Software Installation Guidance for Coursework 2

Chongshou Li

sjucli@leeds.ac.uk

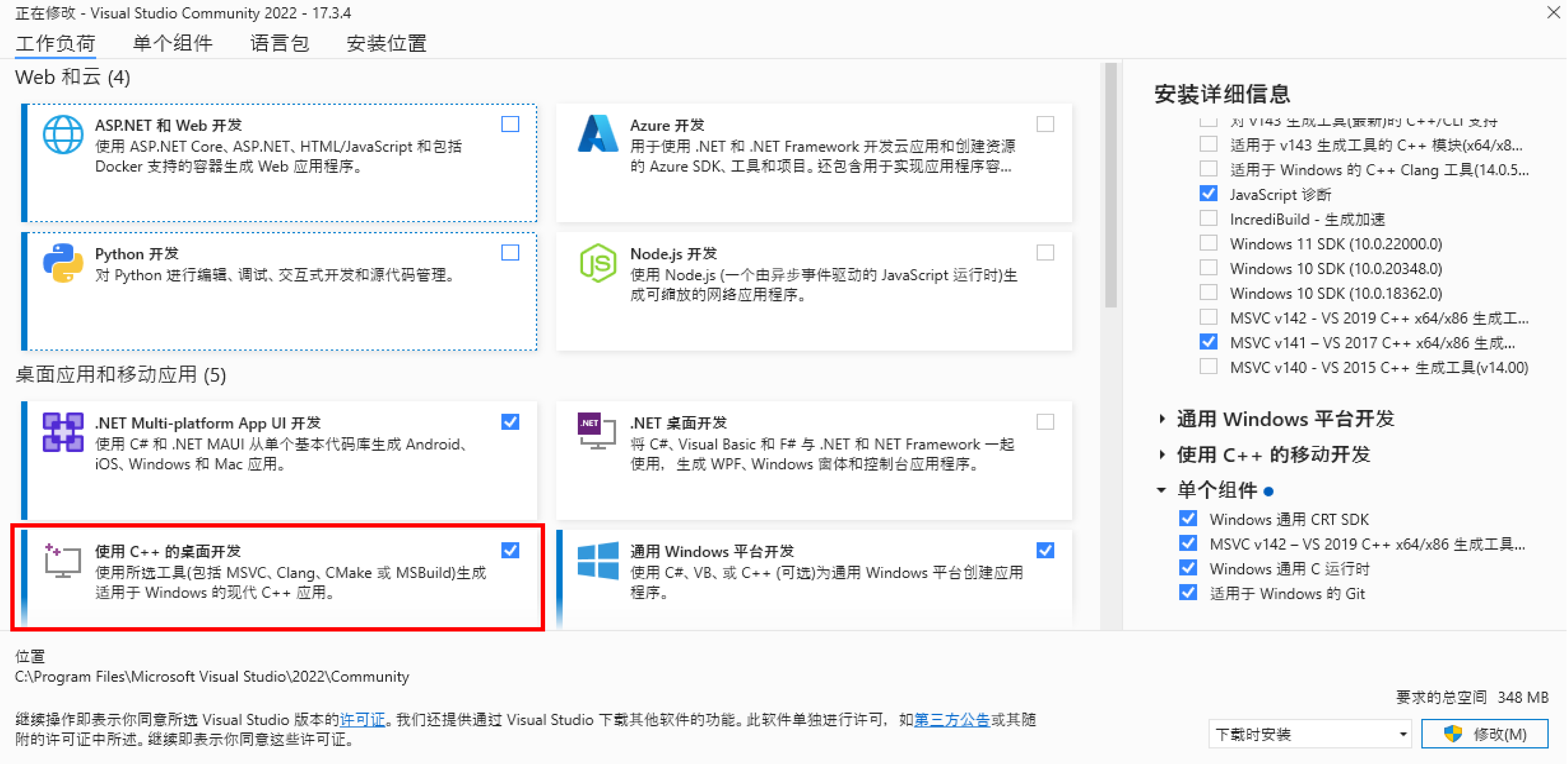
SWJTU-LEEDS JOINT SCHOOL

Fresh Install

Visual Studio Community 2022

- Select *Desktop development with C++* in the category *Workload* (see the following slide)
- In the category “*Individual components*”
  - *Compilers, build tools, runtimes* → *Windows Universal CRT SDK*
  - *SDKs, libraries, and frameworks* → *Windows Universal C Runtime*
  - *(optional) Code tools* → *Git for Windows*

Note: the total installation seems to require about 9GB of disk space.



正在修改 - Visual Studio Community 2022 - 17.3.4

Web 和云 (4)

- ASP.NET 和 Web 开发 □
  使用 ASP.NET Core、ASP.NET、HTML/JavaScript 和包括 Docker 支持的容器生成 Web 应用程序。
- Azure 开发 □
  用于使用 .NET 和 .NET Framework 开发云应用和创建资源的 Azure SDK、工具和项目。还包含用于实现应用程序容...
- Python 开发 □
  对 Python 进行编辑、调试、交互式开发和源代码管理。
- Node.js 开发 □
  使用 Node.js (一个由异步事件驱动的 JavaScript 运行时)生成可缩放的网络应用程序。

桌面应用和移动应用 (5)

- .NET Multi-platform App UI 开发 ☑
  使用 C# 和 .NET MAUI 从单个基本代码库生成 Android、iOS、Windows 和 Mac 应用。
- .NET 桌面开发 □
  将 C#、Visual Basic 和 F# 与 .NET 和 NET Framework 一起使用，生成 WPF、Windows 窗体和控制台应用程序。
- 使用 C++ 的桌面开发 ☑
  使用所选工具(包括 MSVC、Clang、CMake 或 MSBuild)生成适用于 Windows 的现代 C++ 应用。
- 通用 Windows 平台开发 ☑
  使用 C#、VB、或 C++ (可选)为通用 Windows 平台创建应用程序。

安装详细信息

- □ 对 v143 生成工具(最新)的 C++/CLI 支持
- □ 适用于 v143 生成工具的 C++ 模块(x64/x8...
- □ 适用于 Windows 的 C++ Clang 工具(14.0.5...
- ☑ JavaScript 诊断
- □ IncrediBuild - 生成加速
- □ Windows 11 SDK (10.0.22000.0)
- □ Windows 10 SDK (10.0.20348.0)
- □ Windows 10 SDK (10.0.18362.0)
- □ MSVC v142 - VS 2019 C++ x64/x86 生成工...
- ☑ MSVC v141 – VS 2017 C++ x64/x86 生成...
- □ MSVC v140 - VS 2015 C++ 生成工具(v14.00)

▸ 通用 Windows 平台开发

▸ 使用 C++ 的移动开发

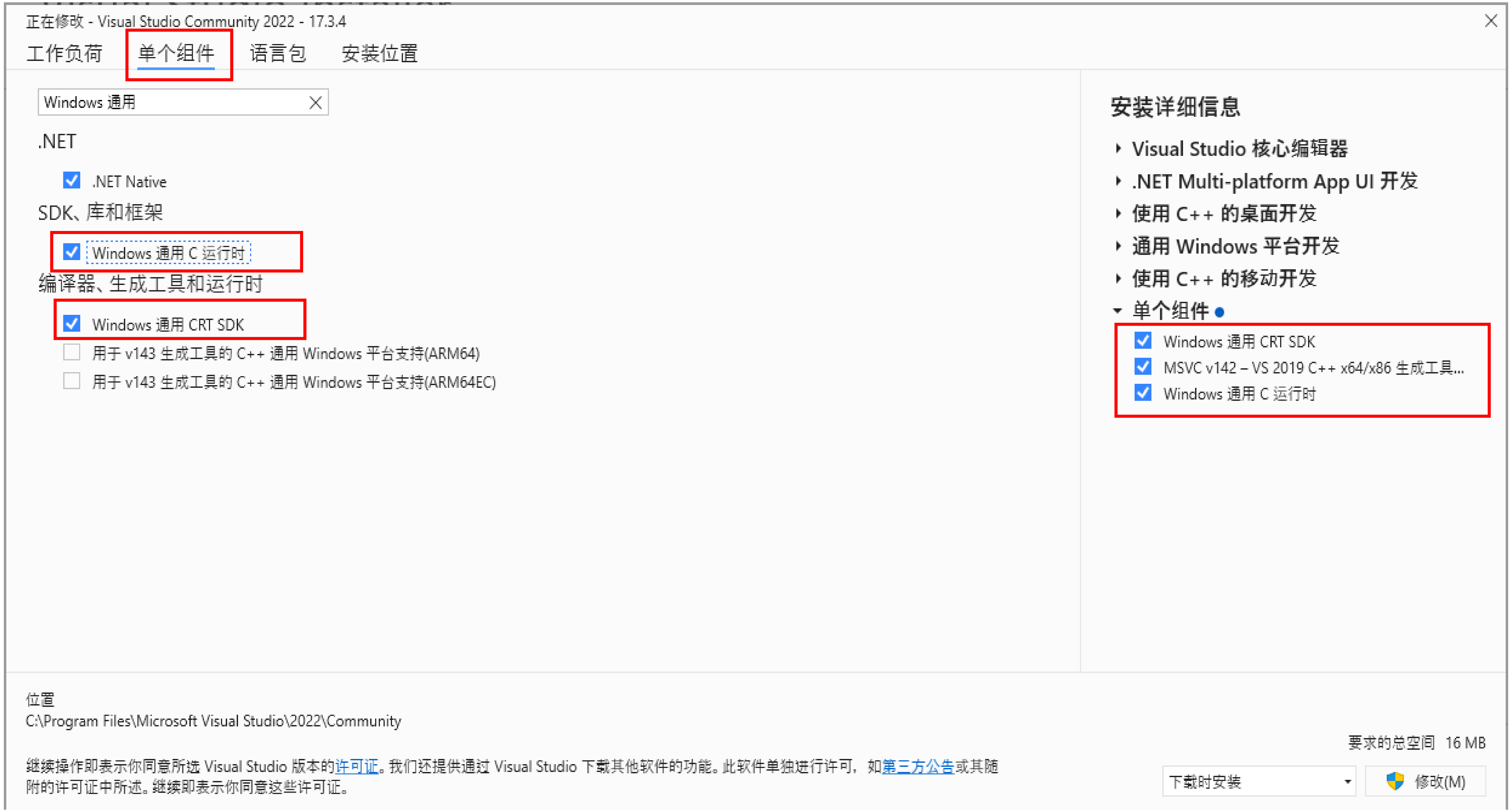
▾ 单个组件 ●

- ☑ Windows 通用 CRT SDK
- ☑ MSVC v142 – VS 2019 C++ x64/x86 生成工具...
- ☑ Windows 通用 C 运行时
- ☑ 适用于 Windows 的 Git

位置
C:\Program Files\Microsoft Visual Studio\2022\Community

继续操作即表示你同意所选 Visual Studio 版本的许可证。我们还提供通过 Visual Studio 下载其他软件的功能。此软件单独进行许可，如第三方公告或其随附的许可证中所述。继续即表示你同意这些许可证。

要求的总空间 348 MB



正在修改 - Visual Studio Community 2022 - 17.3.4

Windows 通用

.NET

☑ .NET Native

SDK、库和框架

☑ Windows 通用 C 运行时

编译器、生成工具和运行时

☑ Windows 通用 CRT SDK

□ 用于 v143 生成工具的 C++ 通用 Windows 平台支持(ARM64)

□ 用于 v143 生成工具的 C++ 通用 Windows 平台支持(ARM64EC)

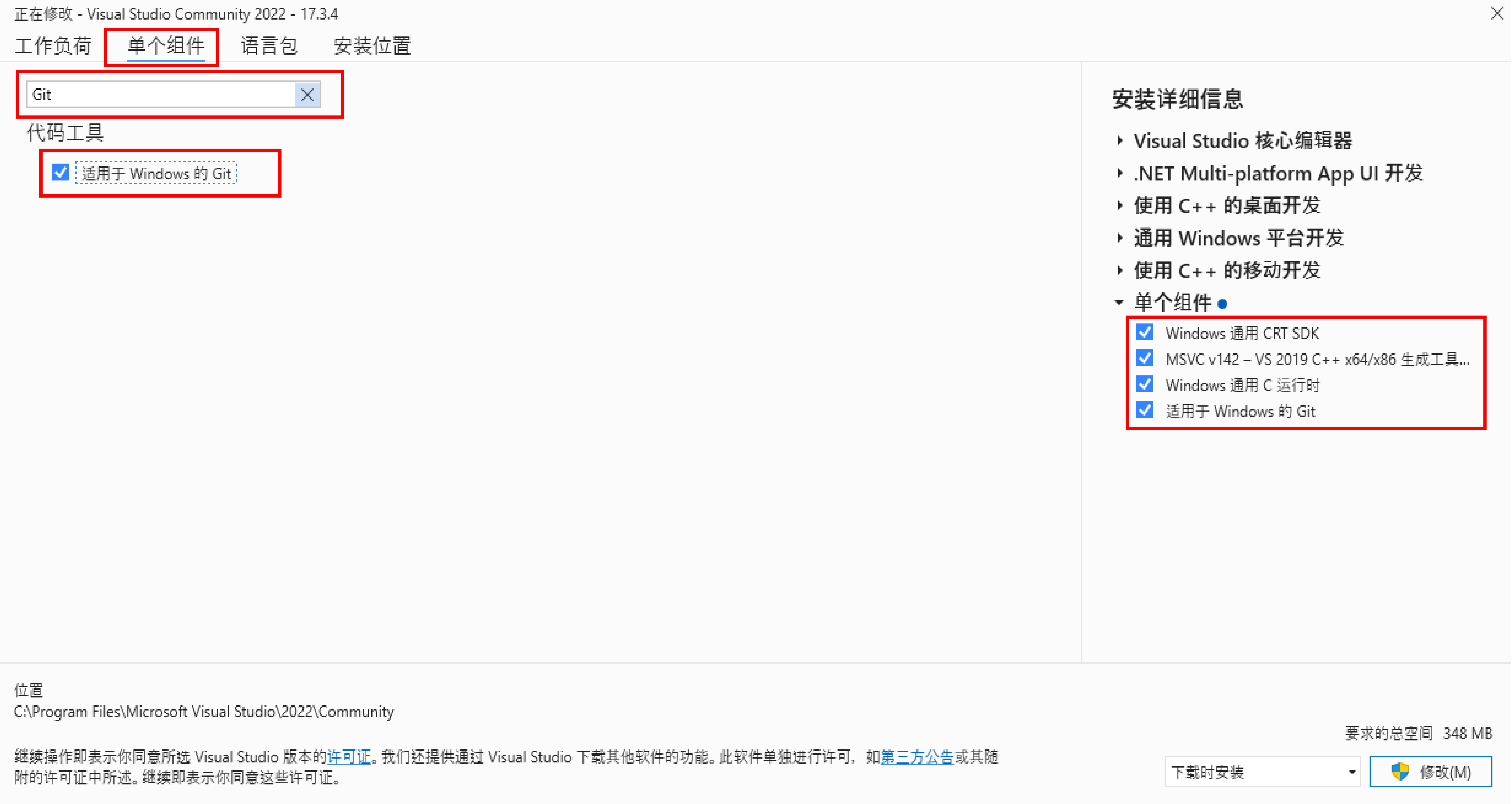
安装详细信息

- Visual Studio 核心编辑器
- .NET Multi-platform App UI 开发
- 使用 C++ 的桌面开发
- 通用 Windows 平台开发
- 使用 C++ 的移动开发
- 单个组件 ●
  - ☑ Windows 通用 CRT SDK
  - ☑ MSVC v142 – VS 2019 C++ x64/x86 生成工具...
  - ☑ Windows 通用 C 运行时

位置
C:\Program Files\Microsoft Visual Studio\2022\Community

继续操作即表示你同意所选 Visual Studio 版本的许可证。我们还提供通过 Visual Studio 下载其他软件的功能。此软件单独进行许可，如第三方公告或其随附的许可证中所述。继续即表示你同意这些许可证。

要求的总空间 16 MB



正在修改 - Visual Studio Community 2022 - 17.3.4

Git

代码工具

☑ 适用于 Windows 的 Git

安装详细信息

- Visual Studio 核心编辑器
- .NET Multi-platform App UI 开发
- 使用 C++ 的桌面开发
- 通用 Windows 平台开发
- 使用 C++ 的移动开发
- 单个组件 ●
  - ☑ Windows 通用 CRT SDK
  - ☑ MSVC v142 – VS 2019 C++ x64/x86 生成工具...
  - ☑ Windows 通用 C 运行时
  - ☑ 适用于 Windows 的 Git

位置
C:\Program Files\Microsoft Visual Studio\2022\Community

继续操作即表示你同意所选 Visual Studio 版本的许可证。我们还提供通过 Visual Studio 下载其他软件的功能。此软件单独进行许可，如第三方公告或其随附的许可证中所述。继续即表示你同意这些许可证。

要求的总空间 348 MB

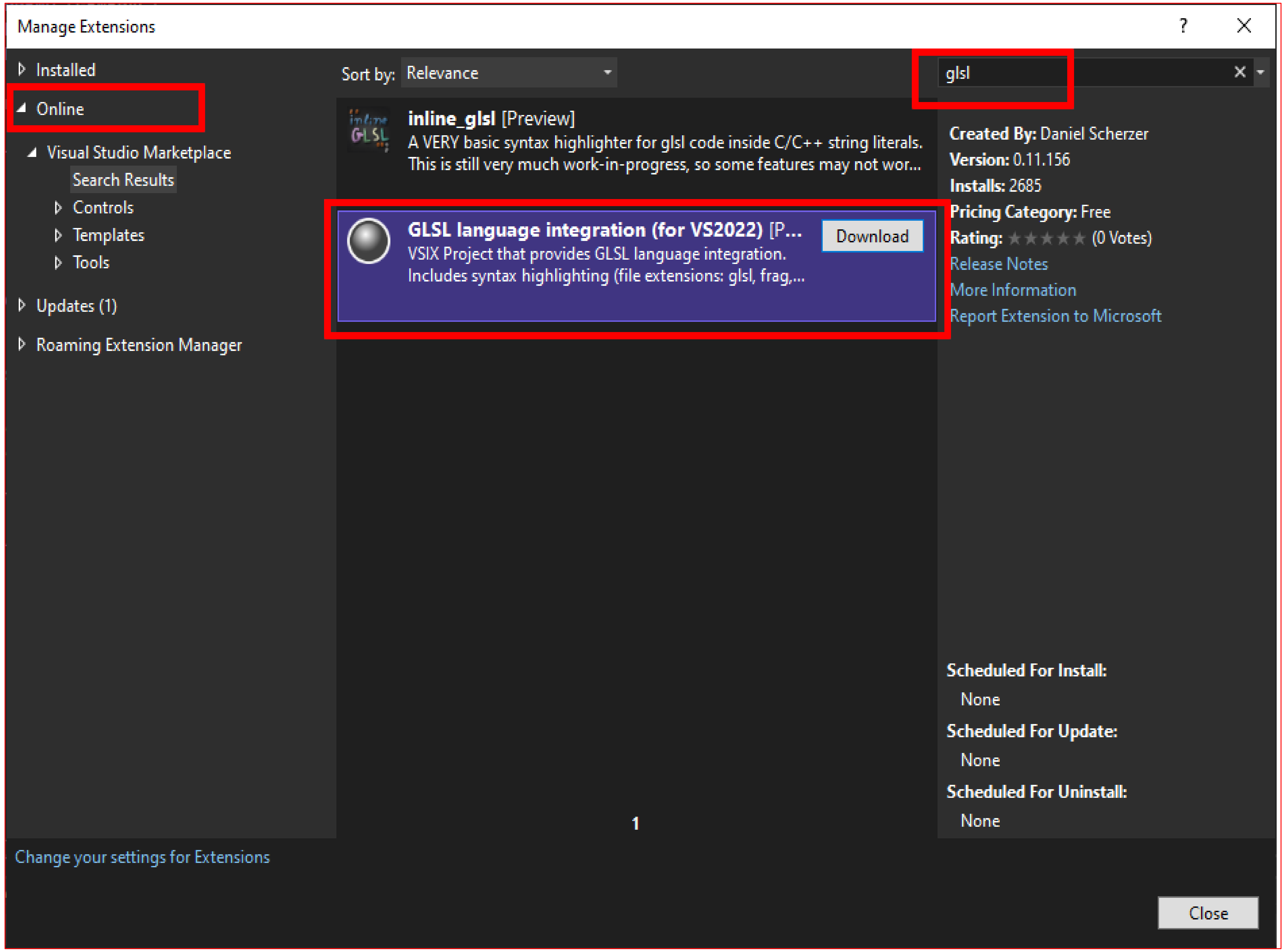
下载时安装

Highlighting for GLSL

- **You will be working with GLSL code quite a bit, but Visual Studio does not natively support GLSL source code. It is recommended that you install an extension that provides syntax highlight for GLSL code. To do so, go to Extensions ! Manage Extensions in the Visual Studio menu. This should bring up a popup (the following slide) where you can manage and install extensions. Here, select Online in the left-hand menu, and search for glsl in the top-right search field. You should find an extension called GLSL language integration (for VS2022) [Preview].**

- **Press Download to install the extension. You will likely need to restart VS to complete the installation.**

Highlighting for GLSL

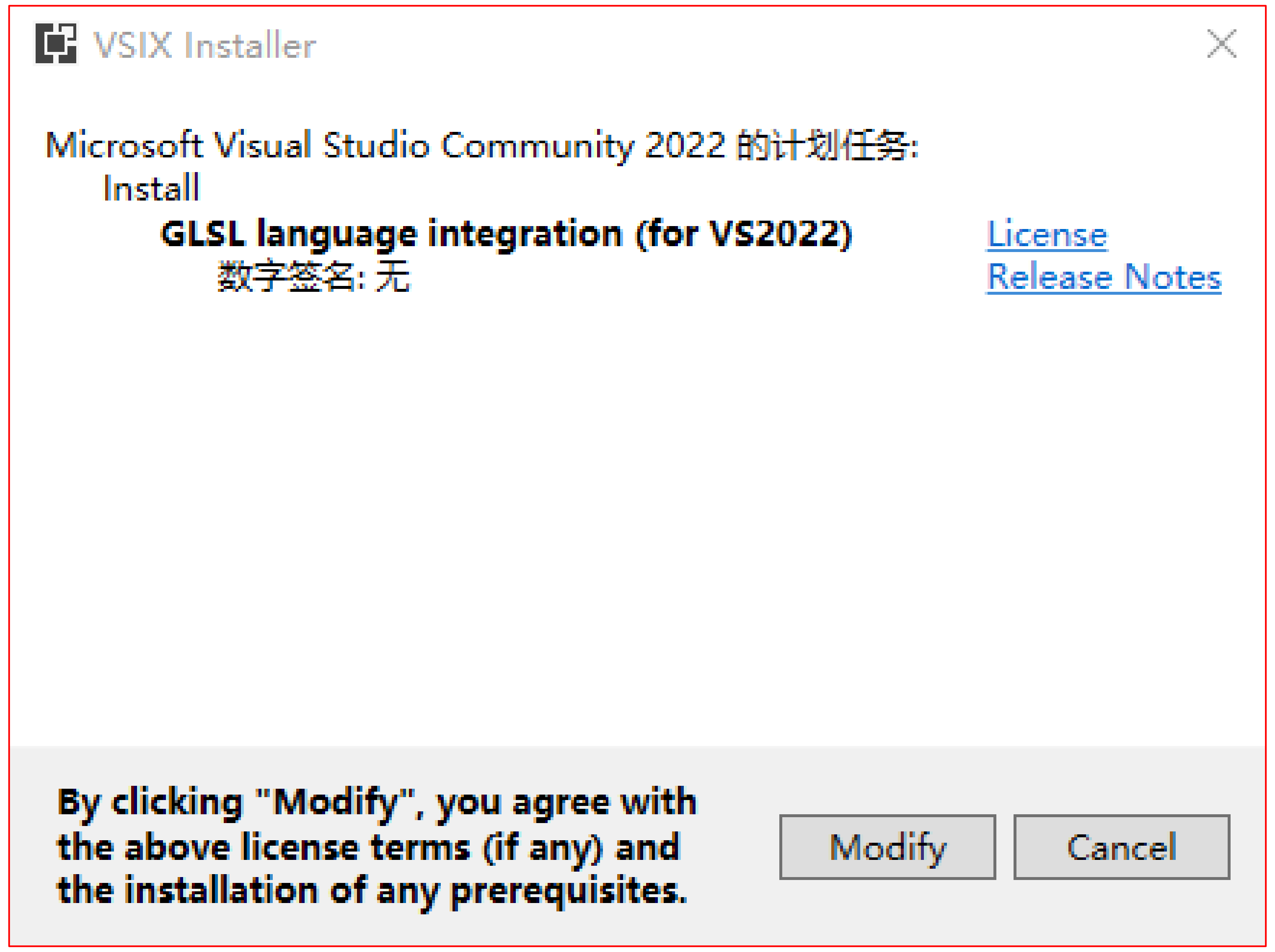


inline_glsl [Preview]
A VERY basic syntax highlighter for glsl code inside C/C++ string literals. This is still very much work-in-progress, so some features may not wor...

GLSL language integration (for VS2022) [P...
VSIX Project that provides GLSL language integration. Includes syntax highlighting (file extensions: glsl, frag,...

Created By: Daniel Scherzer
Version: 0.11.156
Installs: 2685
Pricing Category: Free
Rating: ★★★★★ (0 Votes)
Release Notes
More Information
Report Extension to Microsoft

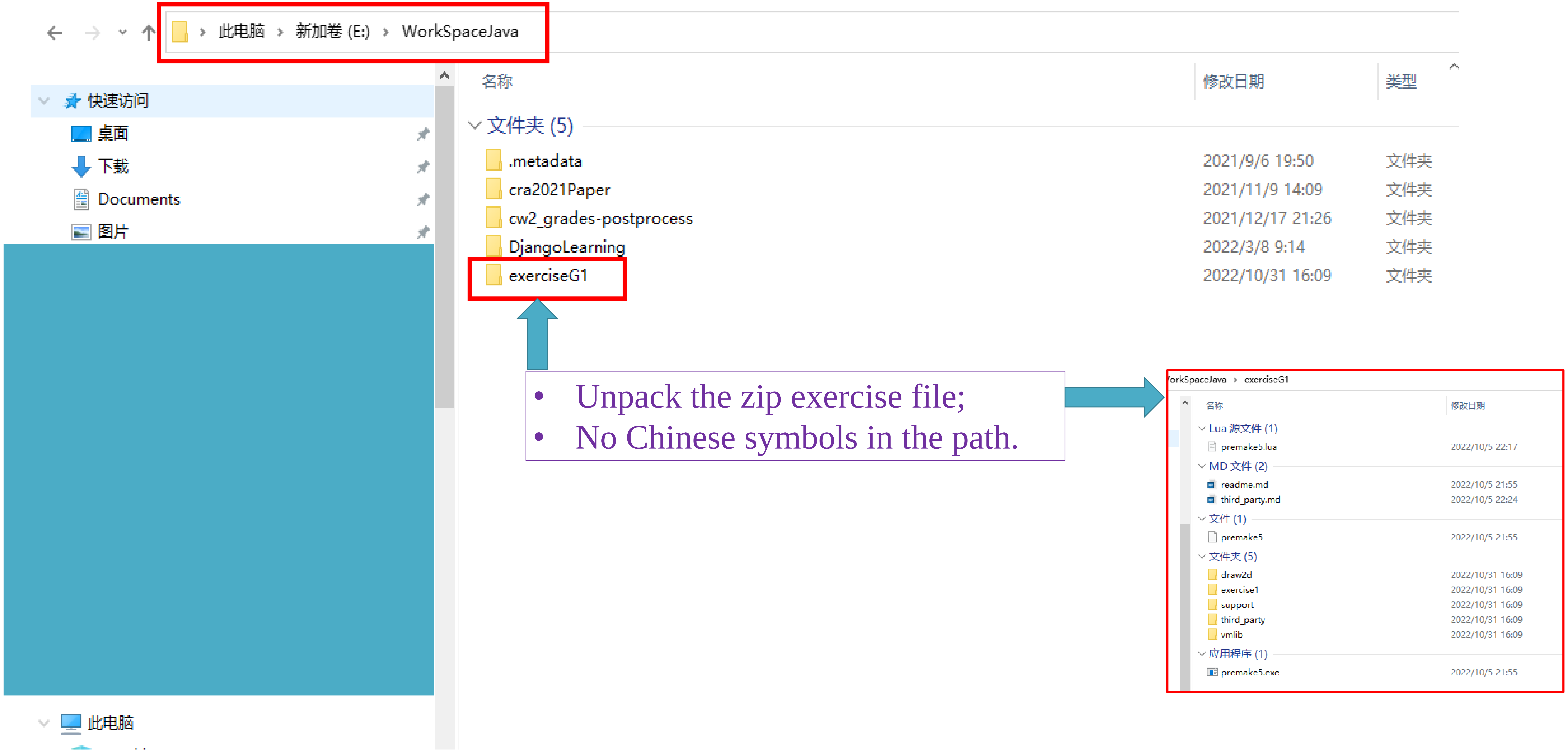
Scheduled For Install:
None
Scheduled For Update:
None
Scheduled For Uninstall:
None



Microsoft Visual Studio Community 2022 的计划任务:
Install
GLSL language integration (for VS2022)
数字签名: 无

By clicking "Modify", you agree with the above license terms (if any) and the installation of any prerequisites.

Building on Windows



此电脑 › 新加卷 (E:) › WorkSpaceJava

文件夹 (5)

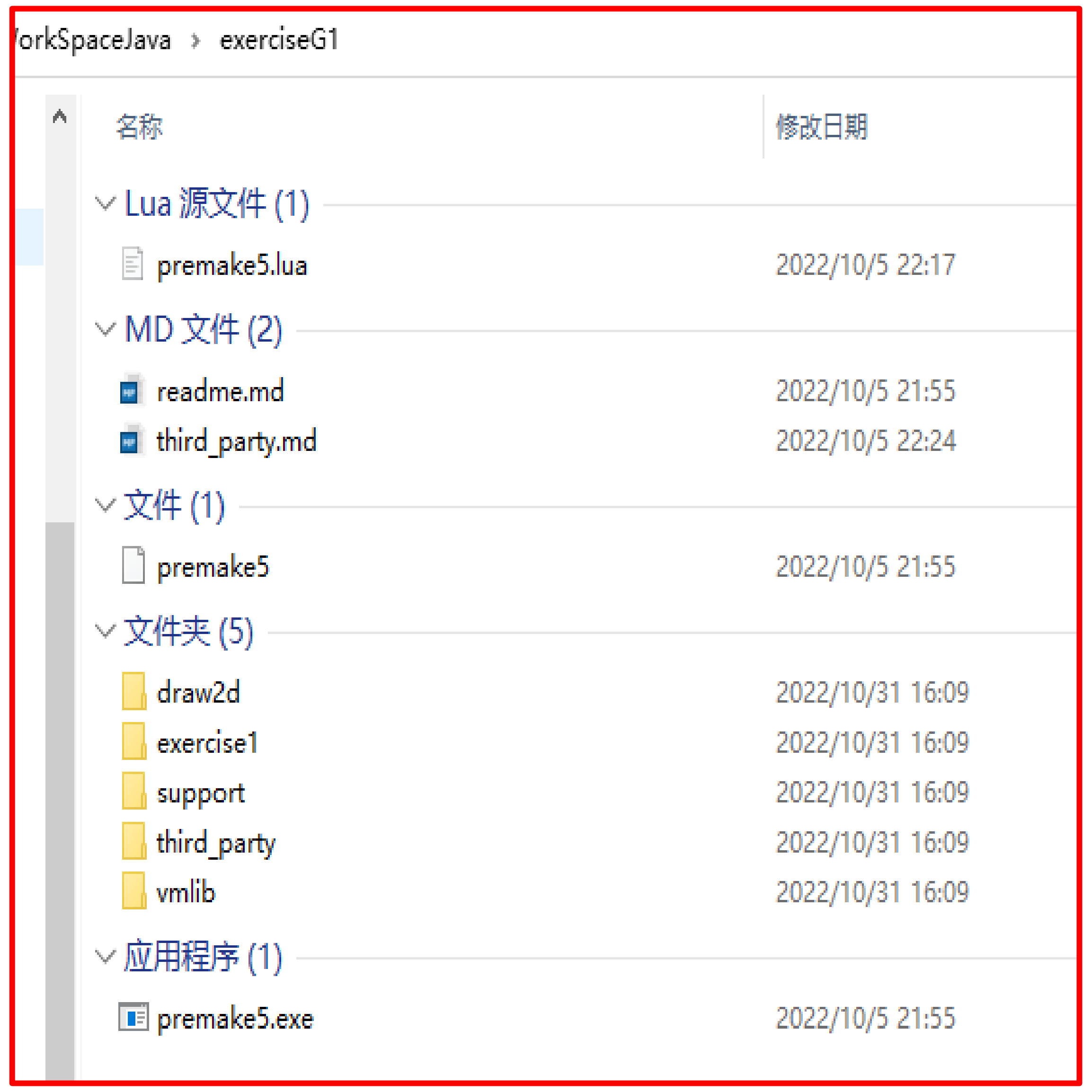
名称	修改日期	类型
.metadata	2021/9/6 19:50	文件夹
cra2021Paper	2021/11/9 14:09	文件夹
cw2_grades-postprocess	2021/12/17 21:26	文件夹
DjangoLearning	2022/3/8 9:14	文件夹
exerciseG1	2022/10/31 16:09	文件夹

- Unpack the zip exercise file;
- No Chinese symbols in the path.

'orkSpaceJava › exerciseG1

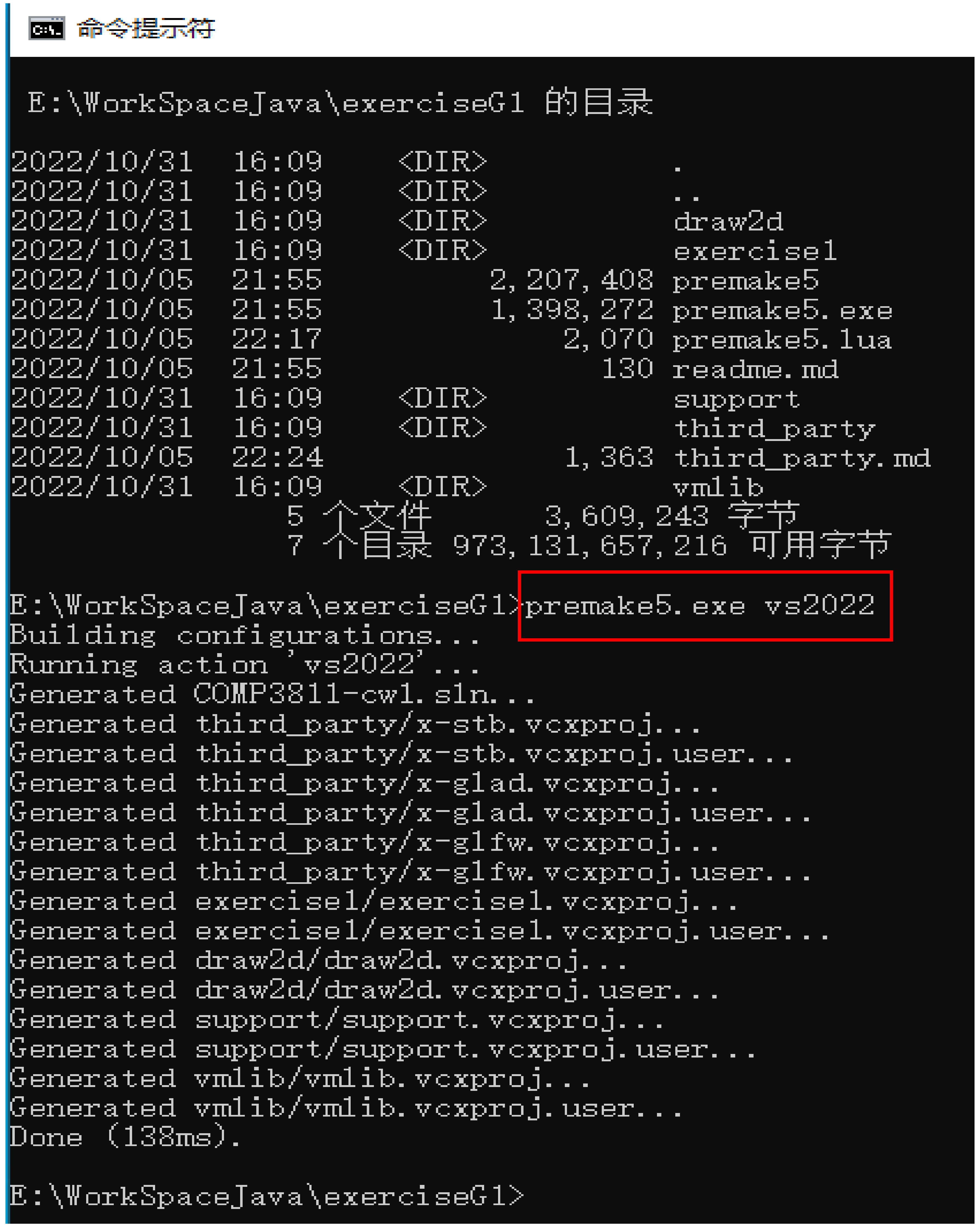
名称	修改日期
Lua 源文件 (1)	
premake5.lua	2022/10/5 22:17
MD 文件 (2)	
readme.md	2022/10/5 21:55
third_party.md	2022/10/5 22:24
文件 (1)	
premake5	2022/10/5 21:55
文件夹 (5)	
draw2d	2022/10/31 16:09
exercise1	2022/10/31 16:09
support	2022/10/31 16:09
third_party	2022/10/31 16:09
vmlib	2022/10/31 16:09
应用程序 (1)	
premake5.exe	2022/10/5 21:55

Step 1: Use premake from command line



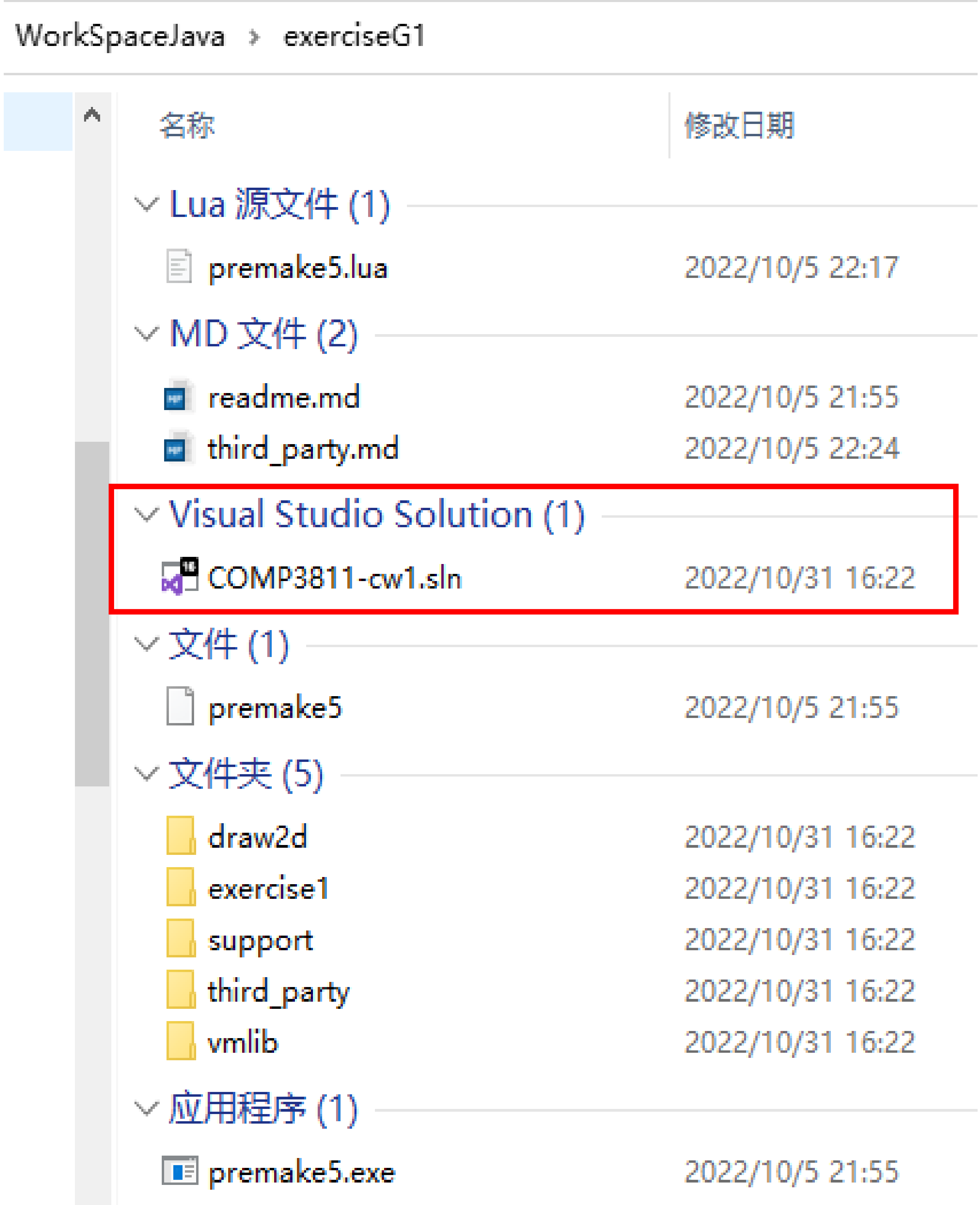
orkSpaceJava › exerciseG1

名称	修改日期
Lua 源文件 (1)	
premake5.lua	2022/10/5 22:17
MD 文件 (2)	
readme.md	2022/10/5 21:55
third_party.md	2022/10/5 22:24
文件 (1)	
premake5	2022/10/5 21:55
文件夹 (5)	
draw2d	2022/10/31 16:09
exercise1	2022/10/31 16:09
support	2022/10/31 16:09
third_party	2022/10/31 16:09
vmlib	2022/10/31 16:09
应用程序 (1)	
premake5.exe	2022/10/5 21:55



命令提示符

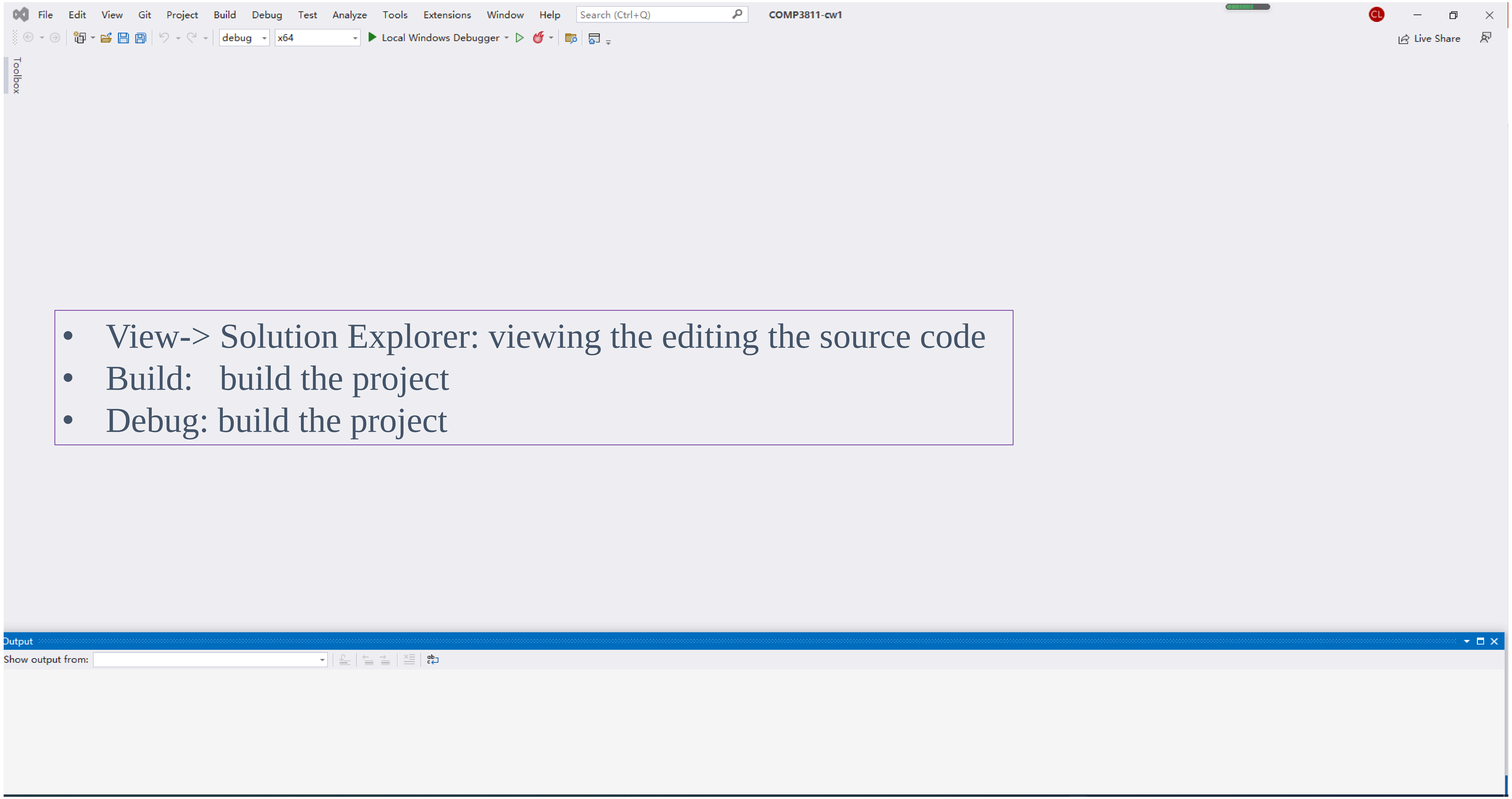
```
 E:\WorkSpaceJava\exerciseG1 的目录

2022/10/31  16:09    <DIR>          .
2022/10/31  16:09    <DIR>          ..
2022/10/31  16:09    <DIR>          draw2d
2022/10/31  16:09    <DIR>          exercise1
2022/10/05  21:55         2,207,408 premake5
2022/10/05  21:55         1,398,272 premake5.exe
2022/10/05  22:17             2,070 premake5.lua
2022/10/05  21:55               130 readme.md
2022/10/31  16:09    <DIR>          support
2022/10/31  16:09    <DIR>          third_party
2022/10/05  22:24             1,363 third_party.md
2022/10/31  16:09    <DIR>          vmlib
               5 个文件      3,609,243 字节
               7 个目录 973,131,657,216 可用字节

E:\WorkSpaceJava\exerciseG1>premake5.exe vs2022
Building configurations...
Running action 'vs2022'...
Generated COMP3811-cw1.sln...
Generated third_party/x-stb.vcxproj...
Generated third_party/x-stb.vcxproj.user...
Generated third_party/x-glad.vcxproj...
Generated third_party/x-glad.vcxproj.user...
Generated third_party/x-glfw.vcxproj...
Generated third_party/x-glfw.vcxproj.user...
Generated exercise1/exercise1.vcxproj...
Generated exercise1/exercise1.vcxproj.user...
Generated draw2d/draw2d.vcxproj...
Generated draw2d/draw2d.vcxproj.user...
Generated support/support.vcxproj...
Generated support/support.vcxproj.user...
Generated vmlib/vmlib.vcxproj...
Generated vmlib/vmlib.vcxproj.user...
Done (138ms).

E:\WorkSpaceJava\exerciseG1>
```



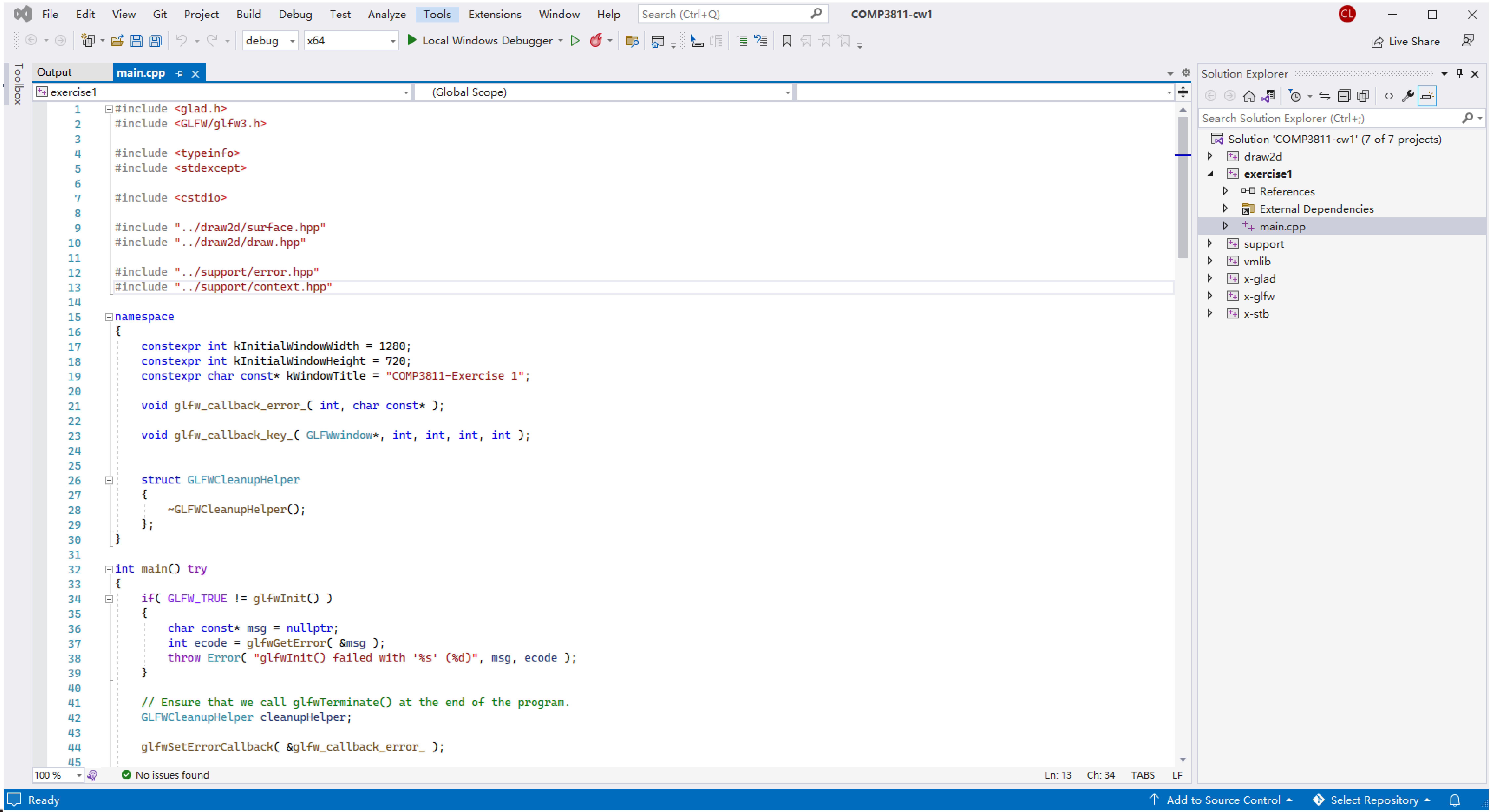
WorkSpaceJava › exerciseG1

名称	修改日期
Lua 源文件 (1)	
premake5.lua	2022/10/5 22:17
MD 文件 (2)	
readme.md	2022/10/5 21:55
third_party.md	2022/10/5 22:24
Visual Studio Solution (1)	
COMP3811-cw1.sln	2022/10/31 16:22
文件 (1)	
premake5	2022/10/5 21:55
文件夹 (5)	
draw2d	2022/10/31 16:22
exercise1	2022/10/31 16:22
support	2022/10/31 16:22
third_party	2022/10/31 16:22
vmlib	2022/10/31 16:22
应用程序 (1)	
premake5.exe	2022/10/5 21:55

Step 2: Developing under Visual Studio



- View-> Solution Explorer: viewing the editing the source code
- Build: build the project
- Debug: build the project



```cpp
#include <glad.h>
#include <GLFW/glfw3.h>

#include <typeinfo>
#include <stdexcept>

#include <cstdio>

#include "../draw2d/surface.hpp"
#include "../draw2d/draw.hpp"

#include "../support/error.hpp"
#include "../support/context.hpp"

namespace
{
    constexpr int kInitialWindowWidth = 1280;
    constexpr int kInitialWindowHeight = 720;
    constexpr char const* kWindowTitle = "COMP3811-Exercise 1";

    void glfw_callback_error_( int, char const* );

    void glfw_callback_key_( GLFWwindow*, int, int, int, int );


    struct GLFWCleanupHelper
    {
        ~GLFWCleanupHelper();
    };
}

int main() try
{
    if( GLFW_TRUE != glfwInit() )
    {
        char const* msg = nullptr;
        int ecode = glfwGetError( &msg );
        throw Error( "glfwInit() failed with '%s' (%d)", msg, ecode );
    }

    // Ensure that we call glfwTerminate() at the end of the program.
    GLFWCleanupHelper cleanupHelper;

    glfwSetErrorCallback( &glfw_callback_error_ );

```