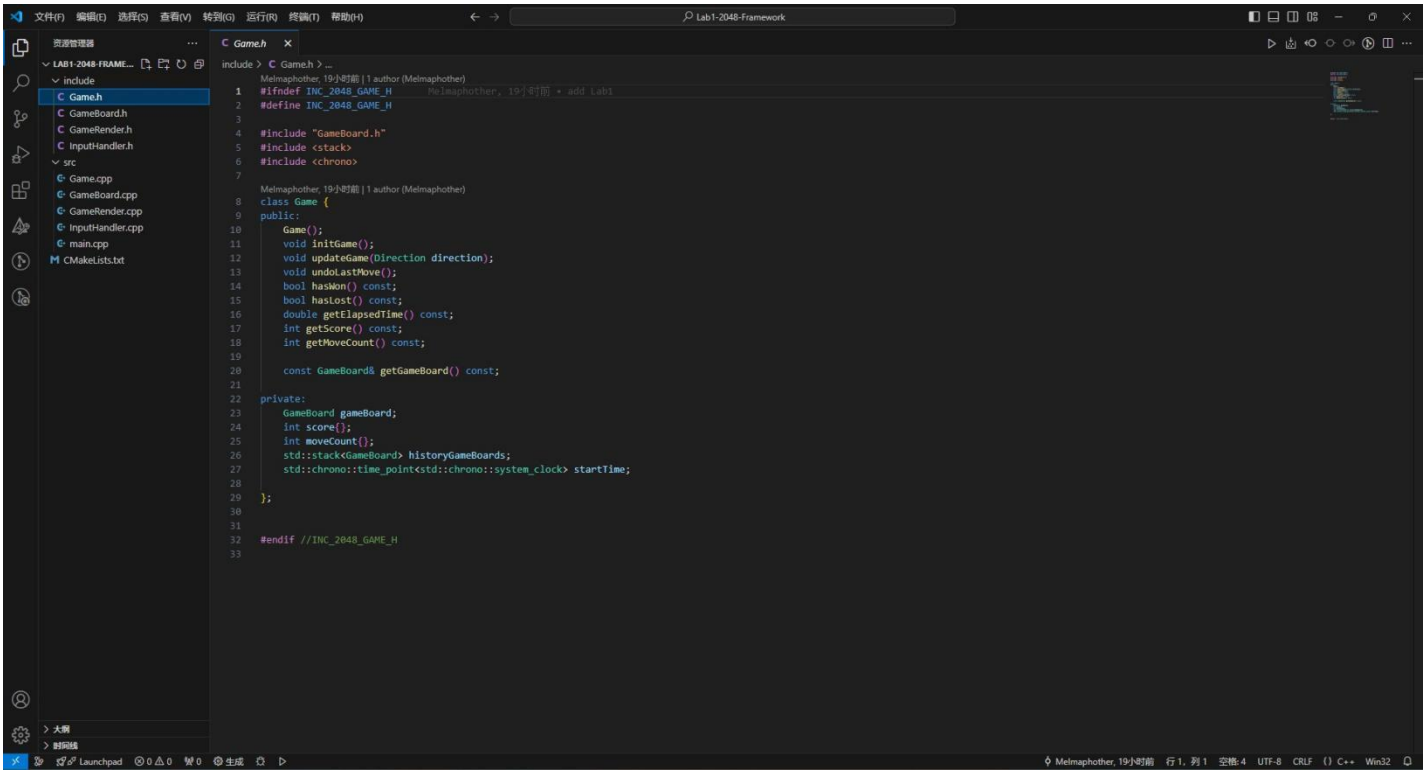


项目构建步骤说明(Windows版本)

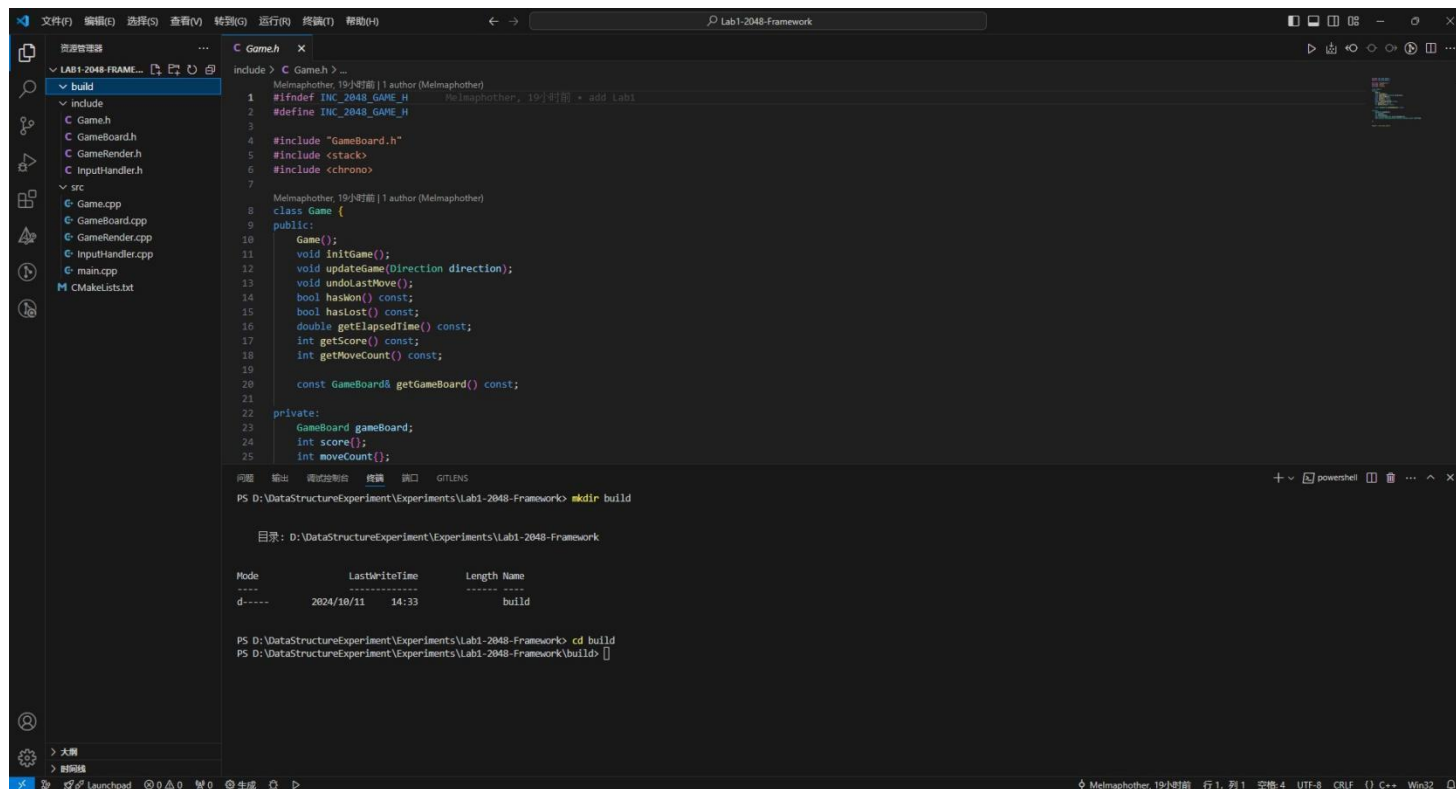
在你的代码填写完成后，初始的代码组织结构为：

```
Lab1
├─ CMakeLists.txt
├─ include
│   ├── Game.h
│   ├── GameBoard.h
│   ├── GameRender.h
│   └─ InputHandler.h
└─ src
    ├── Game.cpp
    ├── GameBoard.cpp
    ├── GameRender.cpp
    ├── InputHandler.cpp
    └─ main.cpp
```



创建build文件夹

在vscode代码下面的窗口打开终端，使用 `mkdir build` 创建 `build` 文件夹，用于放置构建项目依赖的文件以及构建好的文件，然后使用 `cd build` 命令进入 `build` 文件夹。



在构建之前需要先下载好两个工具

MinGW

使用官网下载 <https://github.com/GorvGoyl/MinGW64/releases/download/v2.0/MinGW64.zip>（如果有魔法）或从群里助教上传的 mingw64 文件下载，下载到本地后，需要添加环境变量，按以下步骤进行：

- 找到 `mingw64/bin` 的绝对路径：

| | | | | | |
|--|-----------------|-----------------|--------|----------|--|
| D:\download\x86_64-8.1.0-rel | | | | | |
| D:\download\x86_64-8.1.0-release-posix-sjlj-r1-x6-rev0\mingw64\bin | | | | | |
| 在 bin 中搜索 | | | | | |
| 新建 删除 重命名 移动 复制 粘贴 打印 排序 查看 更多 | | | | | |
| 预览 | | | | | |
| WP5云盘 | 名称 | 修改日期 | 类型 | 大小 | |
| 主文件夹 | add2line | 2018/5/11 18:46 | 应用程序 | 1,031 KB | |
| 图库 | ar | 2018/5/11 18:46 | 应用程序 | 1,055 KB | |
| OneDrive | as | 2018/5/11 18:46 | 应用程序 | 1,853 KB | |
| | c++ | 2018/5/11 22:16 | 应用程序 | 1,914 KB | |
| | c++filt | 2018/5/11 18:46 | 应用程序 | 1,030 KB | |
| Windows (C:) | cpp | 2018/5/11 22:16 | 应用程序 | 1,913 KB | |
| lab3 | dlltool | 2018/5/11 18:46 | 应用程序 | 1,087 KB | |
| qq_download | dllwrap | 2018/5/11 18:46 | 应用程序 | 52 KB | |
| download | dwp | 2018/5/11 18:46 | 应用程序 | 3,032 KB | |
| lab2 | efedit | 2018/5/11 18:46 | 应用程序 | 38 KB | |
| src | g++ | 2018/5/11 22:16 | 应用程序 | 1,914 KB | |
| | gcc | 2018/5/11 22:16 | 应用程序 | 1,912 KB | |
| | gcc-ar | 2018/5/11 22:16 | 应用程序 | 58 KB | |
| 此电脑 | gcc-nm | 2018/5/11 22:16 | 应用程序 | 58 KB | |
| Windows (C:) | gcc-ranlib | 2018/5/11 22:16 | 应用程序 | 58 KB | |
| Data (D:) | gcov | 2018/5/11 22:16 | 应用程序 | 1,507 KB | |
| 网络 | gcov-dump | 2018/5/11 22:16 | 应用程序 | 1,371 KB | |
| Linux | gcov-tool | 2018/5/11 22:16 | 应用程序 | 1,425 KB | |
| | gdb | 2018/5/11 23:59 | 应用程序 | 59 KB | |
| | gdborig | 2018/5/11 23:59 | 应用程序 | 9,960 KB | |
| | gdbserver | 2018/5/11 23:59 | 应用程序 | 507 KB | |
| | gendef | 2018/5/11 22:19 | 应用程序 | 55 KB | |
| | genid | 2018/5/11 22:19 | 应用程序 | 72 KB | |
| | genpeimg | 2018/5/11 22:20 | 应用程序 | 28 KB | |
| | gfortran | 2018/5/11 22:16 | 应用程序 | 1,914 KB | |
| | gprof | 2018/5/11 18:46 | 应用程序 | 1,094 KB | |
| | ld.bfd | 2018/5/11 18:46 | 应用程序 | 1,455 KB | |
| | ld | 2018/5/11 18:46 | 应用程序 | 1,455 KB | |
| | ld.gold | 2018/5/11 18:46 | 应用程序 | 3,265 KB | |
| | libatomic-1.dll | 2018/5/11 22:18 | 应用程序扩展 | 32 KB | |
| 56 个项目 选中 1 个项目 216 KB | | | | | |

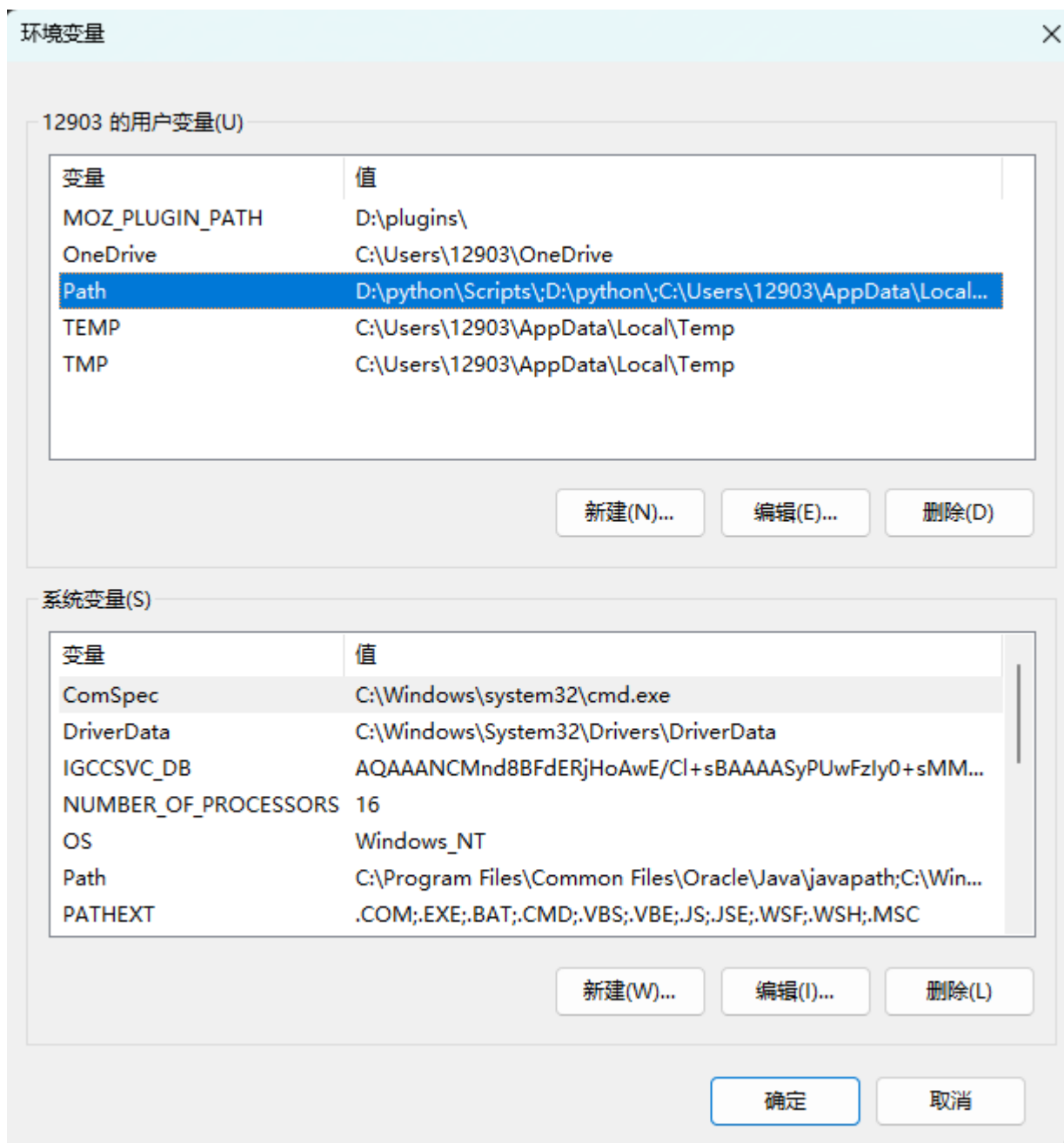
- 在系统菜单里选择编辑系统环境变量



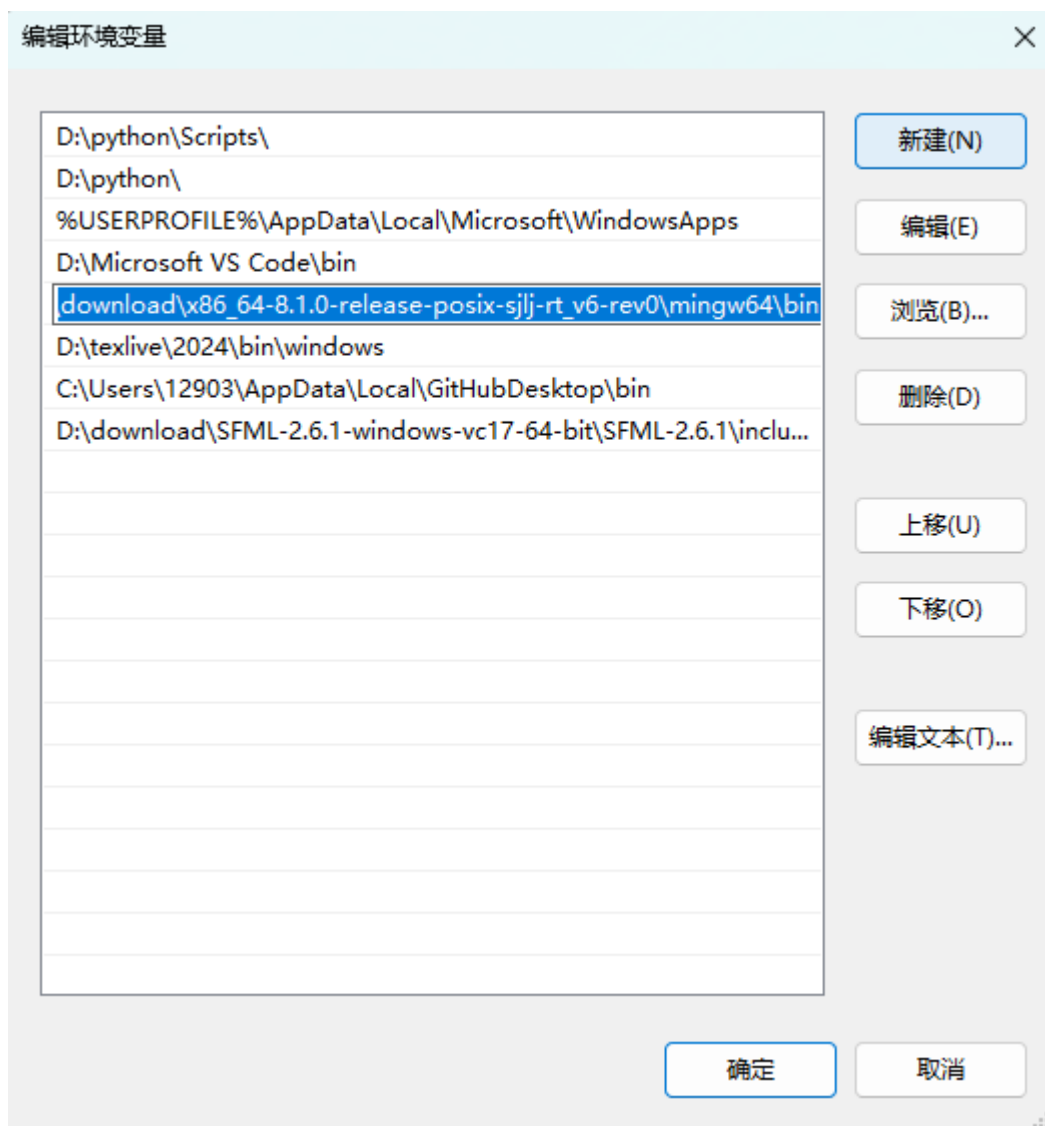
- 点击 系统变量



- 双击选择 Path 进行编辑



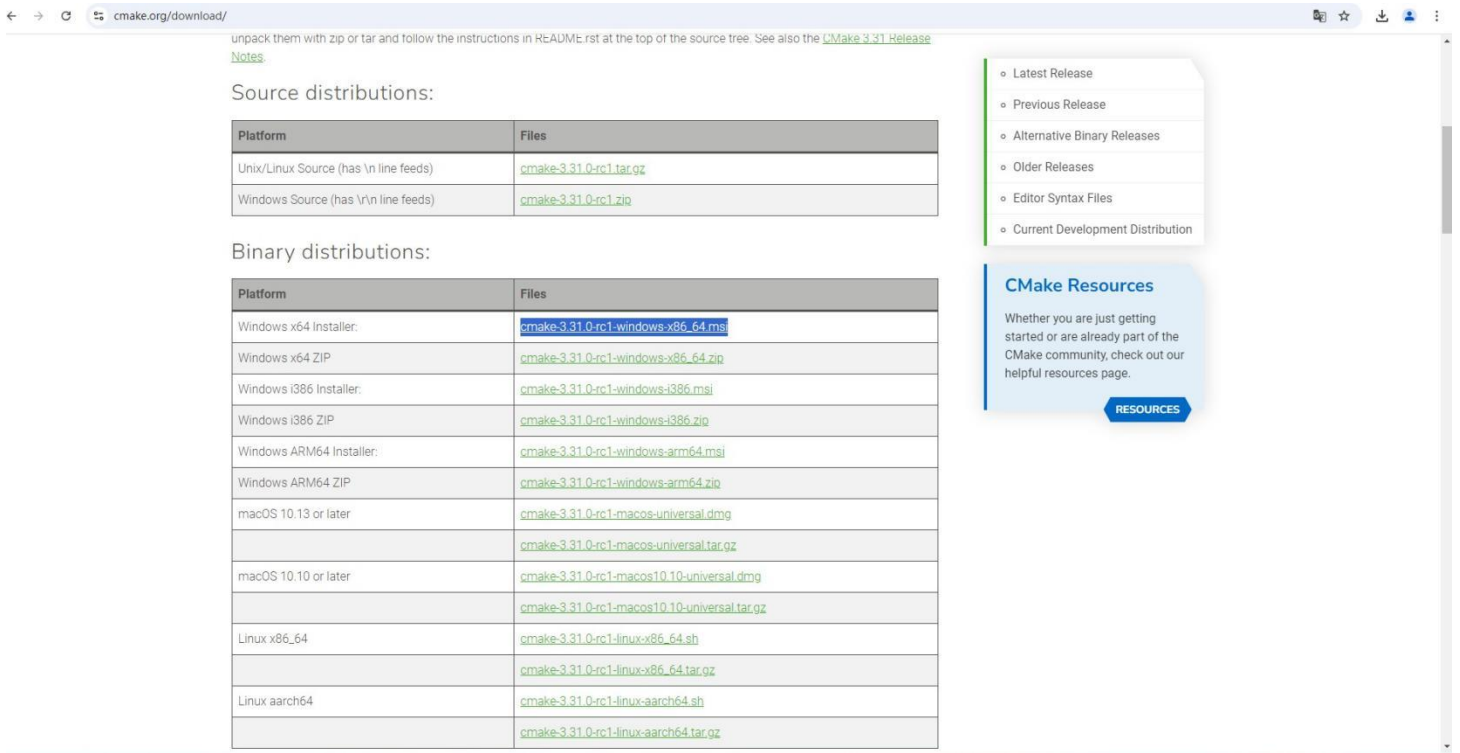
- 新建一栏，粘贴上刚刚找到的 mingw64/bin 的绝对路径，点击确定即可保存。



这个操作是为了让系统能够找到mingw构建工具。

CMake

去CMake的下载网站 <https://cmake.org/download/> 下载



安装过程中会自动勾选 添加至环境变量 ，不用手动设置。

构建过程

1. 直接在终端运行构建命令

- 下载好必需的工具后，在 build 目录下的终端里输入 `cmake .. -G "MinGW Makefiles"` 命令进行 makefile 的生成：

```
PS D:\DataStructureExperiment\Experiments\Lab1-2048-Framework\build> cmake .. -G "MinGW Makefiles"
-- The C compiler identification is GNU 8.1.0
-- The CXX compiler identification is GNU 8.1.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: D:/download/x86_64-8.1.0-release-posix-sjlj-rt_v6-rev0/mingw64/bin/gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: D:/download/x86_64-8.1.0-release-posix-sjlj-rt_v6-rev0/mingw64/bin/c++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done (17.1s)
-- Generating done (0.0s)
-- Build files have been written to: D:/DataStructureExperiment/Experiments/Lab1-2048-Framework/build
PS D:\DataStructureExperiment\Experiments\Lab1-2048-Framework\build>
```

- makefile 生成成功后，执行命令 `mingw32-make`，如果源代码没有问题的话，make 过程输出参考如下：

```

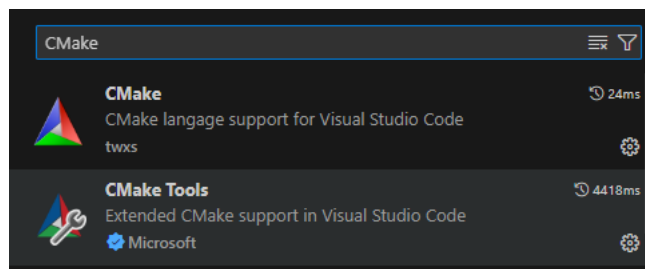
PS D:\DataStructureExperiment\Experiments\Lab1-2048-Framework\build> mingw32-make
[ 16%] Building CXX object CMakeFiles/2048.dir/src/main.cpp.obj
[ 33%] Building CXX object CMakeFiles/2048.dir/src/Game.cpp.obj
[ 50%] Building CXX object CMakeFiles/2048.dir/src/GameBoard.cpp.obj
[ 66%] Building CXX object CMakeFiles/2048.dir/src/InputHandler.cpp.obj
[ 83%] Building CXX object CMakeFiles/2048.dir/src/GameRender.cpp.obj
[100%] Linking CXX executable 2048.exe
[100%] Built target 2048
PS D:\DataStructureExperiment\Experiments\Lab1-2048-Framework\build>

```

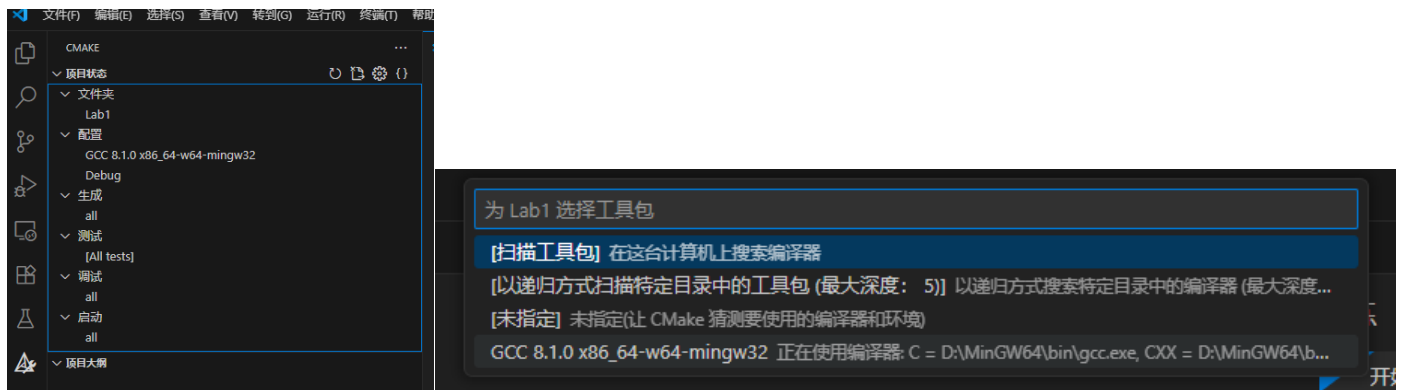
中间构建失败了可能是代码存在错误，需要进行修改。

2. 使用vscode中的CMake插件（推荐，更方便）

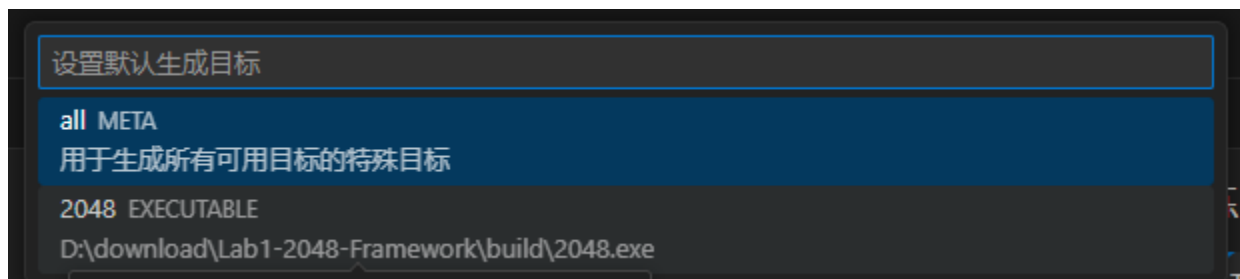
- 在vscode中下载 CMake 和 CMake Tools 插件，能自动检测本地的CMake构建工具



- 配置处选择本地安装的 mingw64 编译工具



- 点击生成可选择生成的可执行文件



- 最终运行或调试程序可直接使用下面的“调试”或“启动”