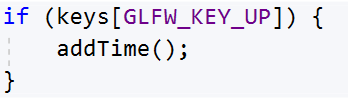
函数addTime(), subTime()在用户按下指定按键时被调用，增加和减少行星的当前年月，使得公转和自传速度发生变化



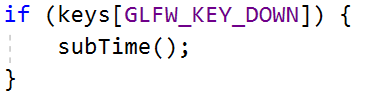
按键enter：用户按下此键，行星停止转动，并恢复至原渲染初场景



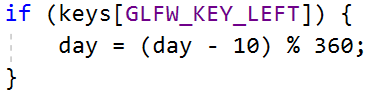
按键upArrow：用户按下此键，可以增加行星的当前年份和天数



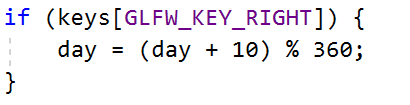
按键downArrow：用户按下此键，可以减少行星的当前年份和天数



按键leftArrow：用户通过此键较少自传天数



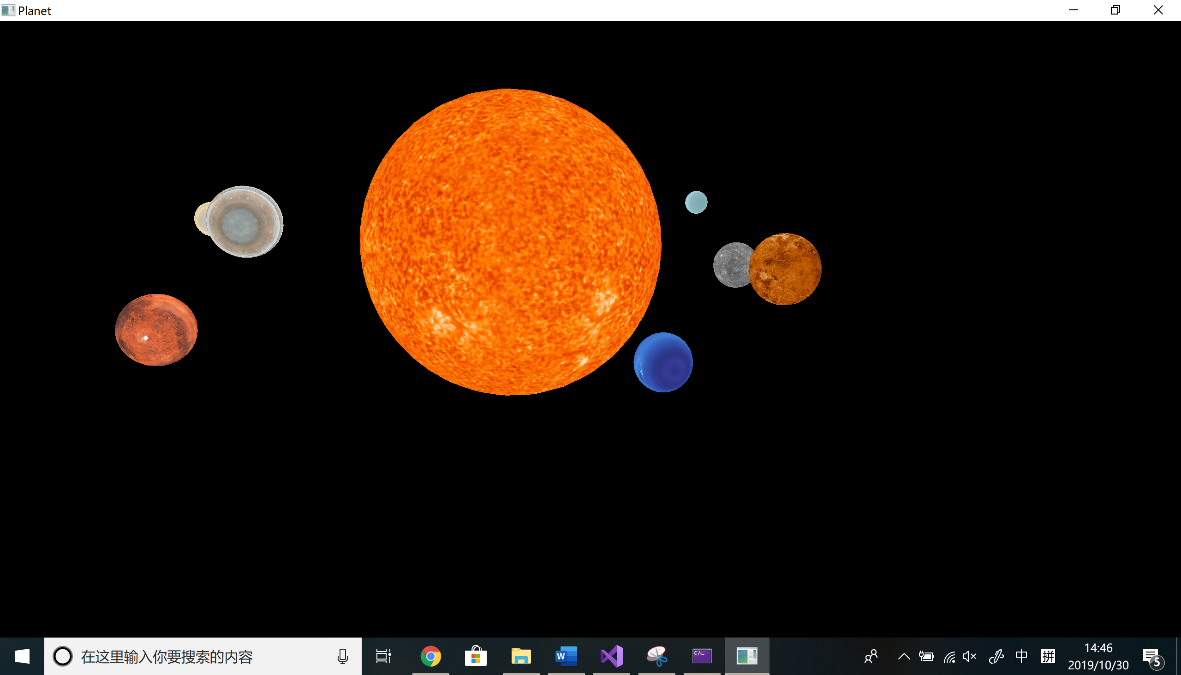
按键rightArrow：用户通过此键增加自传天数

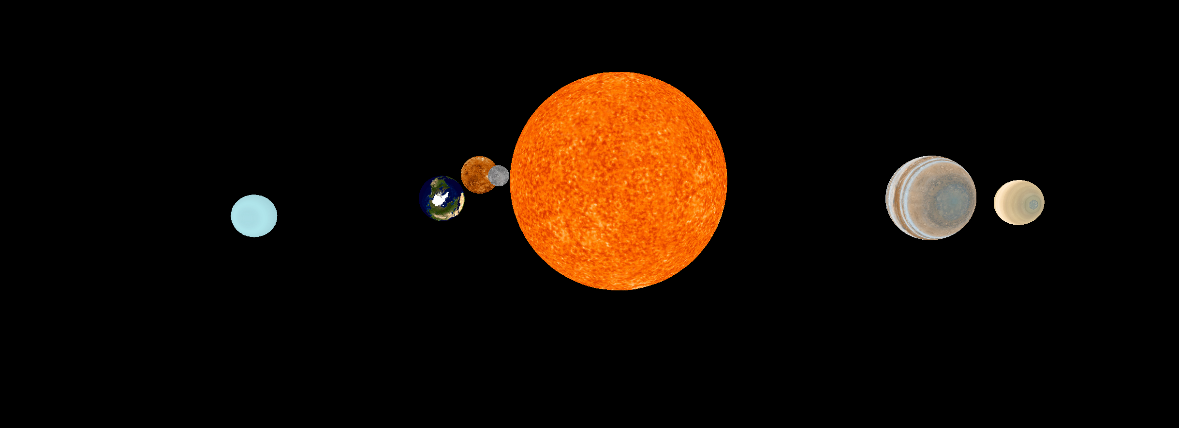


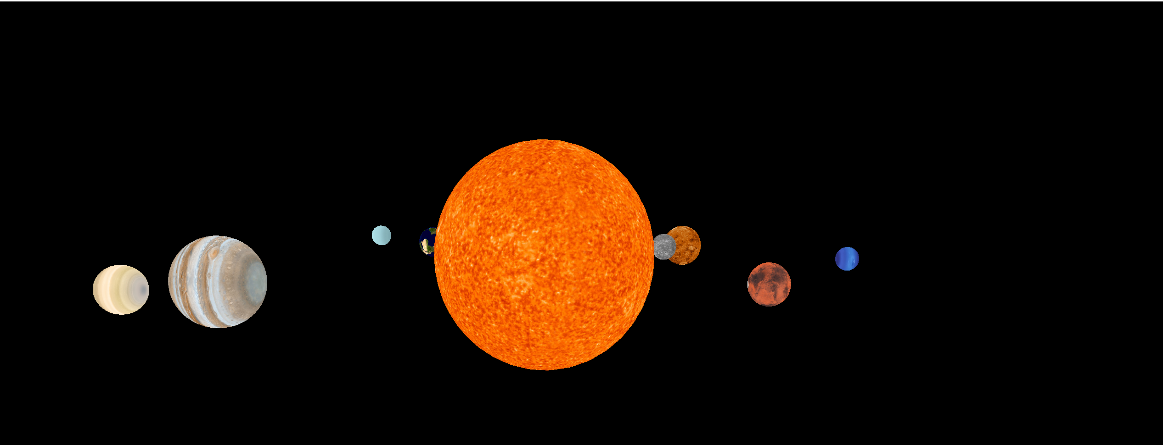
三．结果讨论

1. 结果

运行程序，可以看到原始行星环绕系统如下







通过键盘和鼠标移动的配合，可以实现行星的公转，自传速度，场景的移动（点击图片进行视频播放）



1. 讨论

在完成程序的过程中遇到了一些问题，通过上网查询资料，参考他人代码，对问题有了自己的理解和解决方案。

1. 如何绘制球体

一开始的想法是从网上下载球体模型并将模型导入，但考虑到模型需要进行贴图，下载模型的纹理坐标不能满足程序需求，因此决定自己进行球体的绘制

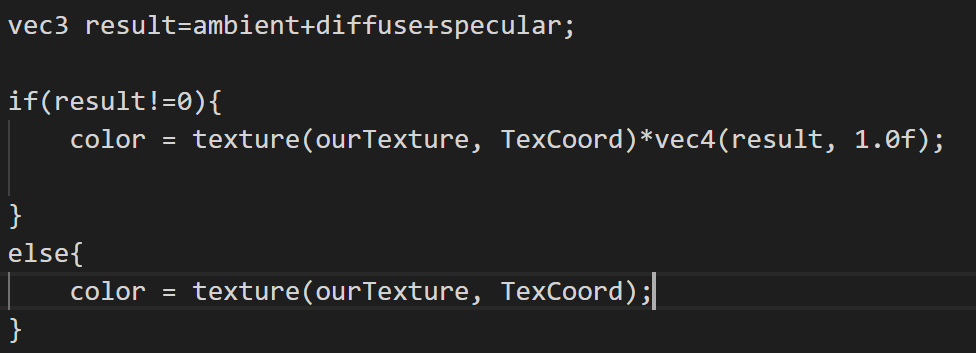
球体的绘制不能将球面上所有点遍历，但是可以通过对球体的层次划分来增加绘制顶点的个数，划分越多，球体模型越平滑。球体的绘制即把整个球体一层一层划分，使用旋转角来遍历组成球体的顶点；在绘画顶点的过程中，可以将每个顶点的法向量和纹理坐标一并算出，为之后的纹理贴图提供数据支持；（注意纹理坐标应该将其处理在0~1之间）。详细讲解参见“算法->球体绘制”，参考教程

<http://www.songho.ca/opengl/gl_sphere.html>

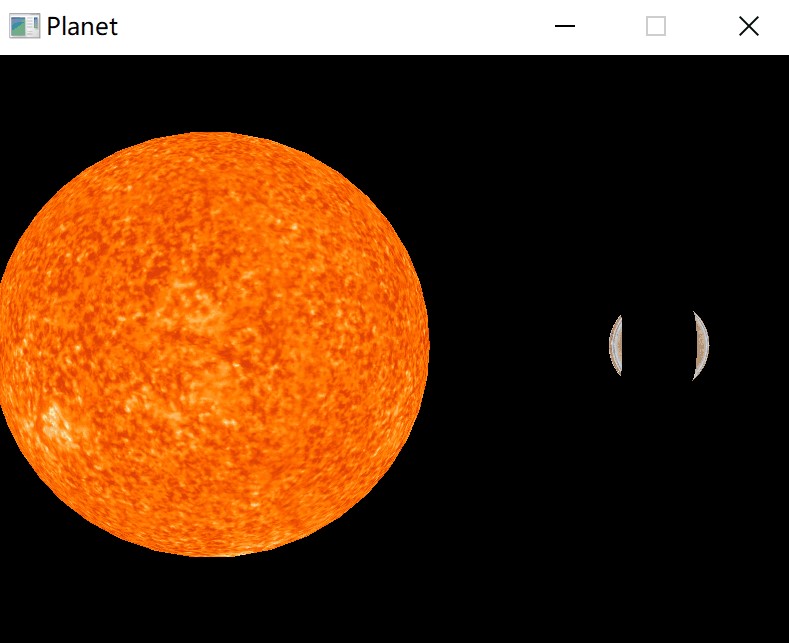
1. 纹理贴图问题

在进行纹理贴图的过程中，采用的方法是二维纹理贴图的方法，即将球体的三维坐标投影映射到纹理图像坐标范围之间，然后对图像进行采样并将其贴在球体上

一开始程序想要将光照和纹理相结合，glsl文件中fragment着色器中这样定义物体输出颜色



程序中只对太阳设置光照，其他星球并未设置；在绘制水星时，对模型变换矩阵model进行平移操作，但星球纹理出现黑带



但与其他同学的交流发现他们的平移旋转和缩放并未出现此错误，于是经过思考和将glsl文件与同学进行比较，得出结论如下：

顶点的法向量定义如下：



在对光照信息进行条件判断时，对model矩阵的缩放没有改变顶点的法向量，平移之后顶点的法向量收到影响；此时即使没有对星球的材质进行设置，但条件判断中的result并不为0，所有造成了星球的纹理黑带；于是对glsl文件进行更正，不要光照信息，直接使得着色器输出颜色为采样纹理，便解决了黑带问题

1. 行星公转旋转设置

要模拟出行星环绕太阳公转自转的效果，需要对物体的模型变换矩阵model进行相应设置；设置过程中要注意旋转和平移的前后顺序

对于星球围绕太阳的公转：星球在固定轨道上围绕太阳金星旋转，则应该先设置公转参数再进行平移，既在世界坐标系绕y轴旋转的基础上对星球进行固定位置的平移，达到行星在固定轨道上公转的效果

对于星球的自传：应该先设置平移再进行旋转，即在平移后固定的位置上，星球绕自己的局部坐标系y轴进行旋转，达到星球在固定位置上自转的效果

每个行星的旋转速度和轨道并不一样，通过对不同行星的model矩阵进行相应设置，设置每个行星的大小，公转轨道和公转速度，并通过键盘上指定按键控制行星的旋转速度

四．收获和建议

1. 课程收获

计算机图形学课程中，老师讲解清晰，实验课中代码原理实现细节讲解清楚。在课程中，老师会将实验中的基本实现步骤给出，对于本次作业有着很大的帮助。通过对老师的实验指导学习和本次实验的完成，我学习到了很多知识。

* 纹理贴图

本次程序使用二维纹理贴图的方式对行星设置纹理样式；在对行星进行绘制时计算其二维纹理坐标（0~1之间）；通过openGL提供的纹理采样函数读取球体的每个顶点的二维纹理坐标，在相应纹理图像上进行纹理采样，并对三角面片设置纹理样式

纹理贴图的过程有一定的步骤，大致过程为生成纹理—绑定纹理—设置纹理参数—加载图片采样纹理—释放资源；其中设置纹理参数时主要设置纹理环绕（纹理坐标超出0~1）的方式，和纹理过滤（纹理进行放大和缩小时的采样）的方式，设置时有多种选项，不同的选项的效果大不相同。通过对参数的选择并运行程序，采取了效果最理想的参数，并理解了纹理映射的数学原理

1. 项目开发收获

通过本次实验，学习了球体的绘制过程，模型变换矩阵的多种操作，并将原理知识结合到实验中完成了程序要求的功能；在实验过程中收获颇丰

* 球体的绘制

程序中定义了Sphere类进行球体的绘制，使用球体绘制的算法将球体分层绘制；并在绘制的过程中记录顶点的位置信息，法向量信息，纹理信息，为后续的纹理贴图提供数据支持

* 行星旋转矩阵

程序通过对每个行星的模型变换矩阵model进行缩放，平移和旋转设置，来模拟出银河系八大行星环绕太阳公转并自转的场景。模型变换矩阵model将物体从局部坐标系变换到世界坐标系，平移，缩放和旋转的设置需要注意其顺序，不同的顺序使得矩阵变换的结果不相同，最终呈现的场景效果也不同

程序对键盘按键进行监控，记录用户的按键状态，并使用键盘上的指定按键对行星的公转，自转速度进行相应设置，达到场景变换的效果

* 纹理黑带错误解决

对于初始光照和纹理结合的着色器设定，并没有考虑到对模型变换矩阵model平移，缩放和旋转操作产生的对顶点法向量的影响；条件判断不全面，从而产生了纹理采样错误，黑带出现的问题。此过程中再次了解了模型变换矩阵的作用和对物体顶点数据产生的影响，并对phong光照模型的计算方法有了更深刻的认识

1. 课程建议

* 希望每一次实验老师可以选取比较好的作品进行展示和讲解，这样可以学习他人好的代码思维，了解到更多的计算机图形学知识
* 希望老师对于下一节课的内容可以提前告知所讲范围，方便学生的预习