### Mac 配置OpenGL 环境

## 本文中 OpenGL 扩展库

* GLFW 渲染库的作用:将OpenGL 渲染到帧缓冲区的内容，绘制到屏幕上的一个窗口中
* GLEW 扩展库：OpenGL 核心中有一些内置的命令用来支持扩展，为了使用每一个现代命令，需要执行相当复杂的代码行，所以使用扩展库来处理这些细节已经成为标准做法，这样能让程序员可以直接用现代OpenGL 命令。
* OpenGL Mathematics 数学库：3D 图像变成大量使用了向量和矩阵代数。因此，配合一个支持常见数学计算任务的函数库护着类包，能极大的方便OpenGL 的使用。OpenGL Mathematics 简称 GLM。GLM 命令很方便地遵循和GLSL相同的命名惯例，使得来回于都特定应用程序的C++ 和GLSL 代码时更容易。GLM可以在官网下载。

## 准备并安装依赖库

我们首先安装 GLEW 和GLWE。安装这些库的最简单方法可能是使用 “Homebrew”工具， [Homebrew](https://links.jianshu.com/go?to=https://brew.sh/" \t "/Users/lihanxu/Documents\\x/_blank) 是一个软件包管理器，旨在让用户可能简单地在Mac 上安装常用的使用程序。在Mac 上的安装方法如下：

1. 打开终端；
2. 输入如下命令，然后回车，需要等待一段时间，翻墙会快很多

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

接下来，使用新安装的 Homebrew 来安装 GLEW和GLFE，步骤如下：

1. 在终端提示符下输入命令： brew install glfw3；
2. 在终端提示符下输入命令： brew install glew；
3. 请注意，/usr/local/include 路径下现在新增了两个文件夹（在苹果 M1 芯片系统中的路径为 /opt/homebrew/include），分别为 GL 和 GLFW.

接下来我们安装数学库 GLM。在4个库中，它的安装最简单，由于 GLM 是一个仅包含头文件的库，因此只需要如下步骤：

1. 下载 GLM 库，地址：[https://github.com/g-truc/glm/](https://links.jianshu.com/go?to=https://github.com/g-truc/glm/" \t "/Users/lihanxu/Documents\\x/_blank)；
2. 将文件夹中的 glm 文件夹及其内容复制到 /usr/local/include（在苹果 M1 芯片系统中的路径为 /opt/homebrew/include）目录下。

## 准备开发环境

开发环境使用 Xocde 集成IDE，如果你还没有安装Xcode ，那么需要进行安装一下，安装过程也很简单，去App Store 直接搜索 Xcode，然后下载（xcode 比较大，所以需要下载很久）。

安装 Xcode 成功以后，需要配置使其使用OpenGL 以及上述扩展库。一下是我们为 C++/OpenGL 应用程序成功设置 Xcode 的步骤。

1. 运行Xcode，在 macOS 标签下创建一个 command line tool（命令行工具）类型的项目。将语言设置为 C++。
2. 添加系统依赖库 OpenGL.framework 和 CoreFoundation
3. 设置头文件搜索路径，如下所示。
   1. 单击项目名称，选择 Build Settings 选项卡。向下滚动到“search paths”中（确保上方过滤器选择 All 而非 Basic）。
   2. 在 header search paths 中添加 /usr/local/include（在苹果 M1 芯片系统中的路径为 /opt/homebrew/include）
4. 为链接阶段设置二进制文件，如下所示：
   1. 来到主面板中的 Build phases 下，展开 Link Binary with Libraries；
   2. 将 /usr/local/Cellar/glew/2.2.0\_1/lib （在苹果 M1 芯片系统中的路径为 /opt/homebrew/Cellar/glew/2.2.0\_1/lib）路径下的 libGLEW.2.2.0.dylib 文件拖拽到 Link Binary with Libraries 下；
   3. 将 /usr/local/Cellar/glfw/3.3.4/lib（在苹果 M1 芯片系统中的路径为 /opt/homebrew/Cellar/glfw/3.3.4/lib）路径下的 libglfw.3.3.dylib 文件拖拽到 Link Binary with Libraries 下；

至此库的引入工作已经结束，来到main.cpp 文件，引入头文件并编译

#include <iostream>

#define GLEW\_STATIC

#include <GL/glew.h>

#include <GLFW/glfw3.h>

int main(int argc, const char \* argv[]) {

// insert code here...

std::cout << "Hello, World!\n";

return 0;

}

command+R 运行，如果没有报错的话说明环境搭建成功了，如果你在搭建OpenGL 环境中遇到什么问题，欢迎评论区留言或私信，大家相互学习交流。