正交投影的参数

(left, right, bottom, top, near, far)，前两个参数指定了平截头体的左右坐标，第三和第四参数指定了平截头体的底部和顶部。通过这四个参数我们定义了近平面和远平面的大小，然后第五和第六个参数则定义了近平面和远平面的距离。这个投影矩阵会将处于这些x，y，z值范围内的坐标变换为标准化设备坐标。

透视投影的参数

(fov, width/height, near, far)，它的第一个参数定义了fov的值，它表示的是视野(Field of View)，并且设置了观察空间的大小。第二个参数设置了宽高比，由视口的宽除以高所得。第三和第四个参数设置了平截头体的近和远平面。所有在近平面和远平面内且处于平截头体内的顶点都会被渲染。

实现思路

camera类：

- 使用position，worldUp，yaw和pitch作为构造函数参数。
- 通过yaw和pitch计算出front方向， 通过front和worldUp计算出right和up方向。
- glm::lookAt(Position, Position + Front, Up)返回view矩阵。

- moveForward， moveBack， moveRight，moveLeft方法只需改变camera的Position。
- rotate方法改变camera的Yaw和Pitch，再重新计算front，right，up。

正交投影和透视投影多参数设置：

使用Imgui，根据gui的输入选择使用正交投影矩阵或透视投影矩阵，矩阵参数也由gui输入决定。

视角变换：

```
float camX = sin(glfwGetTime()) * 20;
float camZ = cos(glfwGetTime()) * 20;
view = glm::lookAt(glm::vec3(camX, 0.0, camZ), glm::vec3(0.0, 0.0, 0.0), glm::vec3(0.0,
1.0, 0.0));
```

使用上述代码使得camera在y = 0的XoZ平面上半径为20的圆上进行旋转，目标为原点。