Set up OpenGL

107-2電腦圖學

Outline

- 下載GLEW、GLFW
- 環境設置
- 執行測試

下載GLEW、GLFW

下載GLEW

http://glew.sourceforge.net/

Latest Release: 2.1.0

Download
Usage
Building
Installation
Source Generation
Change Log

GitHub
Issues
Pull Requests
Authors
Licensing

SourceForge Page

Last Update: 07-31-17

OpenGL
GitHub
Travis CI
SOURCEFORGE

The OpenGL Extension Wrangler Library

The OpenGL Extension Wrangler Library (GLEW) is a cross-platform open-source C/C++ extension loading library. GLEW provides efficient run-time mechanisms for determining which OpenGL extensions are supported on the target platform. OpenGL core and extension functionality is exposed in a single header file. GLEW has been tested on a variety of operating systems, including Windows, Linux, Mac OS X, FreeBSD, Irix, and Solaris.

Downloads

GLEW is distributed as source and precompiled binaries.
The latest release is 2.1.0[07-31-17]:

Source	ZIP | TGZ
Binaries	Windows 32-bit and 64-bit

An up-to-date copy is also available using git:

- github
  `git clone https://github.com/nigels-com/glew.git glew`

Supported Extensions

The latest release contains support for OpenGL 4.6, compatibility and forward-compatible contexts and the following extensions:

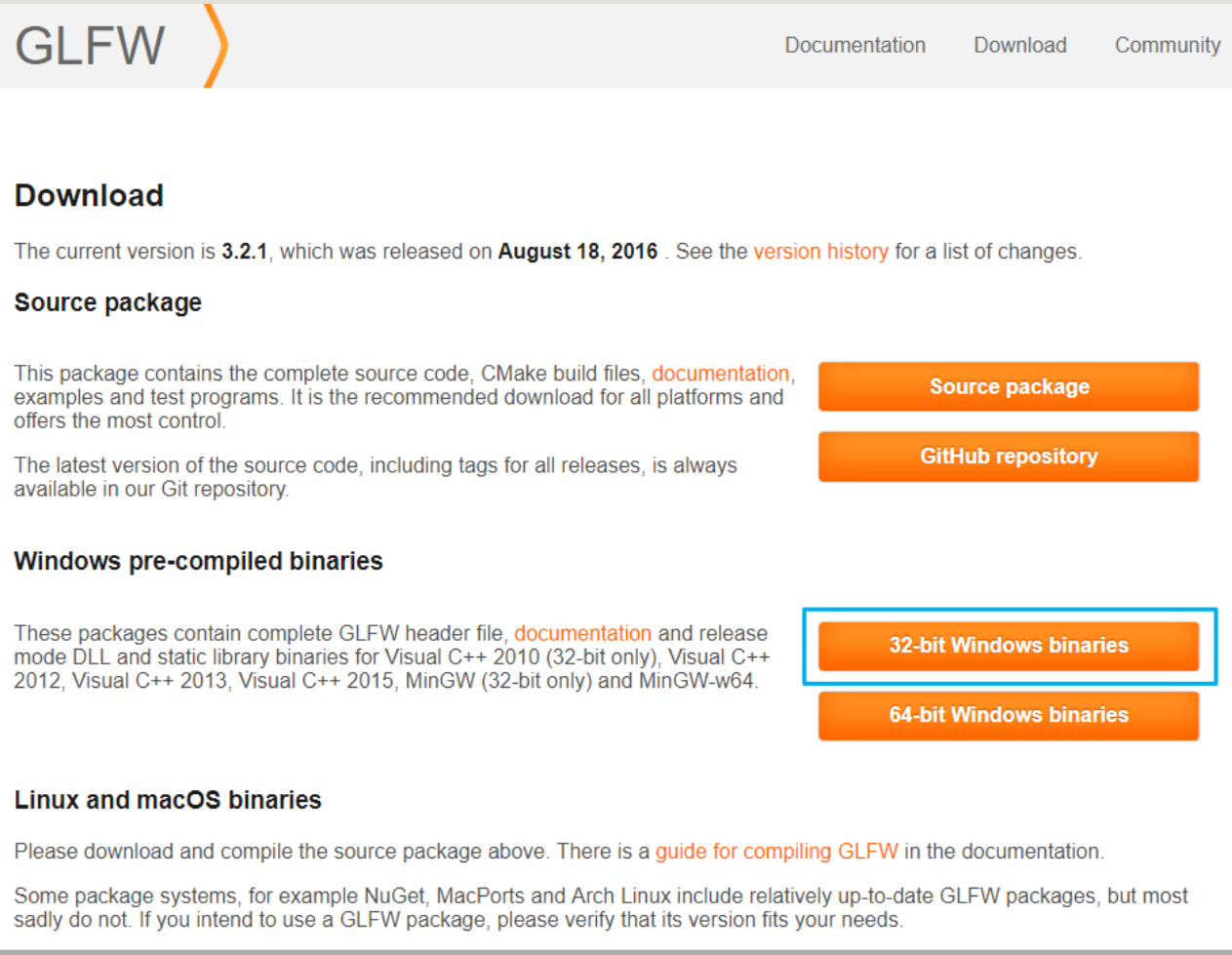
- OpenGL extensions
- WGL extensions
- GLX extensions

News

- [07-31-17] GLEW 2.1.0 adds support for OpenGL 4.6, new extensions and minor bug fixes
- [07-24-16] GLEW 2.0.0 adds support for forward-compatible contexts, adds new extensions, OSMesa and EGL support, MX discontinued and minor bug fixes
- [08-10-15] GLEW 1.13.0 adds support for new extensions, fixes minor bugs
- [26-01-15] GLEW 1.12.0 fixes minor bugs and adds new extensions
- [08-11-14] GLEW 1.11.0 adds support for OpenGL 4.5, new extensions
- [07-22-13] GLEW 1.10.0 adds support for OpenGL 4.4, new extensions
- [08-06-12] GLEW 1.9.0 adds support for OpenGL 4.3, new extensions

下載GLFW

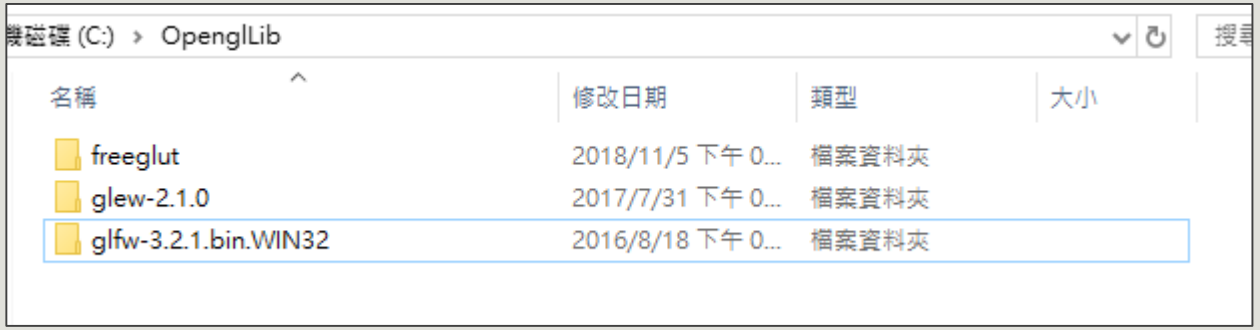
https://www.glfw.org/download.html



環境設置

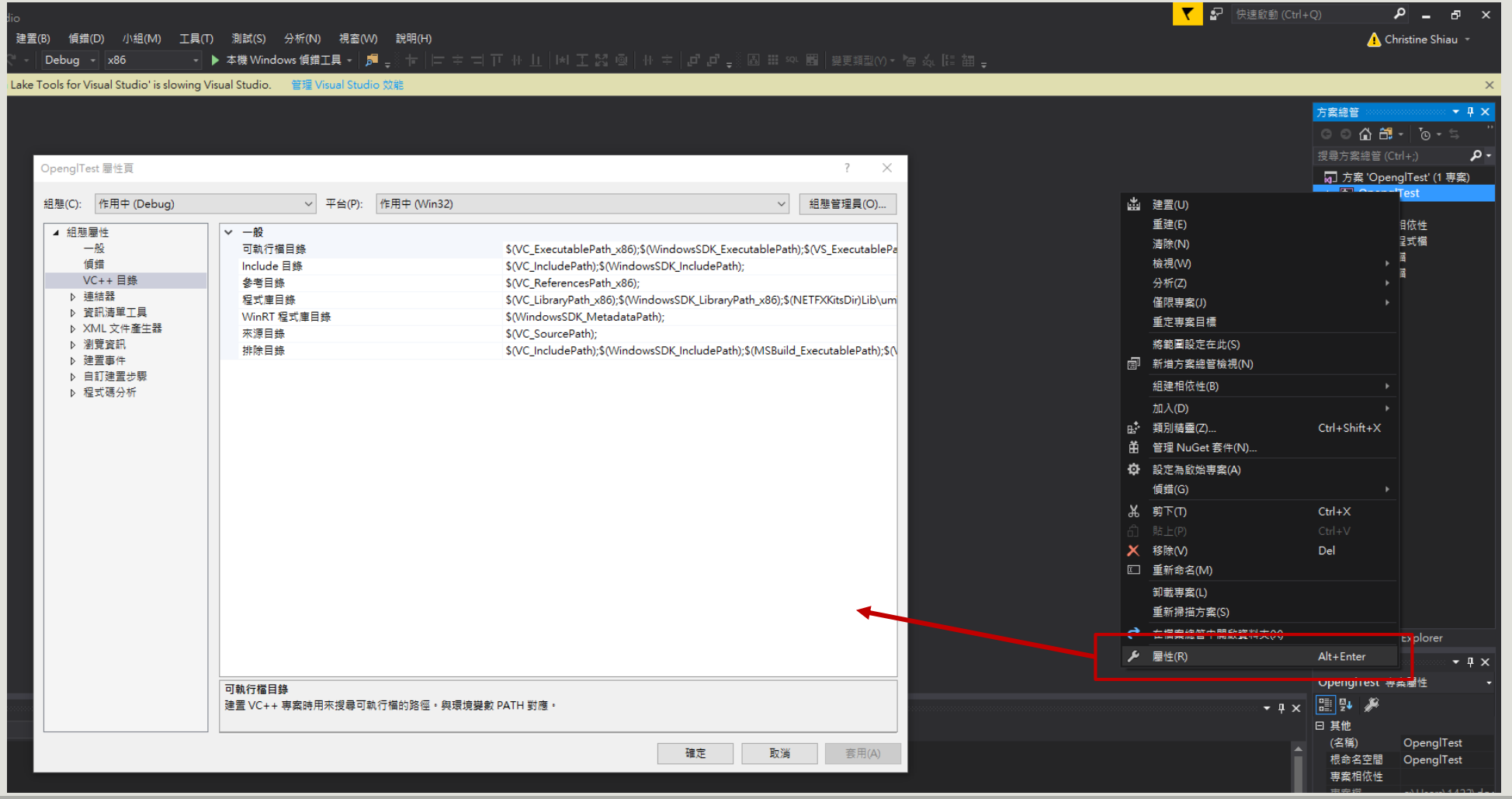
前置作業

解壓縮glew及glfw



VS環境設置為Debug x86

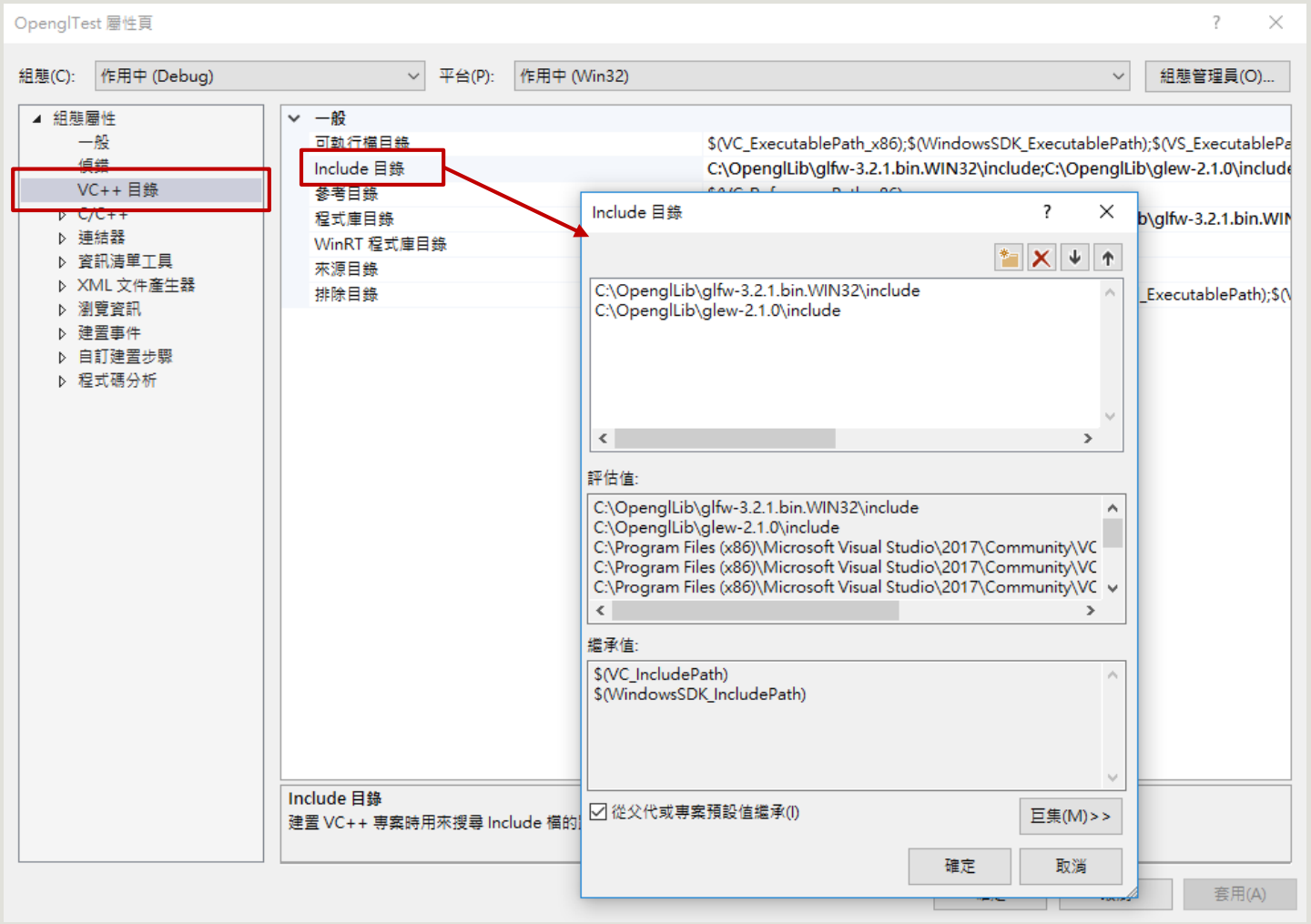
設定專案屬性


建置(B) 偵錯(D) 小組(M) 工具(T) 測試(S) 分析(N) 視窗(W) 說明(H)
Debug x86 本機 Windows 偵錯工具
Lake Tools for Visual Studio' is slowing Visual Studio. 管理 Visual Studio 效能
OpenglTest 屬性頁
組態(C): 作用中 (Debug) 平台(P): 作用中 (Win32) 組態管理員(O)...
組態屬性
一般
偵錯
VC++ 目錄
連結器
資訊清單工具
XML 文件產生器
瀏覽資訊
建置事件
自訂建置步驟
程式碼分析
一般
可執行檔目錄 $(VC_ExecutablePath_x86);$(WindowsSDK_ExecutablePath);$(VS_ExecutablePa
Include 目錄 $(VC_IncludePath);$(WindowsSDK_IncludePath);
參考目錄 $(VC_ReferencesPath_x86);
程式庫目錄 $(VC_LibraryPath_x86);$(WindowsSDK_LibraryPath_x86);$(NETFXKitsDir)Lib\um
WinRT 程式庫目錄 $(WindowsSDK_MetadataPath);
來源目錄 $(VC_SourcePath);
排除目錄 $(VC_IncludePath);$(WindowsSDK_IncludePath);$(MSBuild_ExecutablePath);$(
可執行檔目錄
建置 VC++ 專案時用來搜尋可執行檔的路徑，與環境變數 PATH 對應。
確定 取消 套用(A)
方案總管
方案 'OpenglTest' (1 專案)
建置(U)
重建(E)
清除(N)
檢視(W)
分析(Z)
僅限專案(J)
重定專案目標
將範圍設定在此(S)
新增方案總管檢視(N)
組建相依性(B)
加入(D)
類別精靈(Z)... Ctrl+Shift+X
管理 NuGet 套件(N)...
設定為啟始專案(A)
偵錯(G)
剪下(T) Ctrl+X
貼上(P) Ctrl+V
移除(V) Del
重新命名(M)
卸載專案(L)
重新掃描方案(S)
屬性(R) Alt+Enter
其他
(名稱) OpenglTest
根命名空間 OpenglTest
專案相依性

新增Include目錄

加入

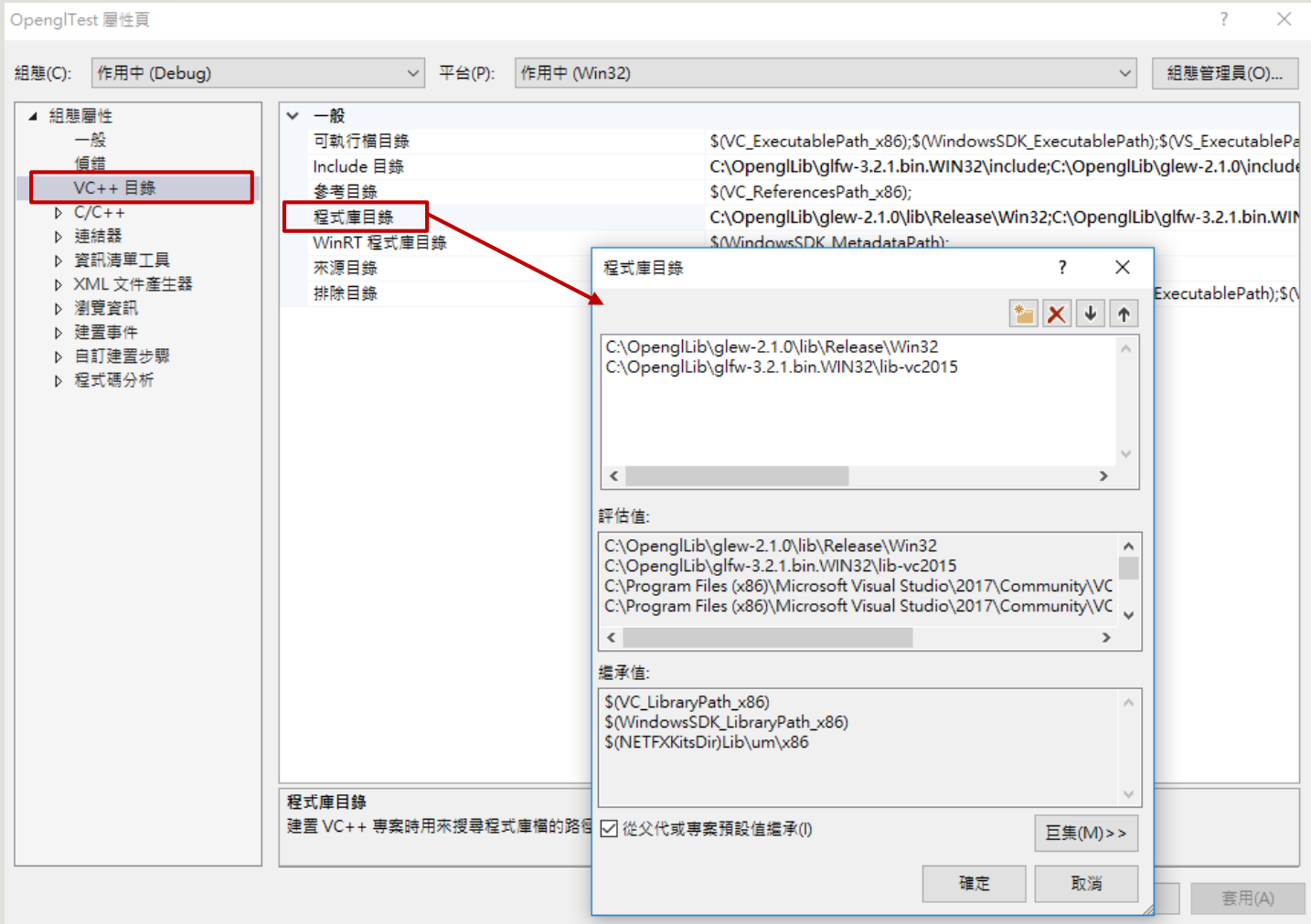
- glfw-3.2.1.bin.WIN32\include
- glew-2.1.0\include



新增程式庫目錄

加入

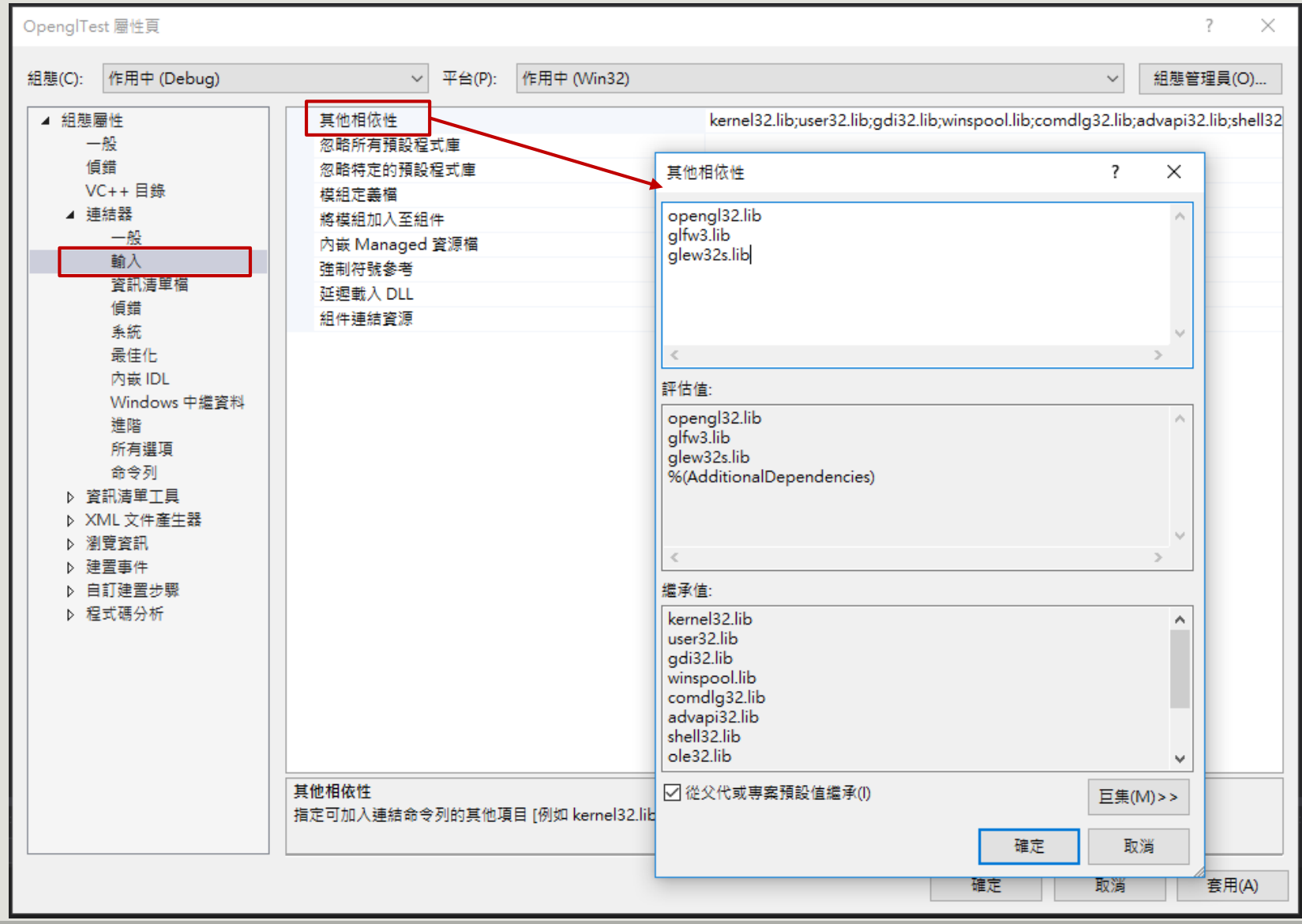
- glew-2.1.0\lib\Release\Win32
- glfw-3.2.1.bin.WIN32\lib-vc2015
  (2015以上版本請選擇 lib-vc2015)



設置鏈接器

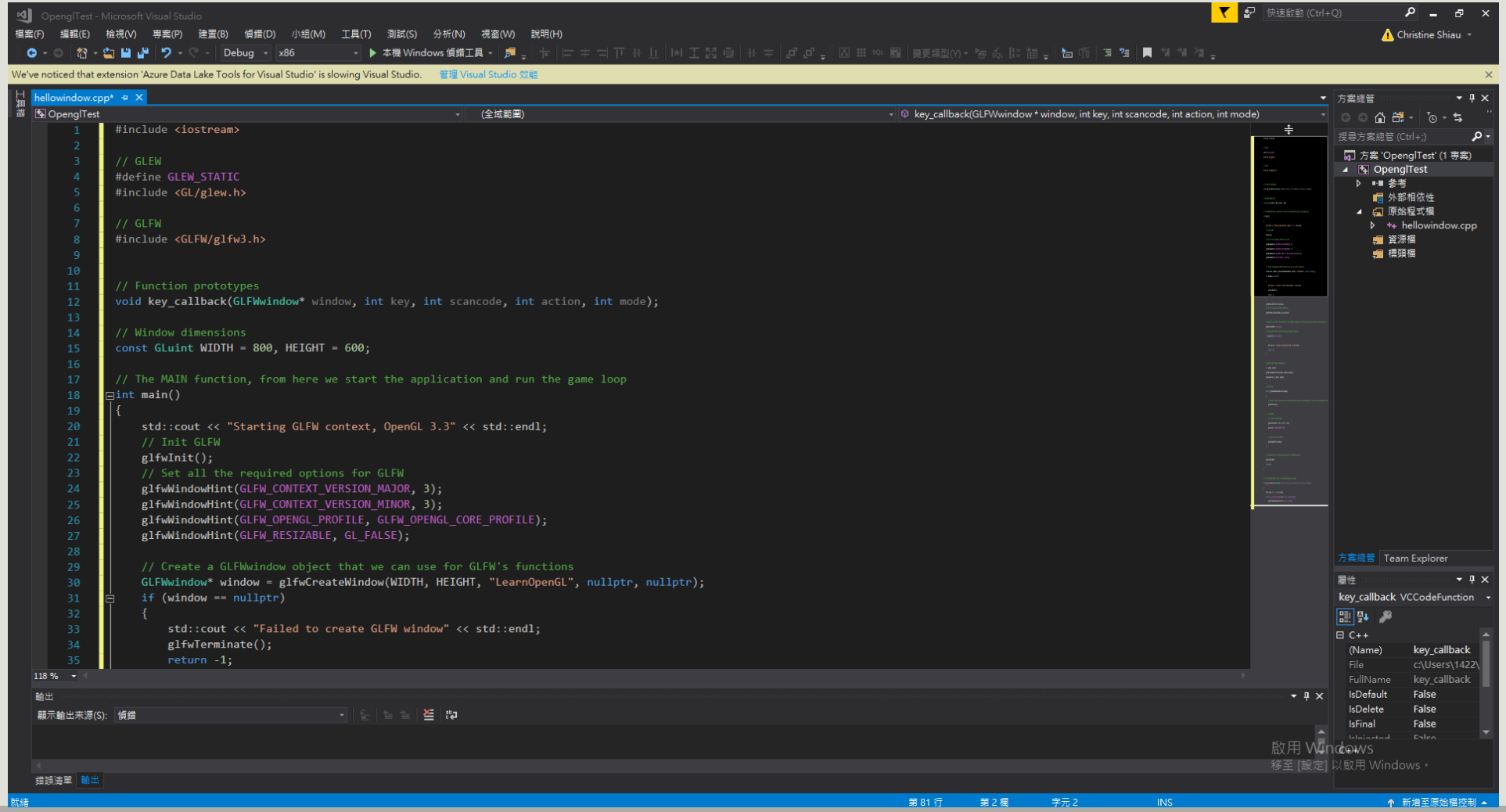
新增

- opengl32.lib
- glfw3.lib
- glew32s.lib

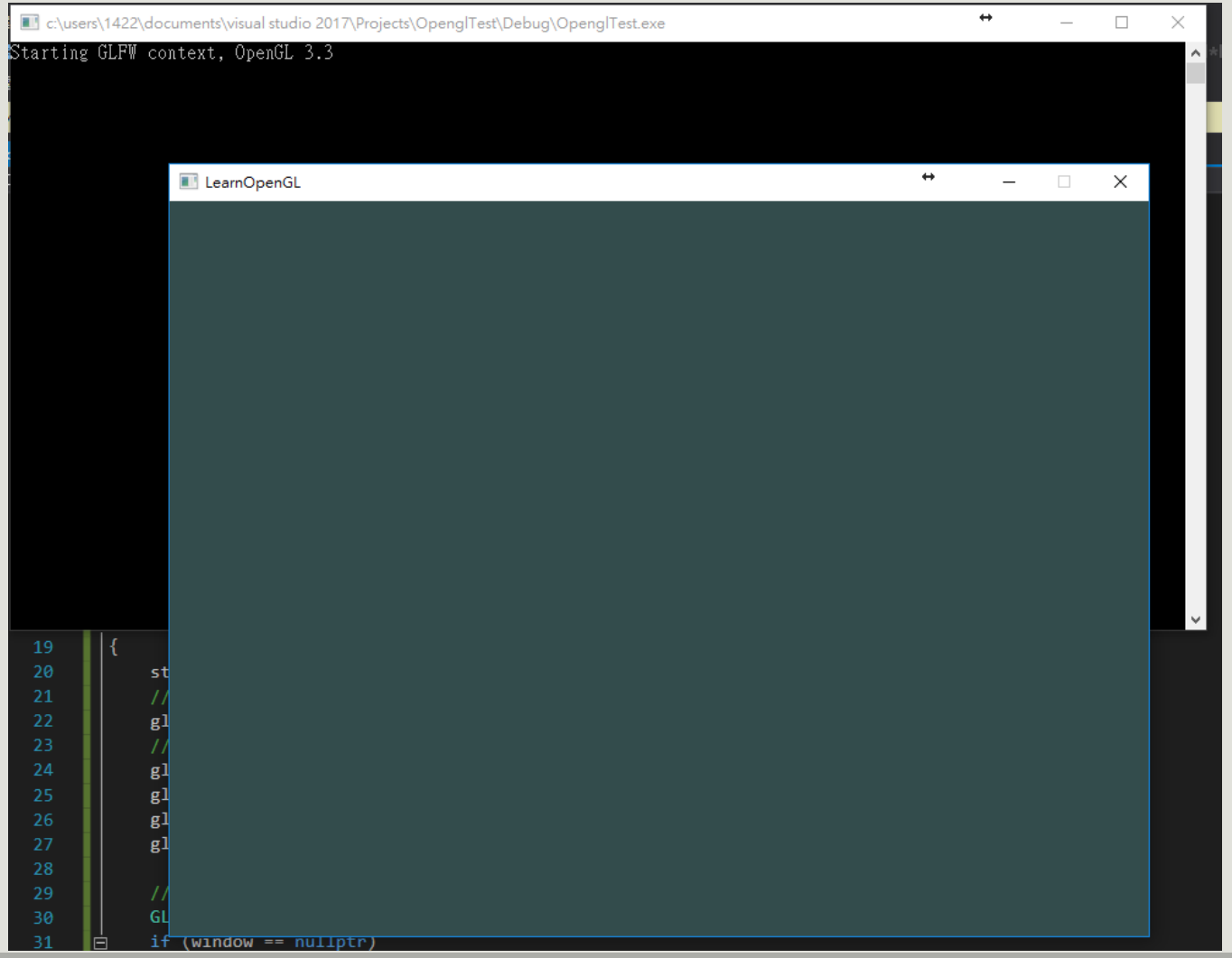


執行測試

執行範例程式


OpenglTest - Microsoft Visual Studio
檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 工具(T) 測試(S) 分析(N) 視窗(W) 說明(H)
Debug x86
本機 Windows 偵錯工具
We've noticed that extension 'Azure Data Lake Tools for Visual Studio' is slowing Visual Studio.
管理 Visual Studio 效能
hellowindow.cpp*
OpenglTest
(全域範圍)
key_callback(GLFWwindow * window, int key, int scancode, int action, int mode)
#include <iostream>
// GLEW
#define GLEW_STATIC
#include <GL/glew.h>
// GLFW
#include <GLFW/glfw3.h>
// Function prototypes
void key_callback(GLFWwindow* window, int key, int scancode, int action, int mode);
// Window dimensions
const GLuint WIDTH = 800, HEIGHT = 600;
// The MAIN function, from here we start the application and run the game loop
int main()
{
std::cout << "Starting GLFW context, OpenGL 3.3" << std::endl;
// Init GLFW
glfwInit();
// Set all the required options for GLFW
glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 3);
glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 3);
glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
glfwWindowHint(GLFW_RESIZABLE, GL_FALSE);
// Create a GLFWwindow object that we can use for GLFW's functions
GLFWwindow* window = glfwCreateWindow(WIDTH, HEIGHT, "LearnOpenGL", nullptr, nullptr);
if (window == nullptr)
{
std::cout << "Failed to create GLFW window" << std::endl;
glfwTerminate();
return -1;
方案總管
方案 'OpenglTest' (1 專案)
OpenglTest
參考
外部相依性
原始程式檔
hellowindow.cpp
資源檔
標頭檔
Team Explorer
key_callback VCCodeFunction
(Name) key_callback
FullName key_callback
IsDefault False
IsDelete False
IsFinal False
輸出
顯示輸出來源(S): 偵錯
錯誤清單 輸出
就緒
第 81 行 第 2 欄 字元 2 INS

執行結果


c:\users\1422\documents\visual studio 2017\Projects\OpenglTest\Debug\OpenglTest.exe
Starting GLFW context, OpenGL 3.3
LearnOpenGL
19 {
20 st
21 //
22 gl
23 //
24 gl
25 gl
26 gl
27 gl
28
29 //
30 GL
31 if (window == nullptr)