OpenGL 开发

Kevin Dai

2024

**目录**

[1. OpenGL 相关库的选用: 2](#_Toc177853340)

[2. 项目的配置: 3](#_Toc177853341)

[3. OpenGL 开发: 4](#_Toc177853342)

[4. 数学基础-向量: 5](#_Toc177853343)

[5. 数学基础-矩阵 7](#_Toc177853344)

## OpenGL 相关库的选用:

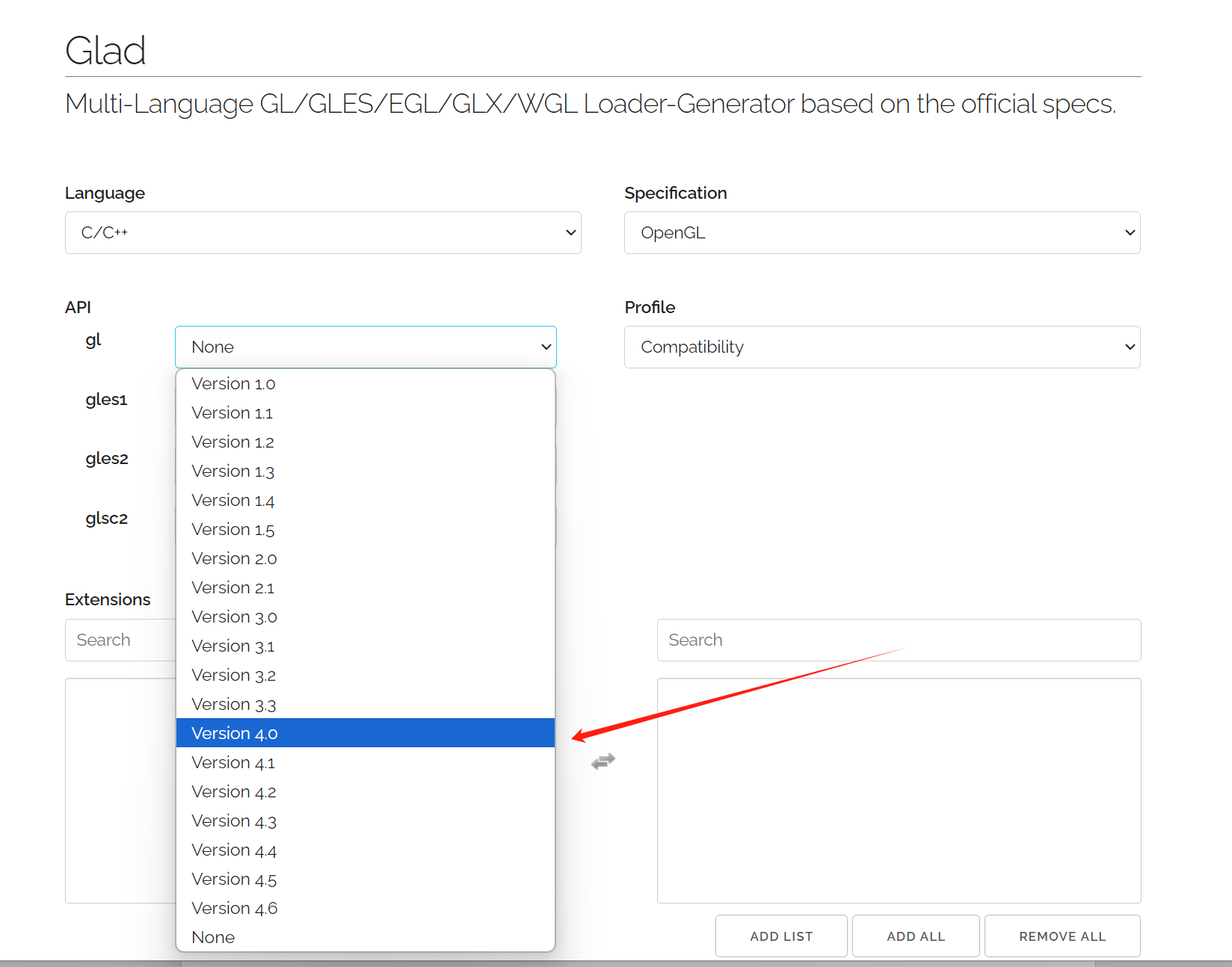
* GLFW是跨平台的(Windows, Linux) OpenGL Wrapper 库.使用较为便利,可以将其列为OpenGL在Windows上开发首选库.

下载地址: https://www.glfw.org/

* GLAD 是OpenGL的Helper库, 因为OpenGL是一种开放图形标准, 但不同硬件(AMD/Nvidia)厂家的实现与调优是不同的，所以在编译与使用时就需要使用GLAD等Helper库来帮助GLFW更好的适配OPENGL, 使得代码的编写/编译更为便利.

下载地址: <https://glad.dav1d.de/>

下载前需要选择所适配的OpenGL版本(选用不要低于3.0的版本,版本越新对新硬件的兼容性越好)



* GLM 是OpenGL的Helper库, 在OpenGL中需要使用到大量的数学矩阵运算用于对3D图形的变换(Transform)

下载地址: https://github.com/g-truc/glm

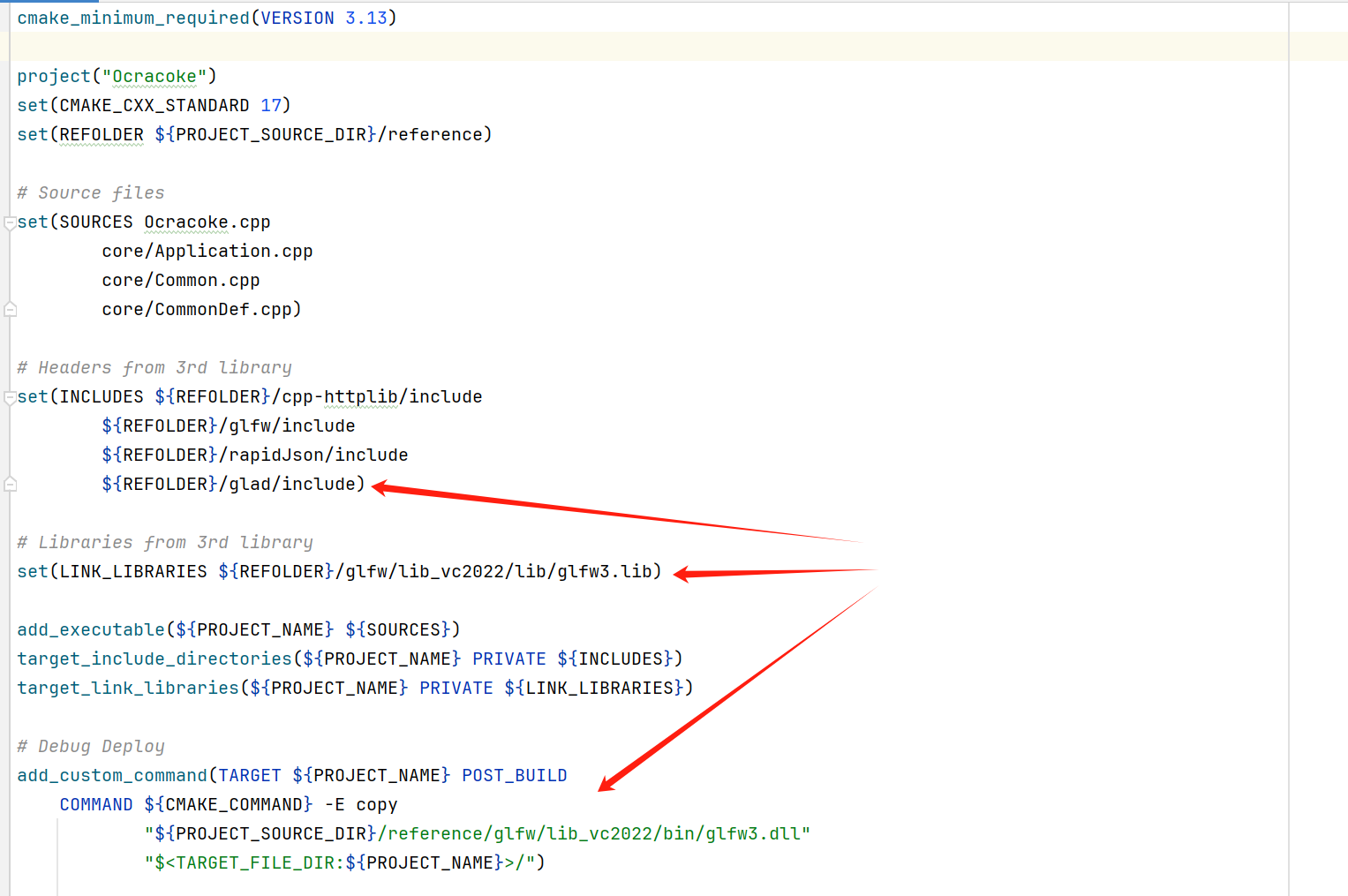
## 项目的配置:

* 关于MSBuild 的相关配置.

配置很简单, 不详细描述

* 关于CMake的相关配置

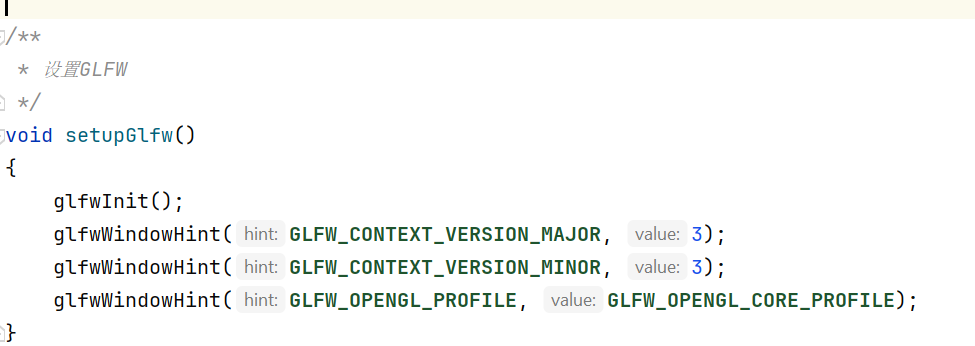
在CMakeList.txt文件中标记红色箭头位置是GLFW配置所需要包含的Header文件, 链接的静态库以及Build完成后对动态库复制的相关处理.



## OpenGL 开发:

* OpenGL的基本工作流程.
* OpenGL的初始化

在使用OpenGL前必须要对其进行初始化.



glfwInit() 是对整个GLFW OpenGL应用库的初始化.

GLFW\_CONTEXT\_VERSION\_MAJOR, 3表示OpenGL的正常使用版本为3.0及以上.

GLFW\_CONTEXT\_VERSION\_MINOR, 3表示OpenGL的最小使用版本为3.0.

GLFW\_OPENGL\_PROFILE, GLFW\_OPENGL\_CORE\_PROFILE 表示OpenGL的核心(标准)版本.

P.77

## 数学基础-向量:

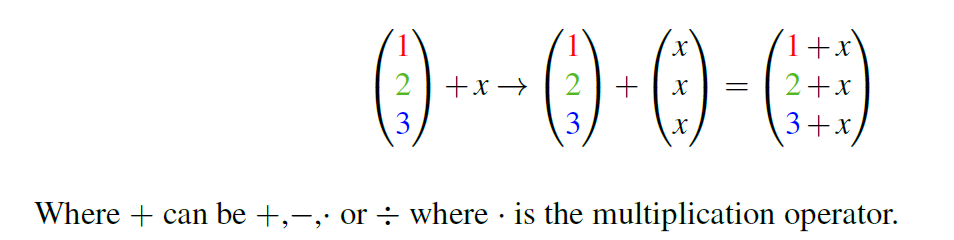
* 向量

向量有方向和量(长度),

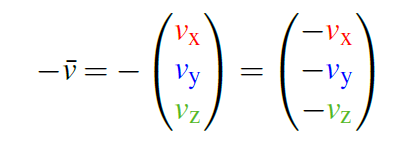
如v1=(3,4), 方向是x=3,y=4 量是5. 三维向量以此类推.

* 向量与单一数字的加减乘除

与普通数字的操作相同(各位数字相加减乘除)

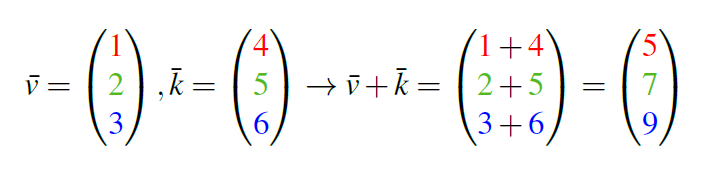


取反



* 向量与向量的加减

加法与普通数字操作相同(各位数字相加)

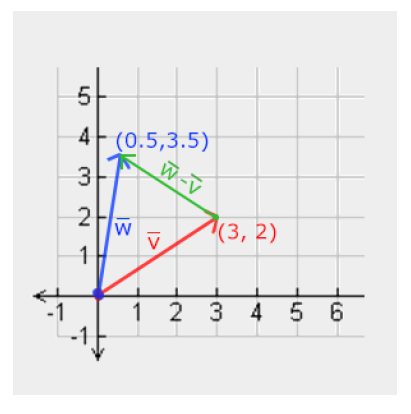


减法与普通数字操作相同(但需要注意图形上的表现)

w= (0.5,3.5)

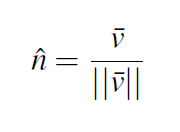
v = (3, 2)

w-v = (-2.5,1.5)



* 单位向量

长度是1的向量



* 向量与向量的乘法

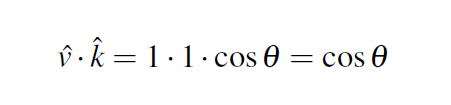
在OpenGL的使用领域需要特别注意两种向量乘法, 点积(数量积), 叉积(向量积, 叉乘)

点积可以用来确认两个单位向量是

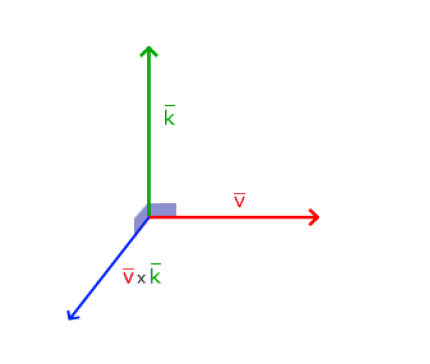
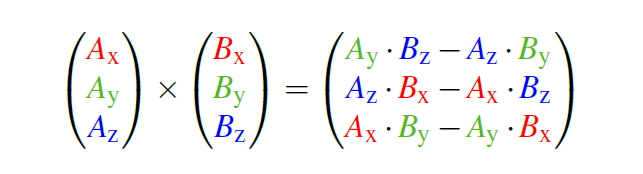
正交(90°, cos θ = 0),

交叉,

平行(0°, cos θ = 1),



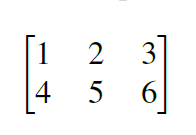
叉积可以在三维空间中获得两个三维向量的正交方向(二维向量)

## 数学基础-矩阵

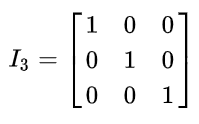
* 矩阵

由数组组成的数字阵列, 如2x3 的矩阵则为2行3列的数组

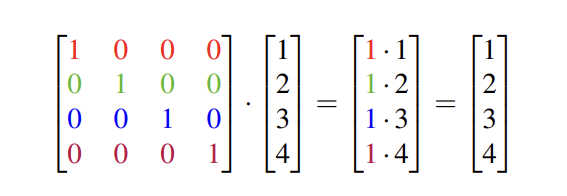


* 单位矩阵

矩阵的斜对角线都为1, 其他都为0的矩阵称为单位矩阵



单位矩阵不会对任何向量产生影响



* 系数矩阵与增广矩阵

例:

二元一次方程组

2X + 3Y = 7

4X - 2Y = 3

系数矩阵指的是将方程的系数提取出来的矩阵

则为:

[ 2 , 3

4 , -2 ]

增广矩阵指的是将方程的系数与常量结果提取出来的矩阵

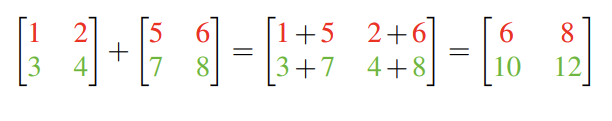
则为:

[ 2 , 3, 7

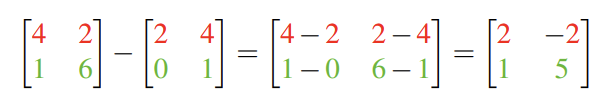
4 , -2 , 3]

* 矩阵与矩阵的加减

矩阵与矩阵的加法, 数字直接相加

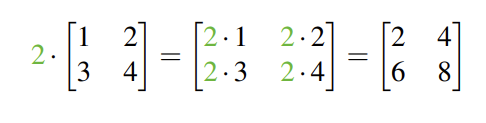


矩阵与矩阵的减法, 数字直接相减



* 矩阵与标量的乘法

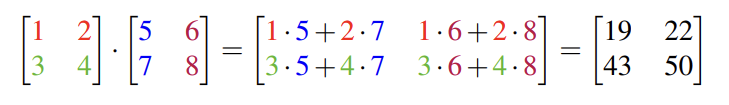
矩阵与标量的乘法, 数字直接相乘



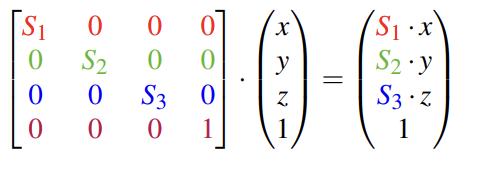
* 矩阵与矩阵的乘法

行列式乘法的口诀:

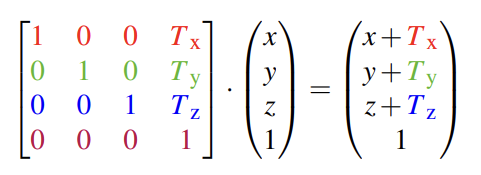
1. A1 \* A2 的行列数等于A1的行数, A2的列数, 也就是3行4列的A1 \* 4行3列的A2 其结果一定为3行3列.
2. A1 \* A2 中 A1的列数必须与A2的行数相同才可计算



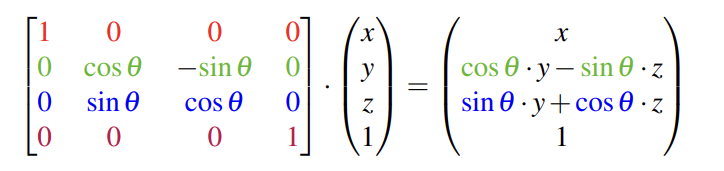
* 实用矩阵, 放大



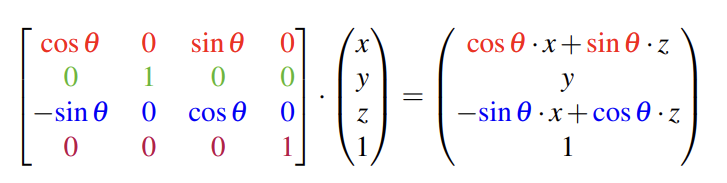
* 实用矩阵, 位移



* 实用矩阵, 绕X轴旋转(俯仰角 Pitch)



* 实用矩阵, 绕Y轴旋转(偏航角 Yaw)



* 实用矩阵, 绕Z轴旋转(翻转角 Roll)

