Homework 5 - Camera

ddl: 4.17 周三 23:59前

Introduction

> 在上一次作业里，大家已经能对物体进行自由变换。本次作业大家将会学习通过投影使物体更接近真实，以及通过视角变换让观察的视角更灵活。最后的Bonus是实现一个camera类，这个camera类将会在后面的作业或者group project里经常用到！

作业严禁抄袭，被发现者当次作业0分！

References

1. 空间坐标系统

CG流水线涉及到多个空间坐标系统，上次作业的物体变换，以及这次作业的视角变换与投影，就已经设计CG世界里最重要的几个空间坐标系统——物体空间(Object Space)、世界空间(World Space)、观察空间(View Space)、裁剪空间(Clip Space)，各个空间对应自身的坐标系。详细说明以及整体流程图查看下方参考链接：

- 坐标系统

2. Projections投影

将三维空间中的物体投影到显示器的二维平面，如何去投影将直接影响到我们看到的结果。在OpenGL中可以使用两种投影：`perspective projection`和`orthographic projection`。

3. 视角移动与欧拉角(Euler Angle)

本次作业的加分项要求实现一个摄像机类，如何通过鼠标来控制摄像机的方向是一个重点。目前主要有欧拉角和四元数两种方法。四元数法要复杂很多，但效果相比欧拉角法要好，大家可以自己查找资料，了解四元数。本次作业大家用欧拉角就好（有能力的同学用四元数也可以，注意在实验报告中注明，可额外加分）。

欧拉角是可以表示3D空间中任何旋转的3个值，一共分3种：俯仰角(Pitch)、偏航角(Yaw)和滚转角(Roll)。由于本次作业的实现是鼠标控制视角变换，而鼠标在屏幕上的移动只有x、y方向两个自由度，也就是说大家的实现只需关注pitch和yaw两个量即可。

- 欧拉角

Homework

Basic:

1. 投影(Projection):

    - 把上次作业绘制的cube放置在(-1.5, 0.5, -1.5)位置，要求6个面颜色不一致
    - 正交投影(orthographic projection)：实现正交投影，使用多组(left, right, bottom, top, near, far)参数，比较结果差异
    - 透视投影(perspective projection)：实现透视投影，使用多组参数，比较结果差异
2. 视角变换(View Changing):
    - 把cube放置在(0, 0, 0)处，做透视投影，使摄像机围绕cube旋转，并且时刻看着cube中心
3. 在GUI里添加菜单栏，可以选择各种功能。 *Hint:* 使摄像机一直处于一个圆的位置，可以参考以下公式：

```
camPosX=sin(clock()/1000.0)*Radius;
camPosZ=cos(clock()/1000.0)*Radius;
```

原理很容易理解，由于圆的公式`a^2+b^2=1`，以及有`sin(x)^2+cos(x)^2=1`，所以能保证摄像机在XoZ平面的一个圆上。

4. 在现实生活中，我们一般将摄像机摆放的空间**View matrix**和被拍摄的物体摆设的空间**Model matrix**分开，但是在OpenGL中却将两个合二为一设为**ModelView matrix**，通过上面的作业启发，你认为是为什么呢？在报告中写入。（Hints：你可能有不止一个摄像机）

Bonus:

1. 实现一个camera类，当键盘输入`w,a,s,d`，能够前后左右移动；当移动鼠标，能够视角移动("look around")，即类似FPS(First Person Shooting)的游戏场景

Hint: camera类的头文件可以参考如下（同样也可以自己定义，只要功能相符即可）：

```
class Camera{
public:
    ...
    void moveForward(GLfloat const distance);
    void moveBack(GLfloat const distance);
    void moveRight(GLfloat const distance);
    void moveLeft(GLfloat const distance);
    ...
    void rotate(GLfloat const pitch, GLfloat const yaw);
    ...
private:
    ...
    GLfloat pfov,pratio,pnear,pfar;
    GLfloat cameraPosX,cameraPosY,cameraPosZ;
    GLfloat cameraFrontX,cameraFrontY,cameraFrontZ;
    GLfloat cameraRightX,cameraRightY,cameraRightZ;
    GLfloat cameraUpX,cameraUpY,cameraUpZ;
    ...
};
```

PS. `void rotate(GLfloat const pitch, GLfloat const yaw)`里的`pitch`、`yaw`均为欧拉角（参考上方References）

作业要求：

1. 把运行结果截图贴到报告里，并回答作业里提出的问题。
2. 报告里简要说明实现思路，以及主要function/algorithm的解释。
3. 虽然learnopengl教程网站有很多现成的代码，但是希望大家全部手打，而不是直接copy。