

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light greenish-blue. They are both tilted at an angle.

OpenGL Environment Setup

Homework 0 - 2022 Computer Graphics



What is OpenGL

- an multi-platform graphics API
 - it requires a language(C/C++) to operate in developed
 - can interact with a GPU to accelerate rendering



IDE & Kit

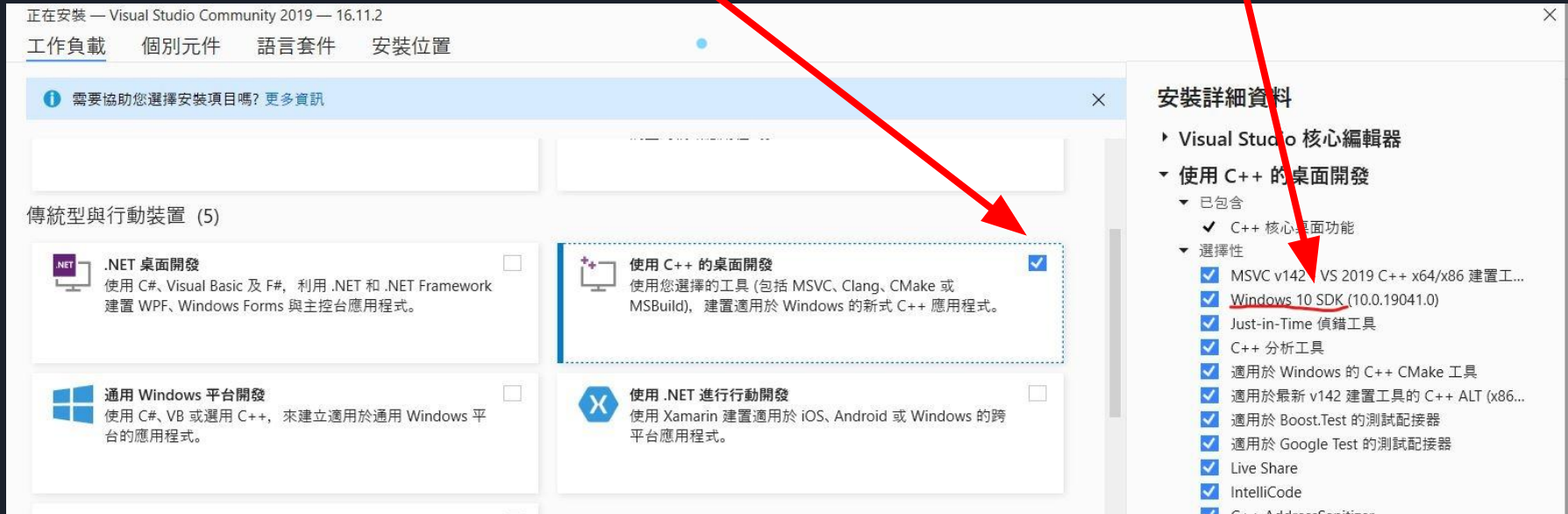
- Visual Studio 2019 - Community(VS2019) ← **REQUIRED**
- GLAD 2 ← provided in HW zip (extern/glad)
 - an OpenGL loading library that **loads pointers to OpenGL functions at runtime**, core as well as extensions.
 - a header file for all OpenGL API and constants is generated by glad2 (extern/glad/include/glad/gl.h)
- GLFW ← provided in HW zip (extern/glfw)
 - an Open Source, multi-platform library for OpenGL, OpenGL ES and Vulkan development on the desktop.
 - It provides a simple API **for creating windows**, contexts and surfaces, receiving input and events.

Download Visual Studio

- Download and launch visual studio installer

Desktop development for c++ is **required**

select latest Windows SDK
(SDK contains opengl32.lib)



Open Project with Visual Studio

- Install Visual Studio 2019
- Download hw0.zip & unzip
- Open file “vs2019/HW0.sln” directly

開始使用



複製存放庫(C)

從像是 GitHub 或 Azure DevOps 等這類線上存放庫取得程式碼



開啟專案或解決方案(P)

開啟本機 Visual Studio 專案或 .sln 檔案



開啟本機資料夾(F)

瀏覽和編輯任何資料夾內的程式碼



建立新的專案(N)

透過程式碼 Scaffolding 選擇專案範本以開始使用

不使用程式碼繼續(W) →

extern
include
lib
out
script
src
vs2019
.clang-format
.gitignore
.gitmodules
CMakeLists.txt
LICENSE
README.md

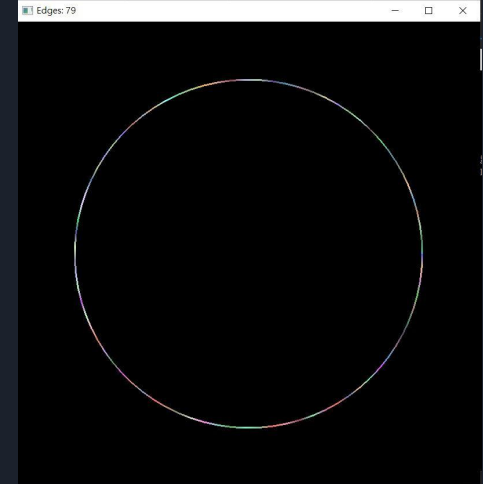
.vs
glfw
HW0.sln
HW0.vcxproj
HW0.vcxproj.filters
HW0.vcxproj.user

Open Project with Visual Studio

- Select config then build (ctrl+shift+B)
- Use F5 to debug or ctrl+F5 to run.



- See the result.
 - in HWO, you will see an animation
 - press ESC to terminate the process





Working on other platforms

- **Warning: You may encounter weird bugs.**
 - You need [CMake](#) and a compiler to build.
 - See README.md for details. →
-
- only compile with cmake on these platforms are tested
 - macOS Monterey (12)
 - Ubuntu 20.04

Dependencies for Windows

Visual Studio

Dependencies for macOS

Xcode

Dependencies for Unix-like systems other than macOS with X11

On Debian and derivatives like Ubuntu and Linux Mint

```
sudo apt install xorg-dev
```

On Fedora and derivatives like Red Hat

```
sudo dnf install libxcursor-devel libXi-devel libXinerama-devel libXrandr-devel
```

On FreeBSD

```
pkg install xorgproto
```

Build instruction

CMake

Build in release mode

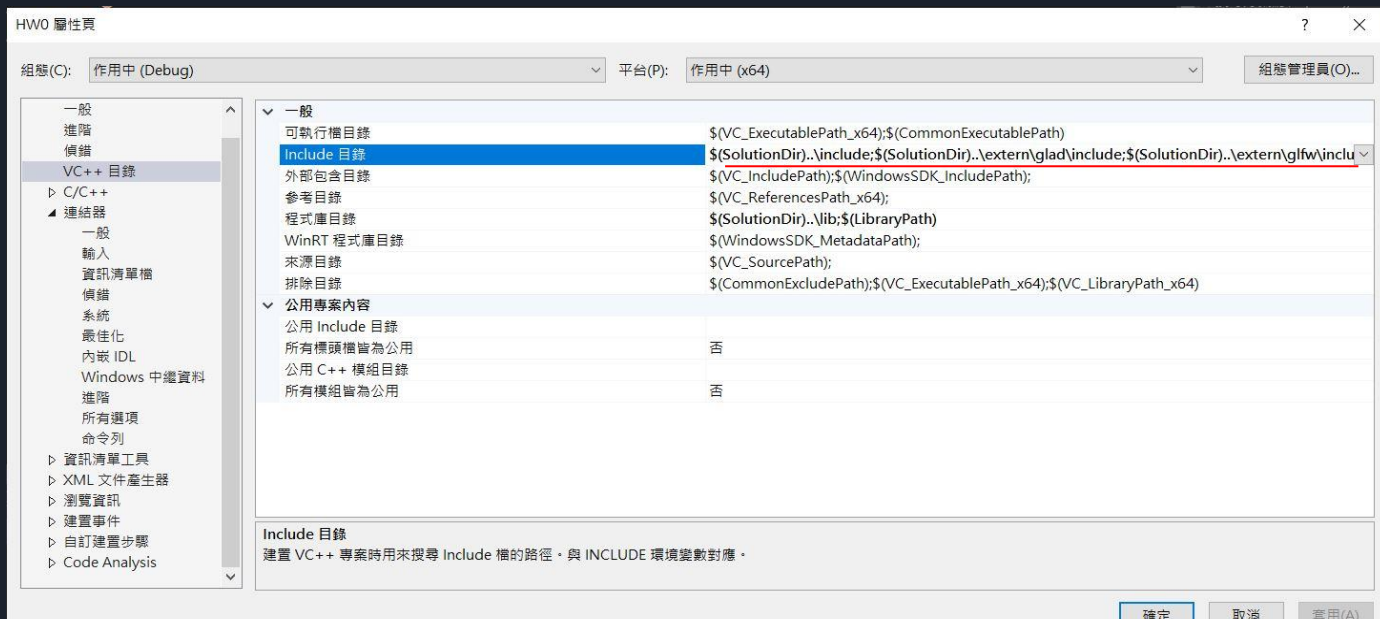
```
cmake -S . -B build -D CMAKE_BUILD_TYPE=Release
cmake --build build --config Release --parallel 8
cd bin
./Hw0
```

Build in debug mode

```
cmake -S . -B build -D CMAKE_BUILD_TYPE=Debug
cmake --build build --config Debug --parallel 8
cd bin
./Hw0
```

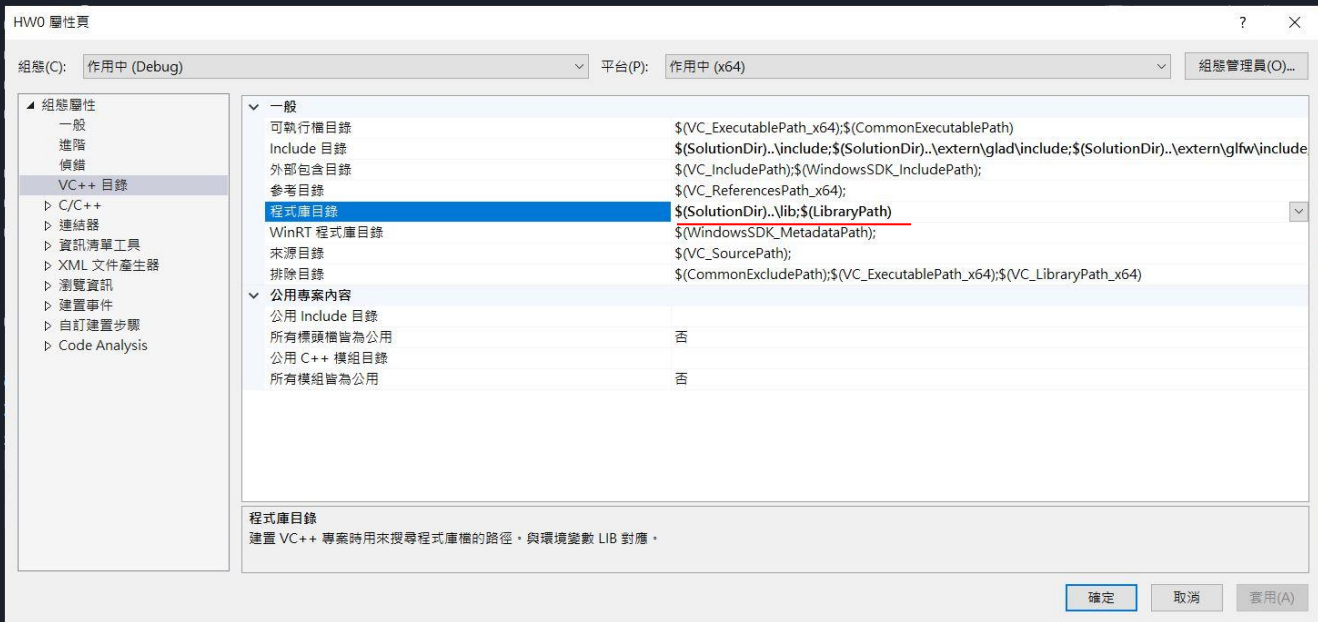
Appendix: Reminders for Creating a New Project

- Include Windows SDK, GLAD(glad/gl.h), GLFW(GLFW/glfw3.h) & your own .h folder
 - in HW0, we put gl.h & glfw3.h in "extern" directory



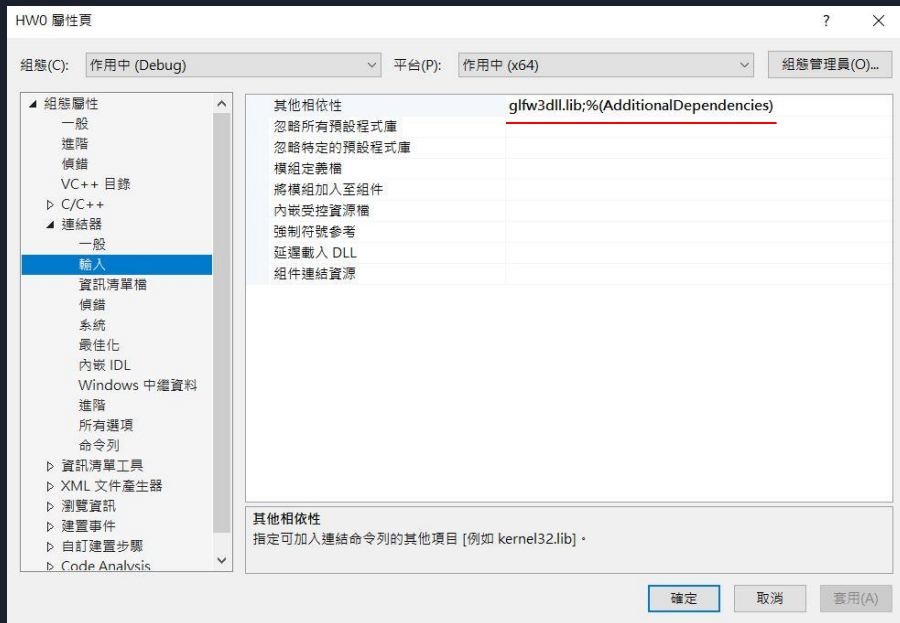
Appendix: Reminders for Creating a New Project

- Check library(ex: glfw3dll.lib, glfw3.lib) path added to project setting
 - in HW0, we put glfw3dll.lib in “lib” directory
 - Win10 for example



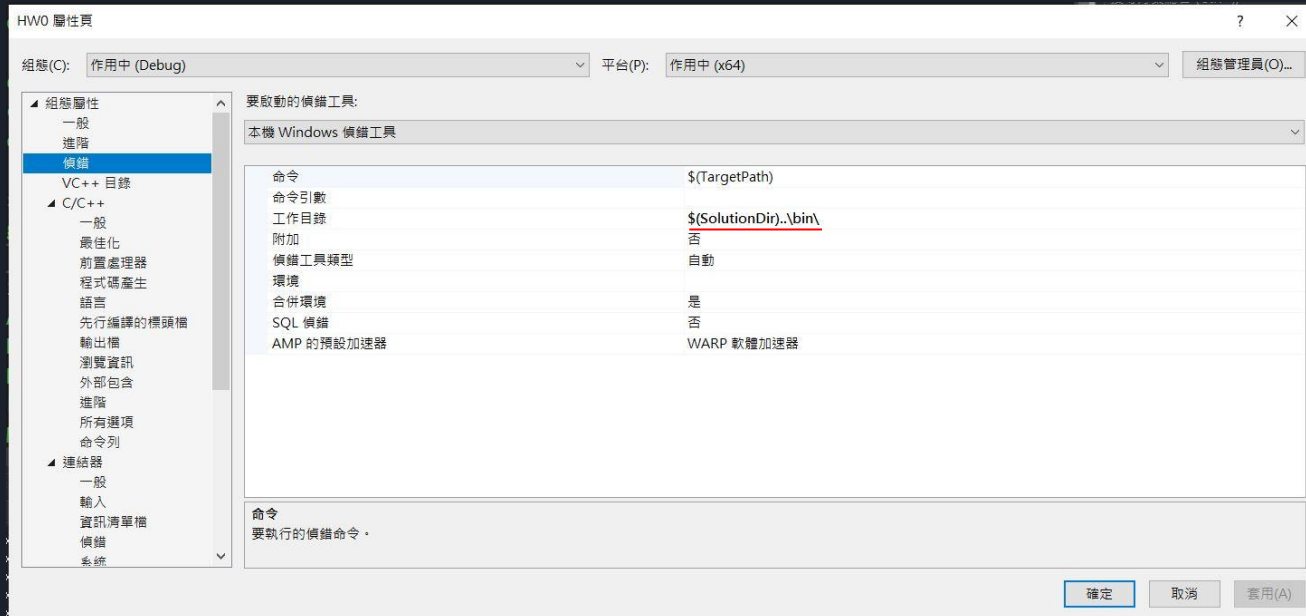
Appendix: Reminders for Creating a New Project

- Check “Linker > Input > Additional Dependencies” links glfw3dll.lib



Appendix: Reminders for Creating a New Project

- Check DLL
 - in HW0, we put glfw3.dll in “bin” directory





Appendix: Useful Reference Links

- <https://www.khronos.org/registry/OpenGL-Refpages/>
 - <https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/>
 - <https://www.khronos.org/registry/OpenGL-Refpages/gl4/>
- <https://www.glfw.org/docs/latest/>
- <https://github.com/g-truc/glm/blob/master/manual.md>