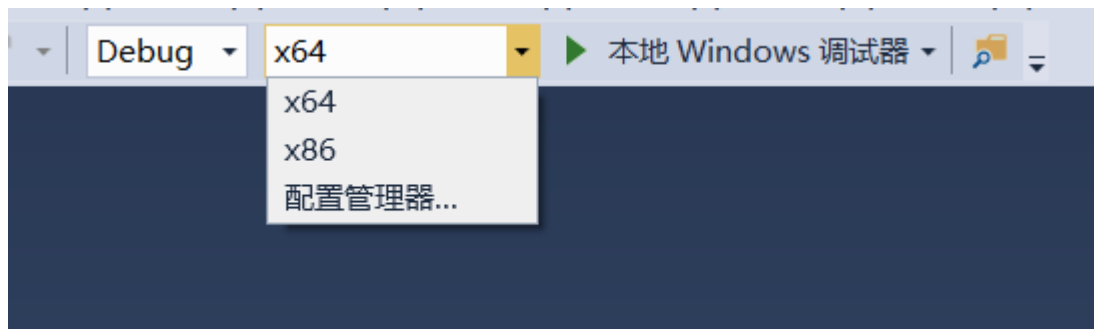


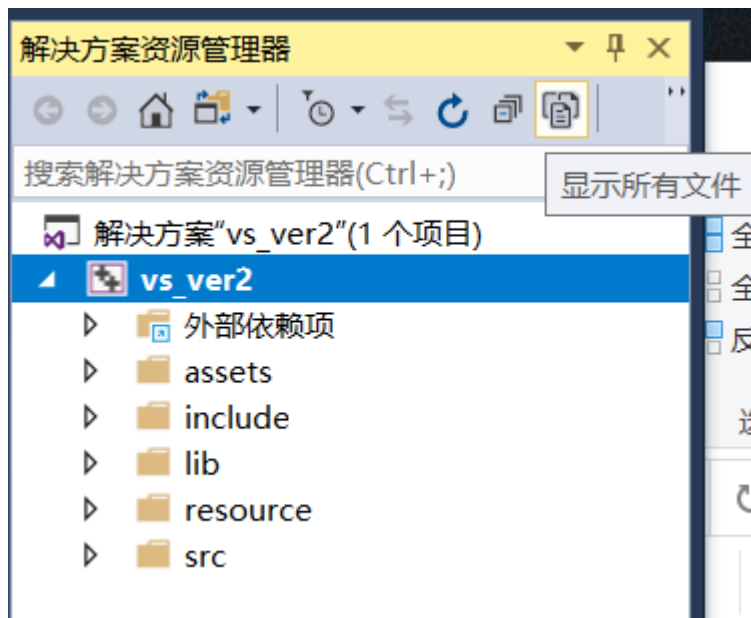
1、创建 vs 空项目，将五个文件夹添加到项目目录：

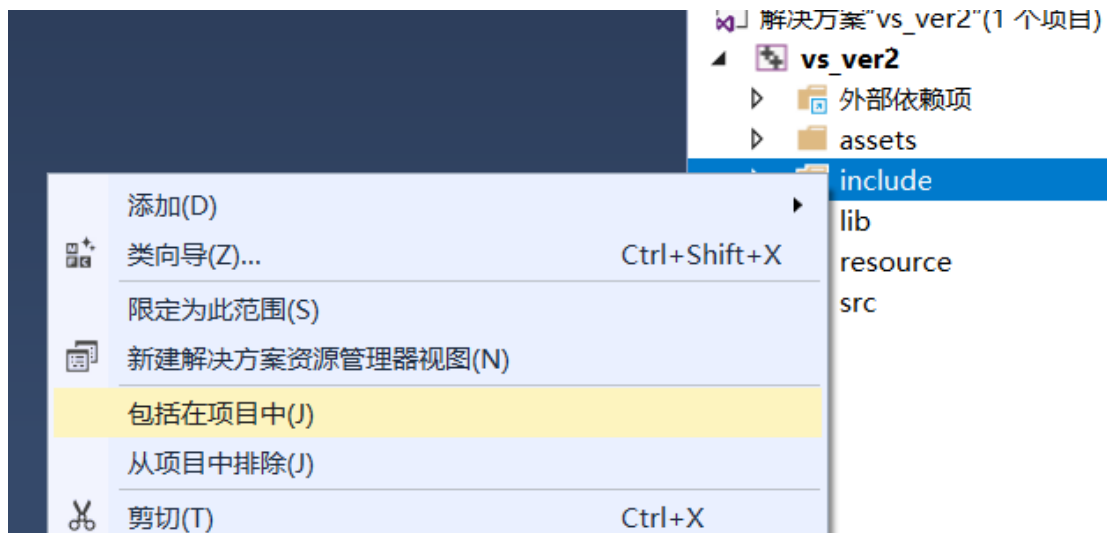
assets	2021/12/6 9:01	文件夹	
include	2021/12/6 9:01	文件夹	
lib	2021/12/6 9:01	文件夹	
resource	2021/12/6 9:01	文件夹	
src	2021/12/6 9:01	文件夹	
vs_ver2.vcxproj	2021/12/6 9:01	VC++ Project	6 KB
vs_ver2.vcxproj.filters	2021/12/6 9:01	VC++ Project Fil...	1 KB
vs_ver2.vcxproj.user	2021/12/6 9:01	Per-User Project...	1 KB

2、将 vs 的编译核改成 x64 版本



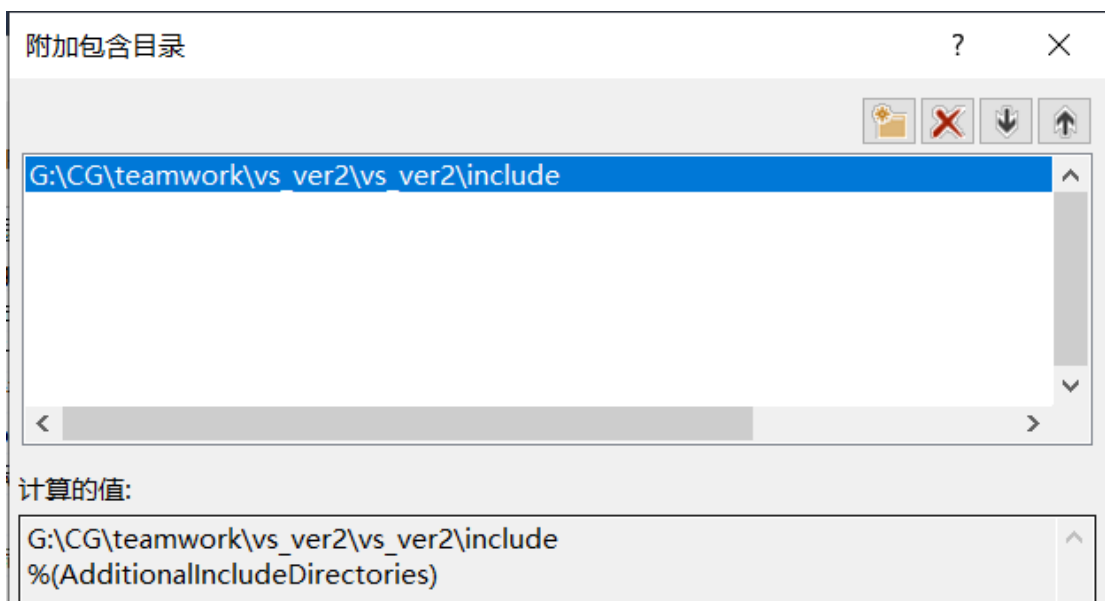
3、在 vs 的解决方案资源管理器中，点击“显示所有文件”，在 include 和 src 文件夹位置右键，选择“包括在项目中”，等待片刻



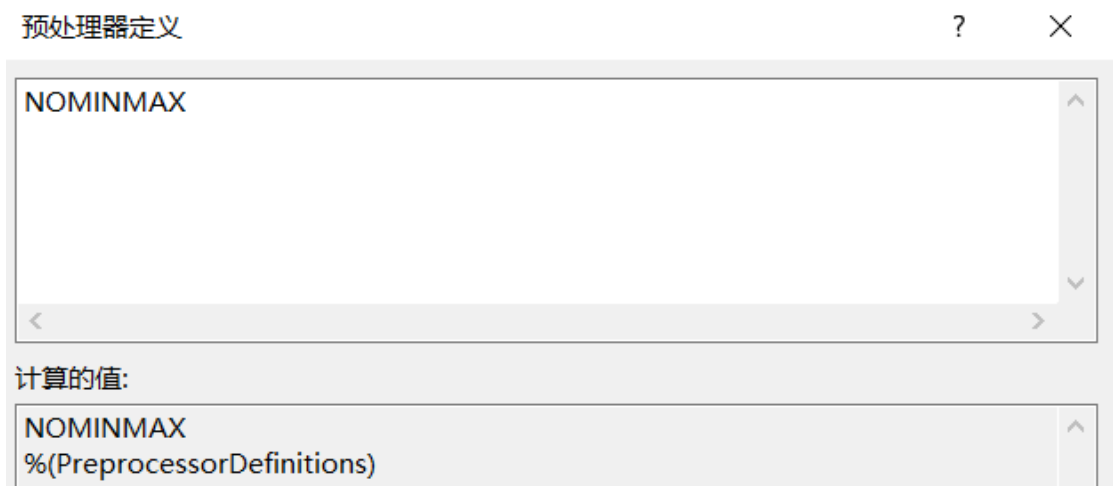


4、项目名处右键（图中为 vs_ver2），属性→配置属性，这里要加入一些配置。

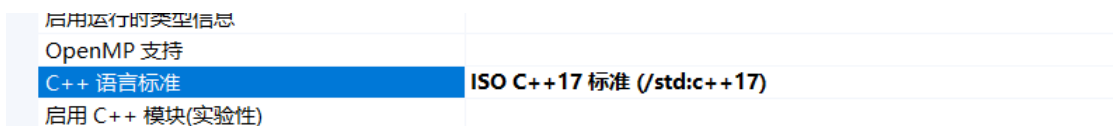
① C/C++→常规→附加包含目录，添加我们的 include 文件夹



② C/C++→预处理器→预处理器定义，添加 NOMINMAX



③ C/C++ → 语言 → C++ 语言标准, 选择 C++17

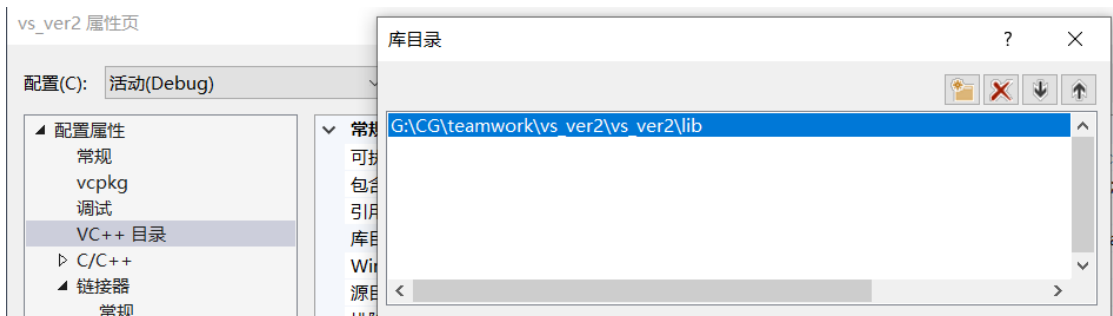


④ 链接器 → 输入 → 添加依赖项, 输入 lib 文件中的所有后缀名为 ".lib" 的文件 (我们共 7 项)

另注我的 vs 版本为 2017, 其中 glfw 是使用 vs2017 编译出来的版本, 理论上你们的 2019 应该也能用, 如果提示 glfw 有问题再跟我说, 我给你们换成 2019 版



⑤ VC++ 目录 → 库目录, 加入我们的 lib 文件夹



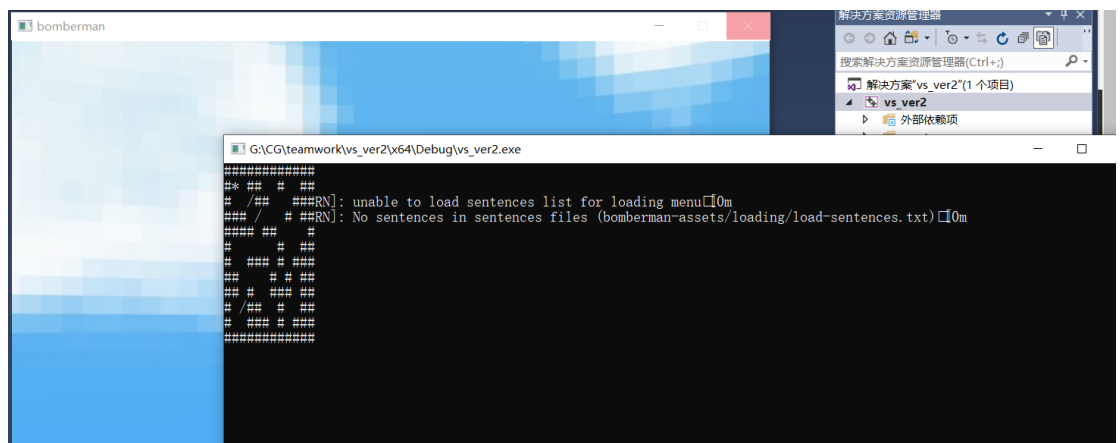
⑥ 点击"应用", "确定", `ctrl + F5` 运行

第一次运行提示找不到两个 `dll` 文件, 把 `lib` 文件夹中的两个 `dll`(`glfw3.dll` 不需要)复制到 `x64` 的 `exe` 文件所在的文件夹中

此电脑 > 新加卷 (G:) > CG > teamwork > vs_ver2 > x64 > Debug					搜索"Debug"
名称	修改日期	类型	大小		
assimp-vc140-mt.dll	2021/11/23 21:01	应用程序扩展	13,900 KB		
glfw3.dll	2021/10/29 0:51	应用程序扩展	208 KB		
SDL2.dll	2021/11/30 12:24	应用程序扩展	2,176 KB		
vs_ver2.exe	2021/12/6 9:26	应用程序	3,586 KB		
vs_ver2.ilc	2021/12/6 9:26	Incremental Link...	11,280 KB		
vs_ver2.pdb	2021/12/6 9:26	Program Debug ...	9,844 KB		

如果提示缺少其它的 `dll` 文件, 则是使用的动态库需要该库依赖, 而 windows 库目录本身缺少对应动态库, 自行补齐即可。(如我提示缺少 `VCRUNTIME140_1D.dll`, 也许你们用的上, 一并放在 `lib` 目录了)

运行成功应该有天空盒和 `cmd` 游戏逻辑



另注: 我在程序里加了一段简单粗暴的测试画模型的代码, 运行不报错, 但目前看不到模型, 不知道是相机位置的问题, 还是模型本身的问题, 先不管, 先把项目跑起来

(另另注: 相机的参数在 `gui.h`, 我只用了 `CAM_POS_OFFSET`, 其它的使用的是 `camera.h` 中构造函数的默认参数)

```
gui.cpp  gui.h  SceneGame.cpp x
vs_ver2  SceneGame  drawGame()

250
251 // draw enemies
252 Shader ourShader("resource/model_loading.vs", "resource/model_loading.fs");
253 Model ourModel("resource/model/box.obj");
254 ourShader.use();
255
256 // view/projection transformations
257 glm::mat4 projection = glm::perspective(glm::radians(this->_gui->cam->Zoom), (float)SCR_WIDTH / (float)SCR_HEIGHT, 0.1f, 100.0f);
258 glm::mat4 view = this->_gui->cam->GetViewMatrix();
259 ourShader.setMat4("projection", projection);
260 ourShader.setMat4("view", view);
261
262 // render the loaded model
263 glm::mat4 model = glm::mat4(1.0f);
264 model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f)); // translate it down so it's at the center of the scene
265 model = glm::scale(model, glm::vec3(5.0f, 5.0f, 5.0f)); // it's a bit too big for our scene, so scale it down
266 ourShader.setMat4("model", model);
267 ourModel.Draw(ourShader);
268
269
270 // draw player
```