

M1 Mac OpenGL 설정

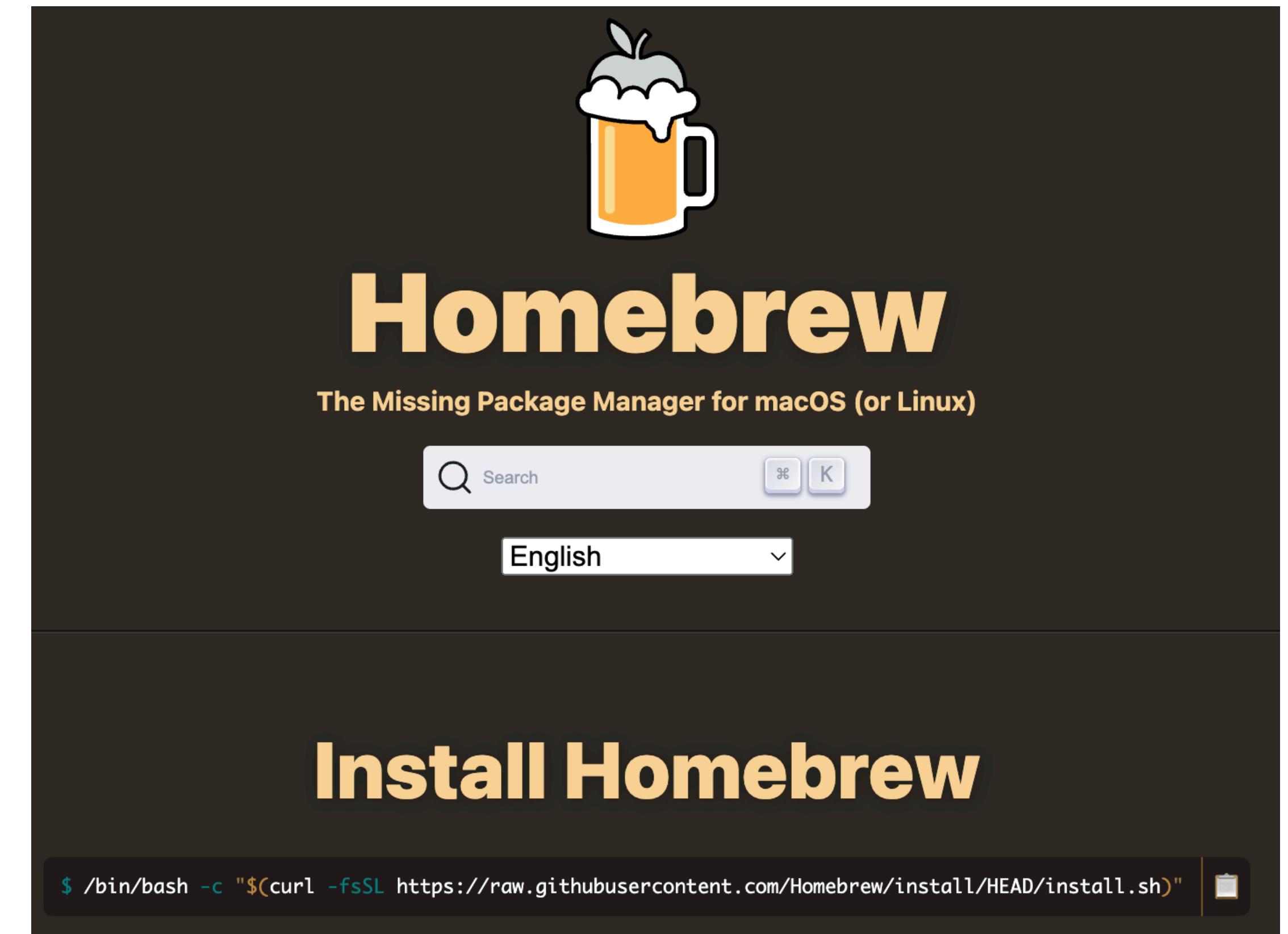
240110 19101188 고은수

Glfw 설치

homebrew를 이용하여 glfw를 설치합니다

homebrew : <https://brew.sh/>

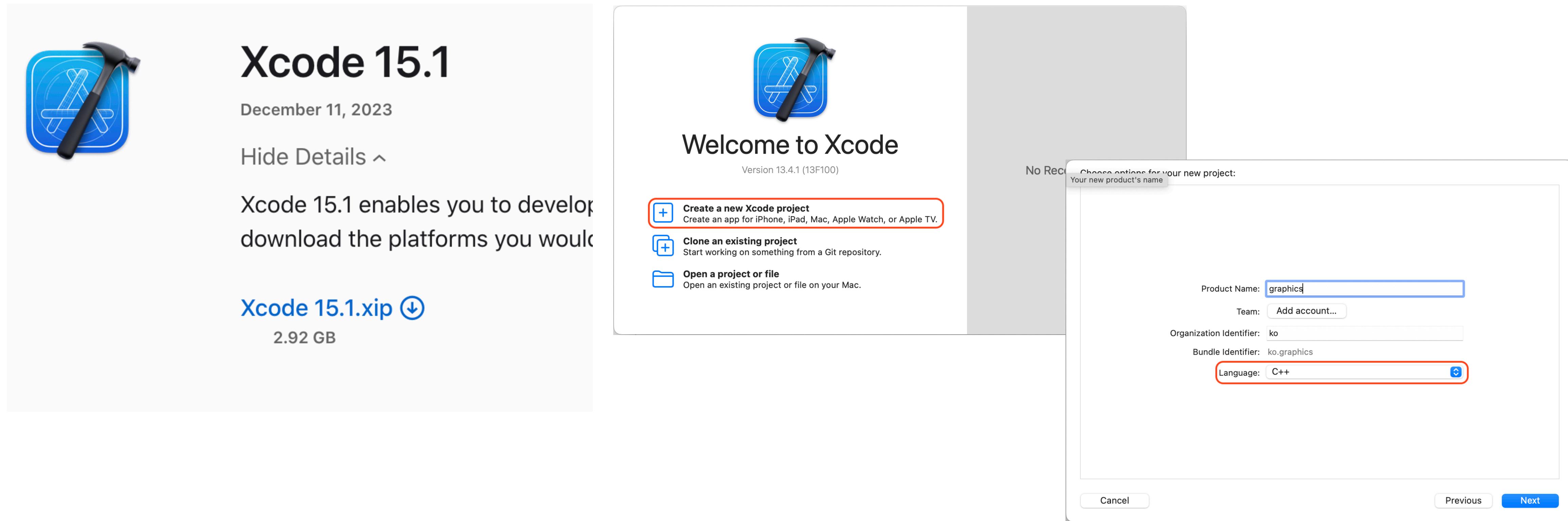
- 터미널에서 brew install glfw 명령어 실행
- 실패하면 arch -arm64 brew install glfw



Xcode 프로젝트 생성

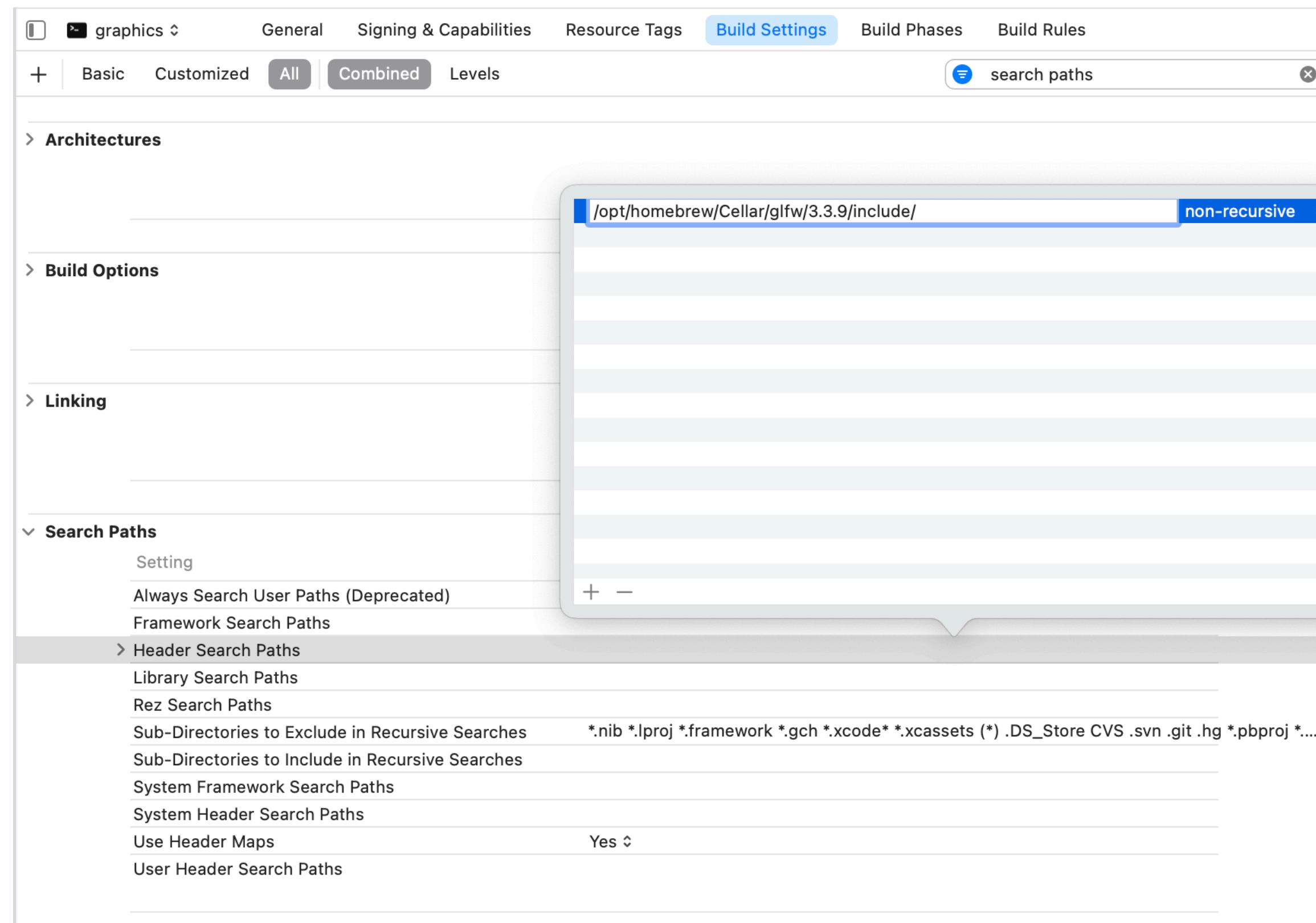
애플 개발자 사이트 developer.apple.com/download/more/ 에서 Xcode 설치

Xcode에서 Mac > Command line tool 선택후 원하는 디렉토리에 프로젝트 생성.



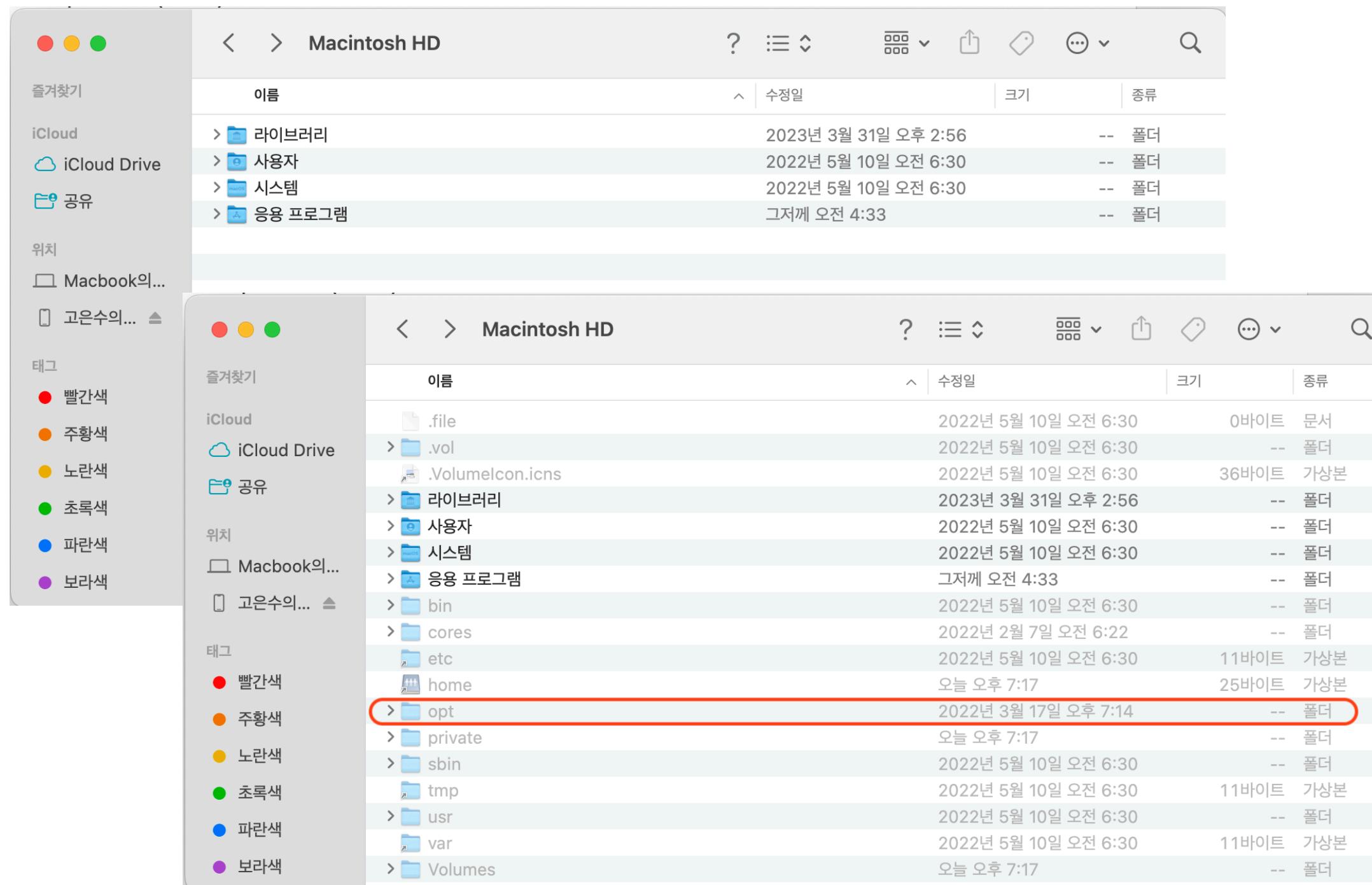
Header Search Paths 추가

Building settings > Search Paths > Header Search Path에 /opt/homebrew/Cellar/glfw/
설치한 버전/include/ 추가

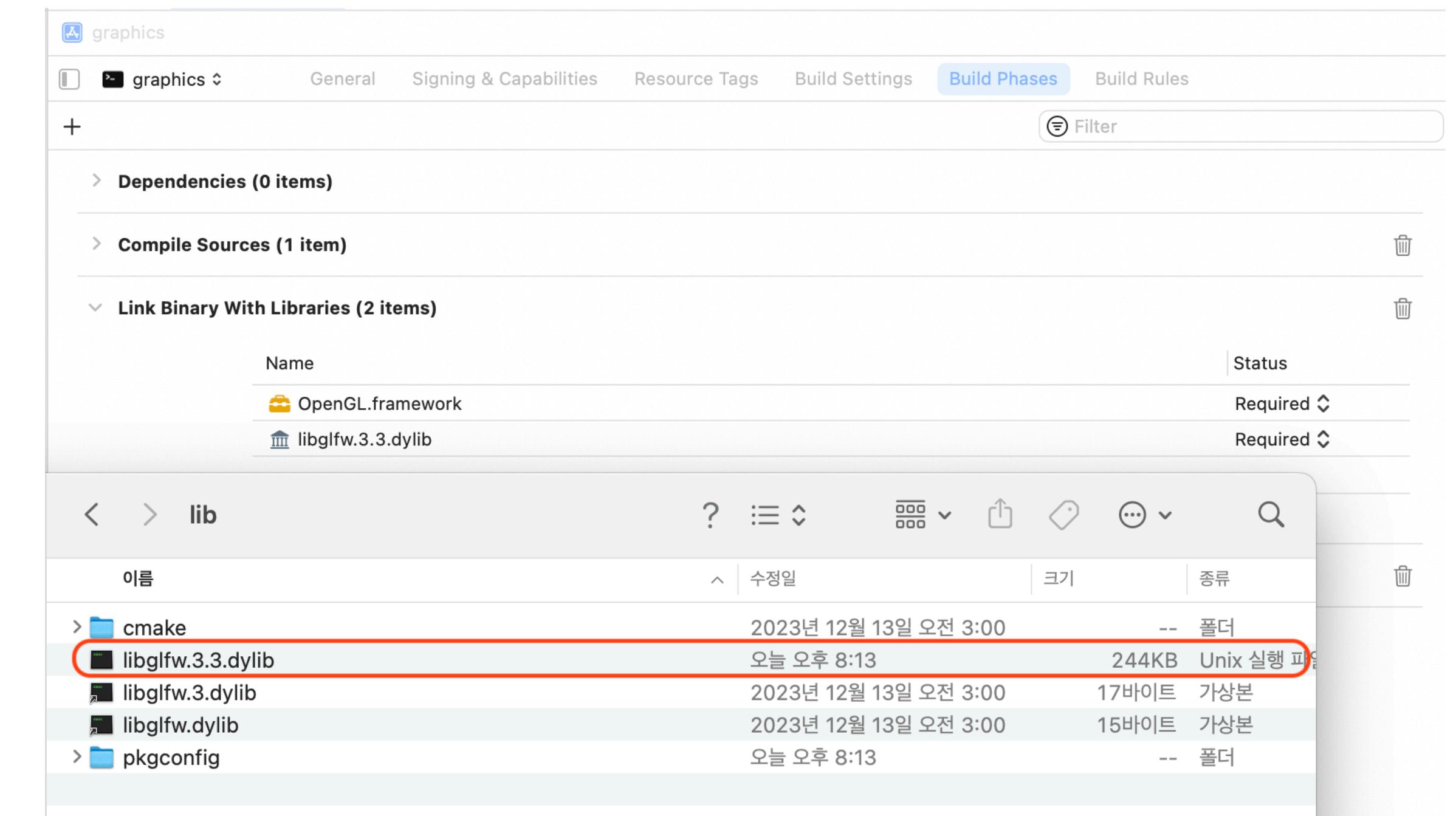


라이브러리 연결

Build Phases > Link Binary With Libraries 항목에 /opt/homebrew/Cellar/glfw/**설치한 버전**/lib/libglfw.3.3.dylib 과 OpenGL.framework를 추가. 아래와 같이 두 개의 라이브러리 연결

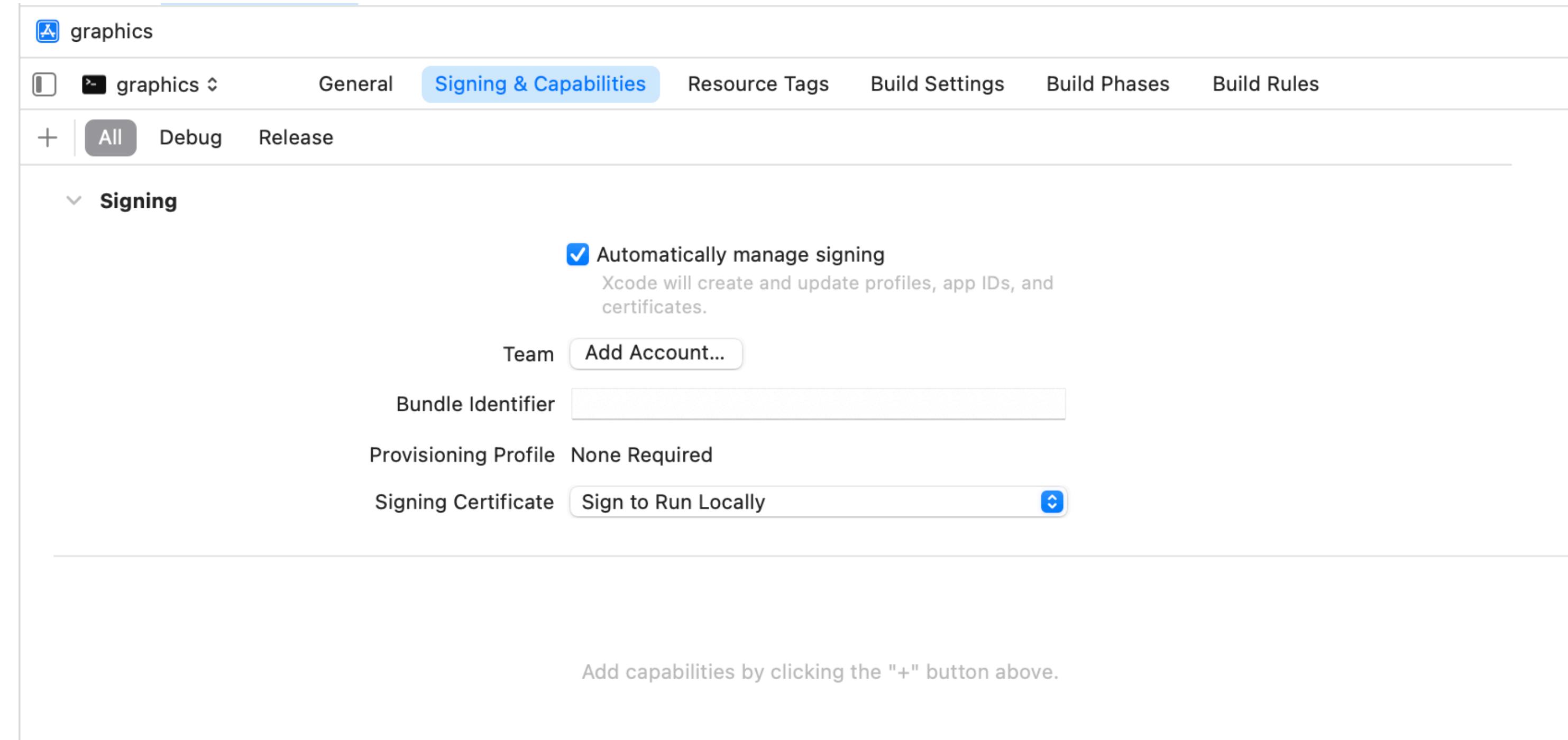


shift + cmd + . 으로 숨겨진 폴더 opt 접근 가능



Hardened Runtime 비활성화

Signing & Capabilities에서 Hardened Runtime을 우측 휴지통 아이콘을 클릭해서 아래와 같이 비활성화. (현재 적용된 Capability 목록에 Hardened Runtime 항목이 없어야 함.)



예제 코드 빌드

예제 코드 실행 : <https://www.glfw.org/documentation.html>

The screenshot shows the Xcode IDE interface. On the left, the 'Build Issues' tab displays 24 build issues, mostly related to deprecated OpenGL usage and migration to Metal. The main editor window shows the 'main.c' file with GLFW code. A warning for 'glClear' is highlighted, stating it is deprecated. To the right, a terminal window shows the application output: 'Hello World' followed by a timestamp and 'Metal API Validation Enabled'.

```
#include <GLFW/glfw3.h>
int main(void)
{
    GLFWwindow* window;
    /* Initialize the library */
    if (!glfwInit())
        return -1;

    /* Create a windowed mode window and its OpenGL context */
    window = glfwCreateWindow(640, 480, "Hello World", NULL, NULL);
    if (!window)
    {
        glfwTerminate();
        return -1;
    }

    /* Make the window's context current */
    glfwMakeContextCurrent(window);

    /* Loop until the user closes the window */
    while (!glfwWindowShouldClose(window))
    {
        /* Render here */
        glClear(GL_COLOR_BUFFER_BIT); // 'glClear' is deprecated: first deprecated in macOS 10.14 - OpenGL A...
        /* Swap front and back buffers */
        glfwSwapBuffers(window);

        /* Poll for and process events */
        glfwPollEvents();
    }
}
```

2024-01-07 20:41:36.029132+0900 graphics[4627:77777] Metal API Validation Enabled

Static 방식으로 빌드

GLFW 소스코드 다운로드

- git clone <https://github.com/glfw glfw.git>

- cd glfw

The screenshot shows the GitHub repository page for 'glfw'. The repository is public and has 388 watchers, 4.8k forks, and 11.6k stars. It features 17 branches and 20 tags. The 'Code' tab is selected. A list of recent commits is displayed, including changes from 'dougbinks' and others across various files like .github, CMake, deps, docs, examples, include/GLFW, src, tests, .appveyor.yml, and .gitattributes. The repository is described as a multi-platform library for OpenGL, OpenGL ES, Vulkan, window and input. It supports platforms like windows, macos, linux, unix, opengl, vulkan, and opengl-es.

Author	Commit Message	Date
dougbinks	Clarify documentation for contributing	b4c3ef9 · 2 weeks ago
	Update to actions/checkout@v3	last year
	Win32: Add library name suffix to pkg-config file	2 months ago
	Update glad to include Vulkan 1.3	2 years ago
	Clarify documentation for contributing	2 weeks ago
	Add window hints for initial position	2 years ago
	Improve documentation relating to key tokens	last month
	Cocoa: Fix segfault querying joystick elements	last month
	Fix glfwinfo output of Vulkan layer spec version	10 months ago
	Add 'latest' branch to all CI builds	3 years ago
	Exclude CI and Git dotfiles from Git export	5 years ago

```
[ko@Macbookui-MacBookAir desktop % git clone https://github.com/glfw glfw.git
'glfw'에 복제합니다...
remote: Enumerating objects: 31292, done.
remote: Counting objects: 100% (405/405), done.
remote: Compressing objects: 100% (170/170), done.
remote: Total 31292 (delta 260), reused 330 (delta 224), pack-reused 30887
오브젝트를 받는 중: 100% (31292/31292), 16.00 MiB | 5.03 MiB/s, 완료.
델타를 알아내는 중: 100% (22095/22095), 완료.
[ko@Macbookui-MacBookAir desktop % cd glfw
ko@Macbookui-MacBookAir glfw % ]
```

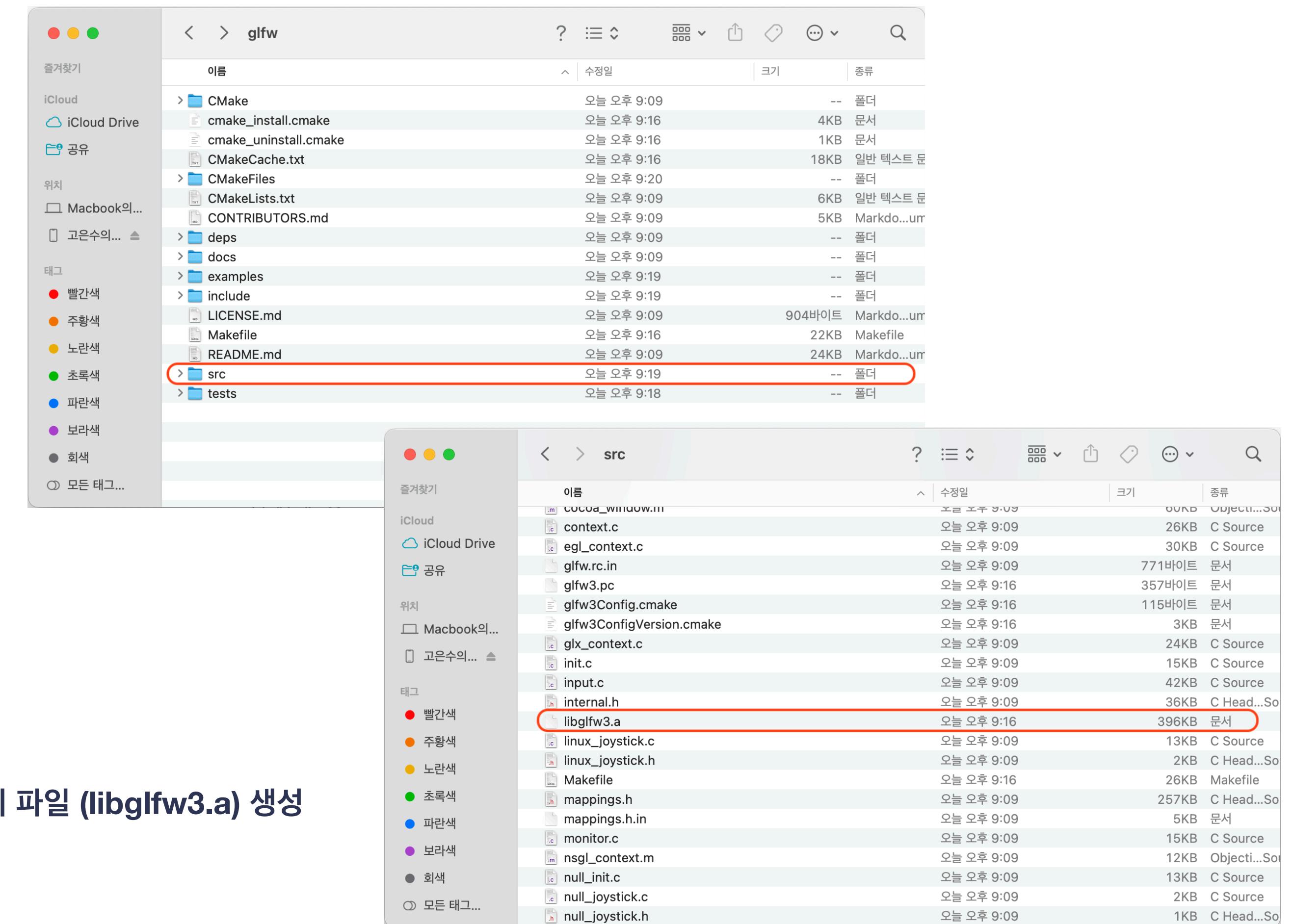
CMake 사용하여 라이브러리 빌드

- cmake -DBUILD_SHARED_LIBS=OFF .

- make

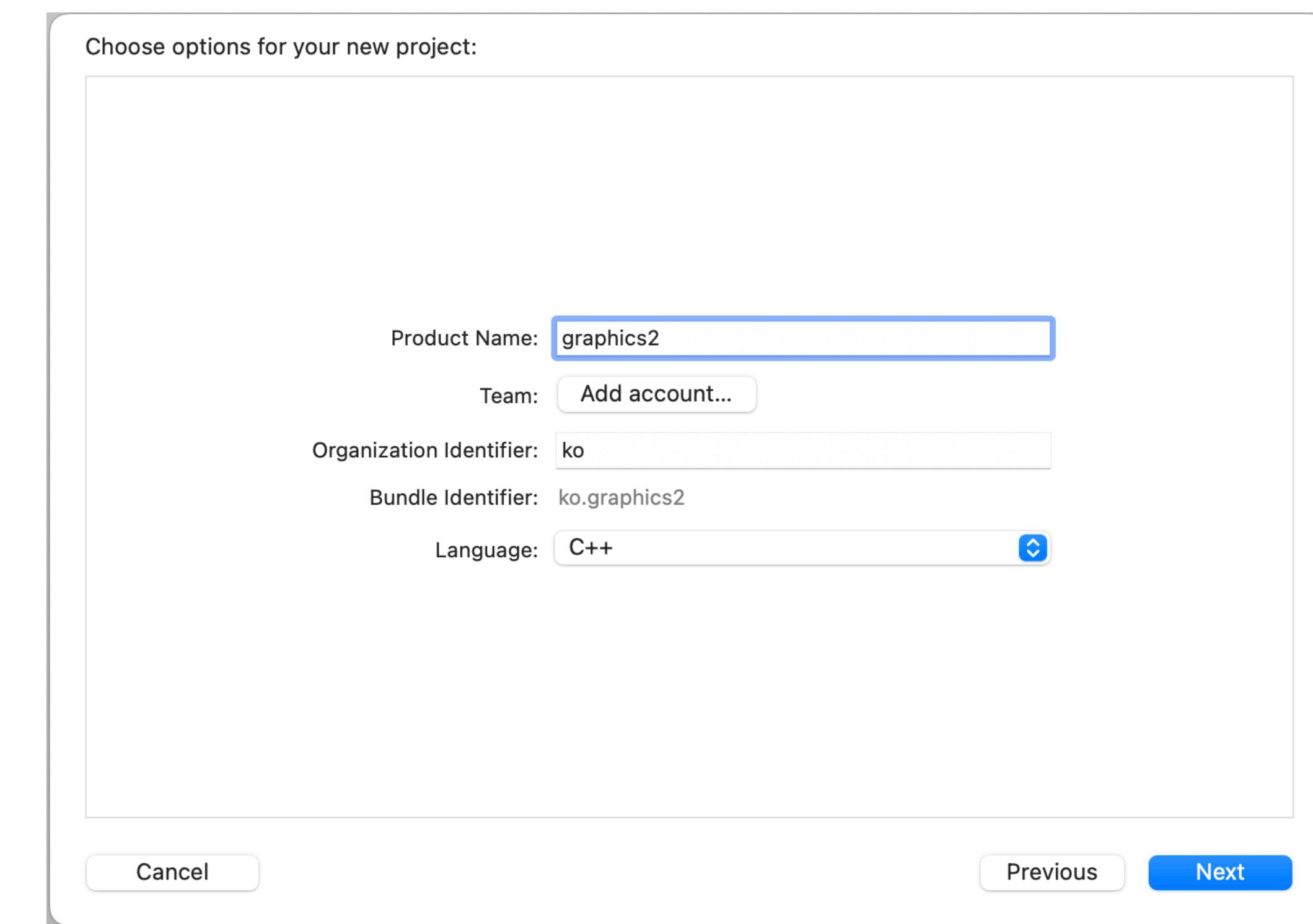
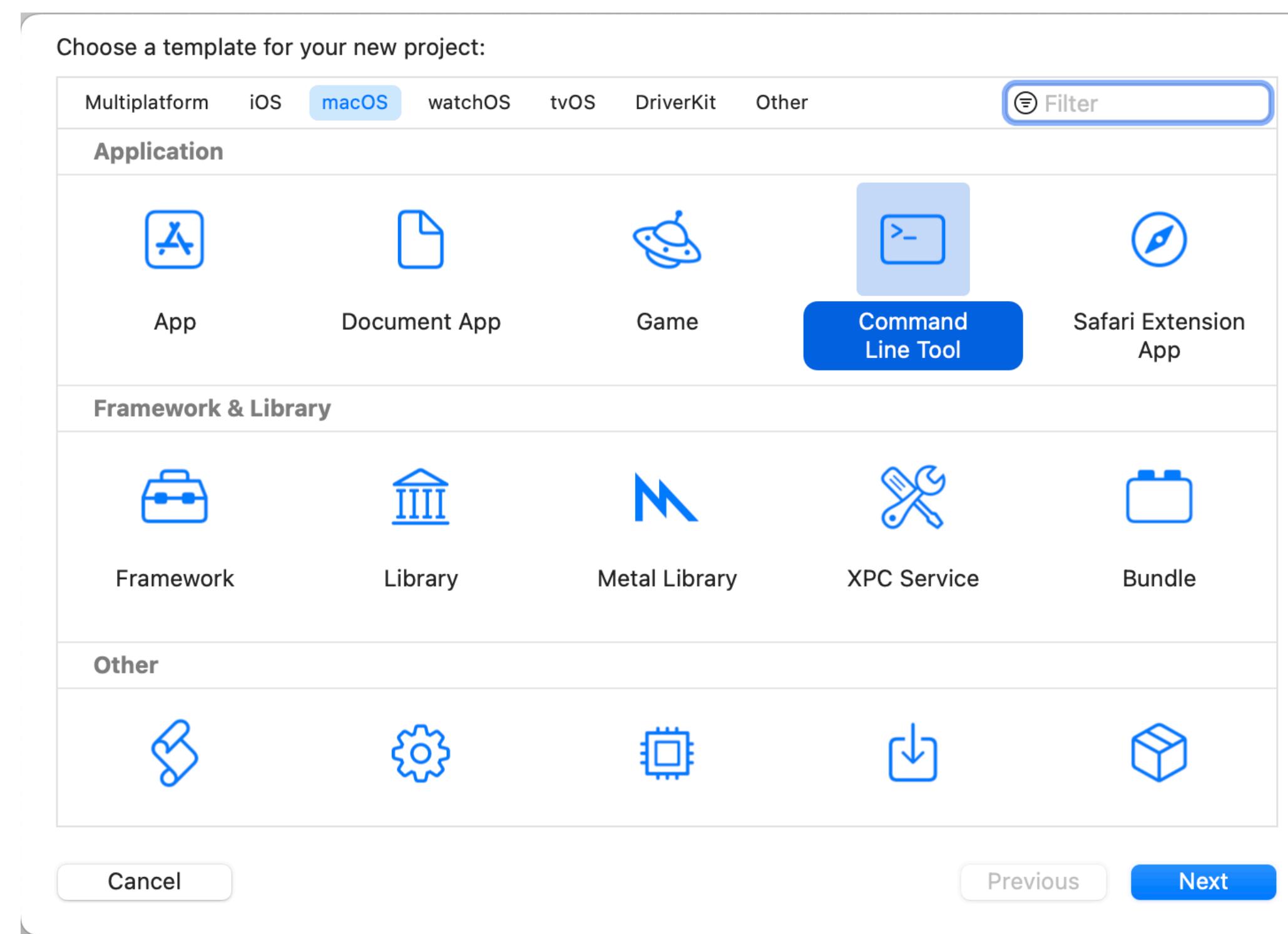
```
glfw — 1d ▾ make — 90x27
[ko@MacbookAir glfw % cmake -DBUILD_SHARED_LIBS=OFF .
-- The C compiler identification is AppleClang 13.1.6.13160021
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success
-- Found Threads: TRUE
-- Could NOT find Doxygen (missing: DOXYGEN_EXECUTABLE)
-- Including Cocoa support
-- Configuring done (29.9s)
-- Generating done (0.1s)
-- Build files have been written to: /Users/ko/Desktop glfw
[ko@MacbookAir glfw % make
[ 1%] Building C object src/CMakeFiles/glfw.dir/context.c.o
[ 2%] Building C object src/CMakeFiles/glfw.dir/init.c.o
[ 3%] Building C object src/CMakeFiles/glfw.dir/input.c.o
[ 4%] Building C object src/CMakeFiles/glfw.dir/monitor.c.o
[ 5%] Building C object src/CMakeFiles/glfw.dir/platform.c.o
[ 6%] Building C object src/CMakeFiles/glfw.dir/vulkan.c.o
[ 7%] Building C object src/CMakeFiles/glfw.dir/window.c.o
[ 8%] Building C object src/CMakeFiles/glfw.dir/egl_context.c.o
[ 9%] Building C object src/CMakeFiles/glfw.dir/osmesa_context.c.o
[10%] Building C object src/CMakeFiles/glfw.dir/null_init.c.o
```

정적 라이브러리 파일 (libglfw3.a) 생성



Xcode 프로젝트 설정

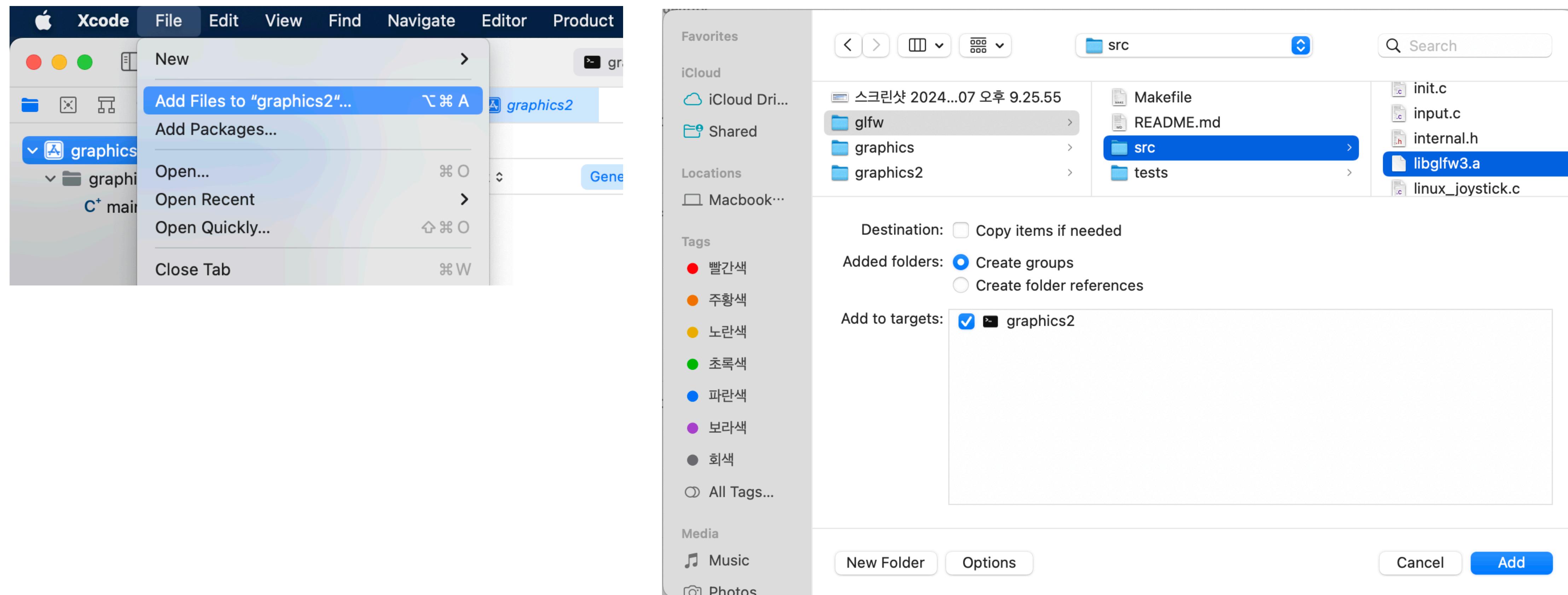
새로운 프로젝트 생성 후 libglfw3.a 파일 프로젝트에 추가



Xcode 프로젝트 설정

File > Add Files to <project> 선택

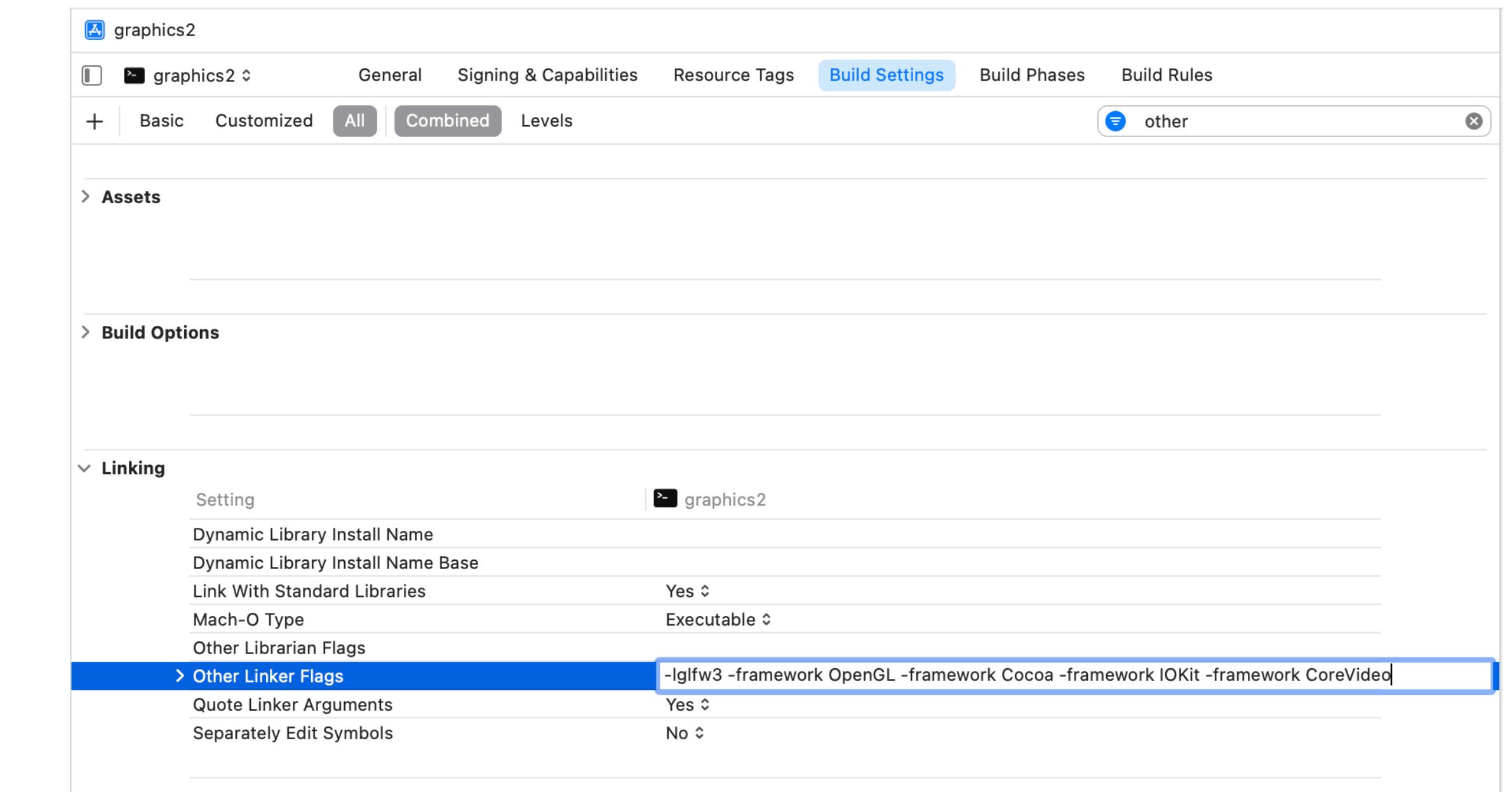
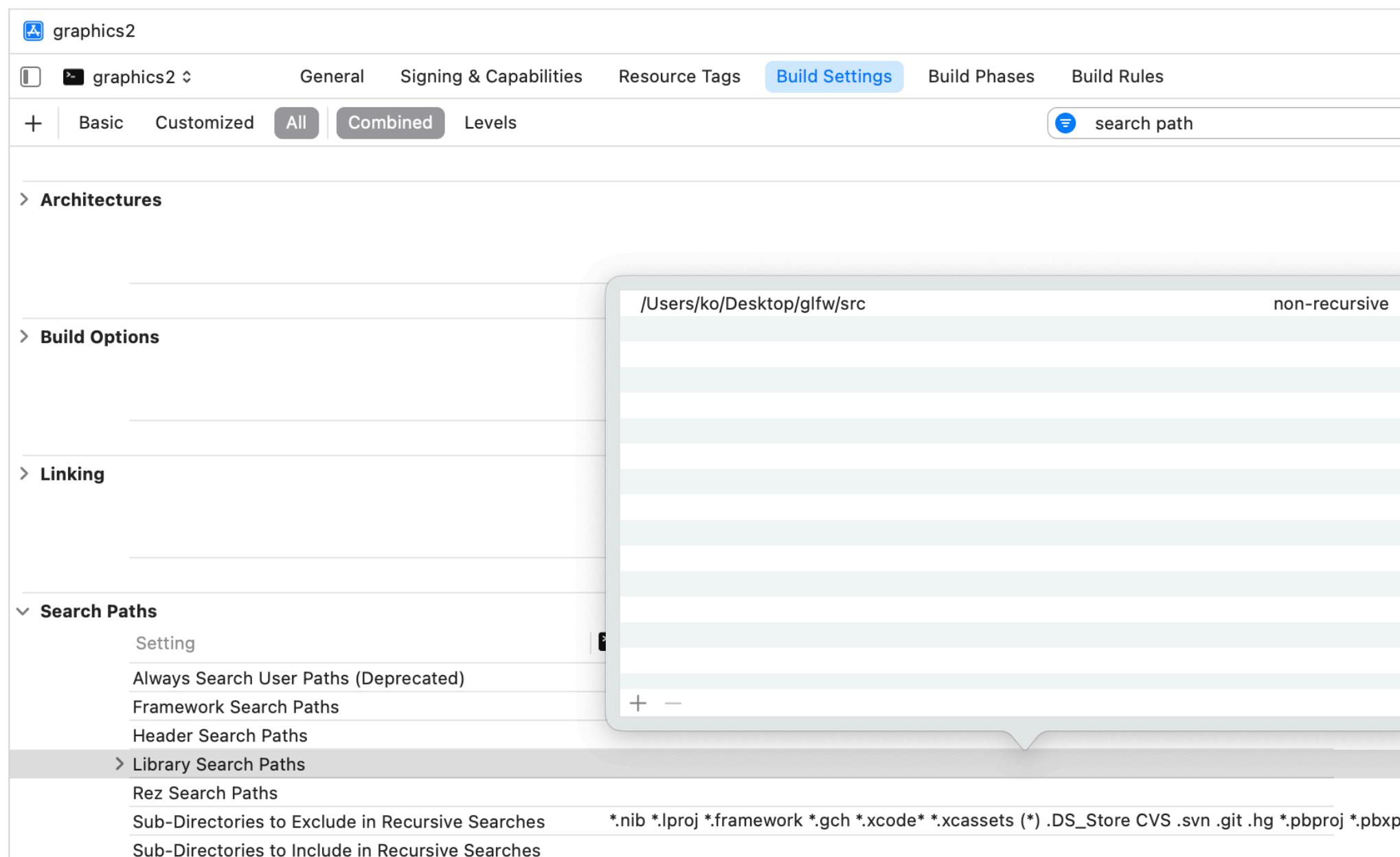
GLFW 라이브러리에서 생성된 libglfw3.a 파일 프로젝트에 추가



빌드 설정

Project Settings > Build Settings > Library Search Paths 설정에 GLFW 라이브러리가 위치한 디렉토리 추가

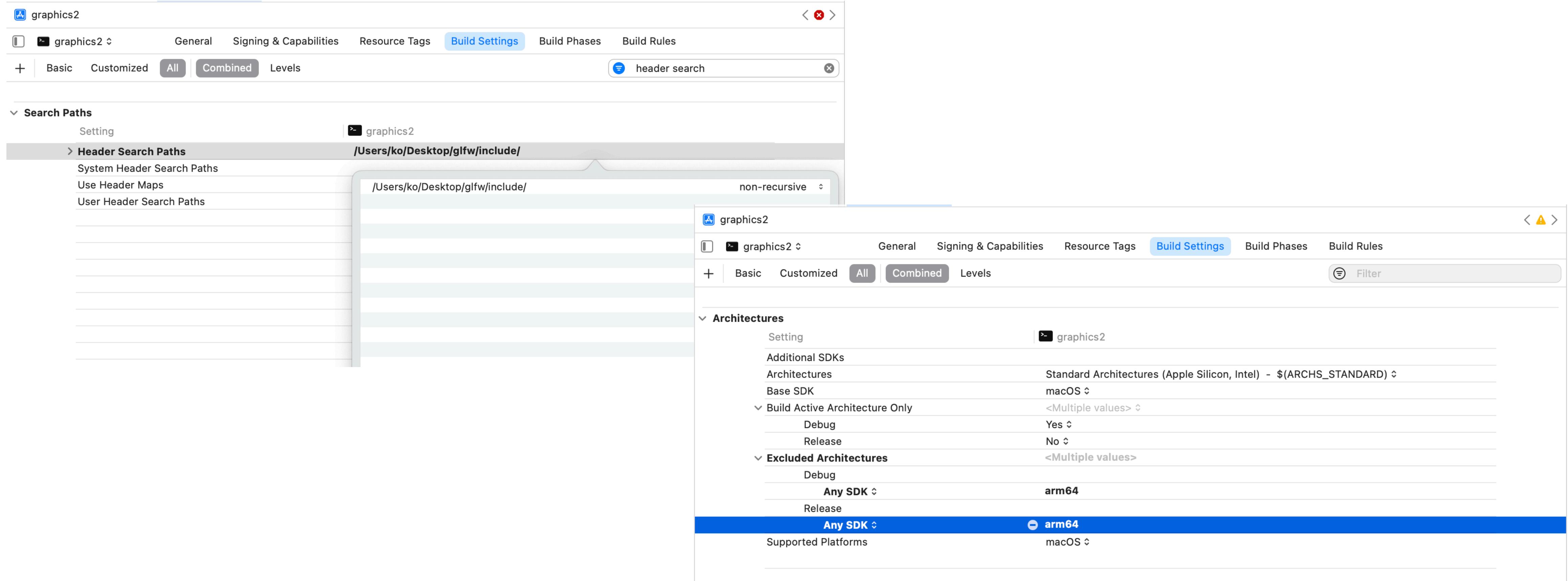
Other Linker Flags 설정에 -Iglfw3 -framework OpenGL -framework Cocoa -framework IOKit -framework CoreVideo 추가



빌드 설정

Search Paths > Header Search Paths에 include 디렉토리 추가

Architectures > Excluded Architectures > Debug,Release에 arm64 추가 (m1)



예제 코드 빌드

The screenshot shows a Mac OS X desktop environment with two windows open. On the left is the Xcode IDE, displaying a project named 'graphics2' in 'master' branch. The 'main.c' file is selected, showing the following C code:

```
1 #include <GLFW/glfw3.h>
2
3 int main(void)
4 {
5     GLFWwindow* window;
6
7     /* Initialize the library */
8     if (!glfwInit())
9         return -1;
10
11    /* Create a windowed mode window and its OpenGL context */
12    window = glfwCreateWindow(640, 480, "Hello World", NULL, NULL);
13    if (!window)
14    {
15        glfwTerminate();
16        return -1;
17    }
18
19    /* Make the window's context current */
20    glfwMakeContextCurrent(window);
21
22    /* Loop until the user closes the window */
23    while (!glfwWindowShouldClose(window))
24    {
25        /* Render here */
26        glClear(GL_COLOR_BUFFER_BIT); ⚠ 'glClear' is deprecated: first deprecated in macOS 10.14 - OpenGL API d...
27
28        /* Swap front and back buffers */
29        glfwSwapBuffers(window);
30
31        /* Poll for and process events */
32        glfwPollEvents();
33    }

```

The status bar at the bottom of the Xcode window shows:

2024-01-07 21:36:02.276725+0900 graphics2[5906:118933] Metal API Validation Enabled
2024-01-07 21:36:02.414728+0900 graphics2[5906:118933] Errors found! Invalidating cache...

To the right of the Xcode window is a terminal window titled 'Hello World' with a black background. It displays the command 'Hello World' and the output of the application's execution.

감사합니다