

Computer Graphics

유주한

컴퓨터AI학부

동아대학교

2025년 01학기



OpenGL

■ 역사?

- 알아서...

■ OpenGL 은 언어가 아닌 API 표준

- C나 C++ 와 같은 언어를 사용하여 OpenGL 라이브러리를 호출하는 형태
- Khronos Group에서 관리



라이브러리, API ???

■ 라이브러리(Library)

- 특정 기능을 포함하는 **코드 집합**(Compiled Code)
- OpenCV, SDL(멀티미디어), FreeType(폰트 렌더링)...

■ API

- 함수 및 명령어들의 "정의(Definition)"를 제공하는 인터페이스 규약
- 실제 코드 구현이 포함되지 않으며, 이를 구현하는 하드웨어 또는 소프트웨어가 따로 필요
 - ✓ 그래픽처리 예) 실제 구현은 **GPU 드라이버**가 담당

➔ API에서 라이브러리를 호출해서 실행



OpenGL

■ OpenGL이 OS나 GPU 제조사(NVIDIA, AMD, Intel 등)의 서로 다른 드라이버에서도 어떻게 호환성을 유지?

- 모든 제조사는 OpenGL 표준을 기반으로 드라이버를 구현
- 운영체제에서 여러 GPU 드라이버를 지원하는 표준 준수

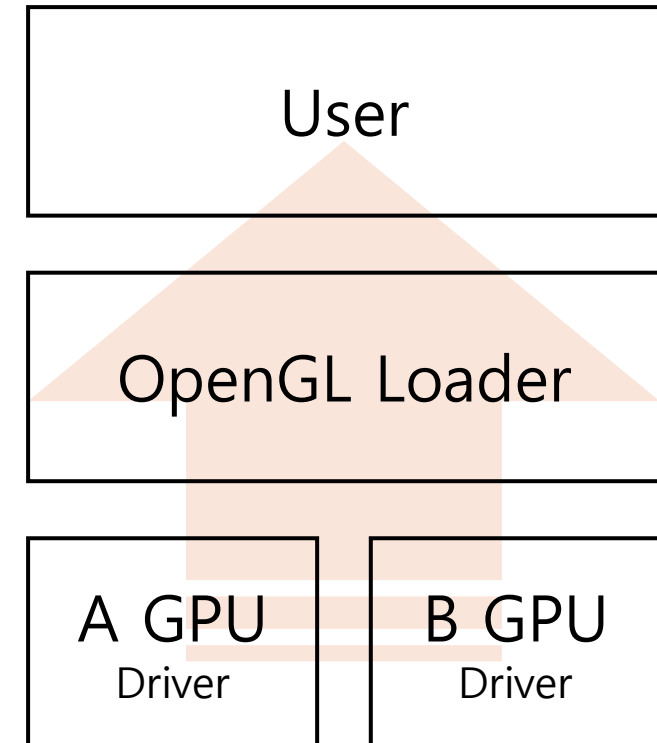
- GLEW, GLAD 등의 로더(Loader)를 통해 드라이버에 맞게 동적 연결 ??? (다음장)
- 제조사별 독자적인 기능을 OpenGL 확장으로 제공

- ➔ OpenGL 명령어(glDrawArrays, glBindBuffer 등)는 표준 API로 정의
- ➔ 각 GPU 제조사는 이를 자사 하드웨어에 맞게 최적화하여 구현
- ➔ OpenGL API는 일정하지만, 내부적으로는 GPU마다 다르게 최적화



“GLEW, GLAD 등의 로더를 통해 드라이버에 맞게 동적 연결 ???”

- OpenGL은 OS 또는 GPU 드라이버에 Static이 아닌 Dynamic하게 연결
→ GLEW, GLAD 같은 **로더(Loader)**가 GPU 드라이버에서 OpenGL 함수를 동적으로 연결
- **OpenGL Loader 역할**
 - 실행 중에 현재 시스템의 GPU 드라이버가 지원하는 OpenGL 기능을 탐색
 - 제조사별 GPU 드라이버에서 해당 OpenGL 함수를 로드하여 호출 가능하게 함
- OpenGL 로더를 사용하면 다양한 GPU 드라이버에서도 동일한 OpenGL API를 호출 가능





OpenGL Loader

- OpenGL API 함수를 실행할 수 있도록 도와주는 라이브러리
- 대표적인 OpenGL Loader 라이브러리

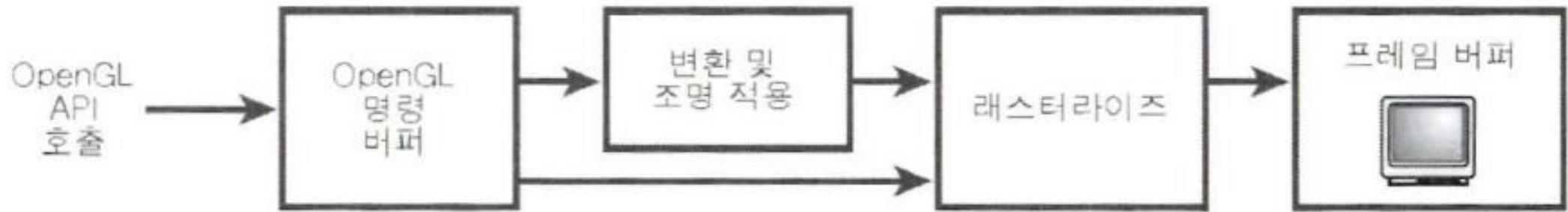
| Loader | 설명 |
|---|-------------------------------|
| GLEW (OpenGL Extension Wrangler Library) | 가장 많이 사용되는 OpenGL Loader 중 하나 |
| GLAD (OpenGL Loader Generator) | 최신 OpenGL 지원 및 크로스플랫폼 기능 강화 |
| GL3W (OpenGL 3+ Loader) | OpenGL 3.0 이상을 위한 경량 Loader |

OpenGL

■ 파이프라인

- 두 개 이상의 분리된 상태 또는 단계가 이어지는 처리 과정

■ 간략한 OpenGL 파이 프라인



- 변환 및 조명 적용 : 물체의 모양을 나타내는 점들이 물체의 위치와 방향에 따라 다시 계산되고, 각 vertex에 입혀질 색상이 적용되는 단계
- 래스터라이즈(rasterization) : 주어진 지오메트리의 외관, 색상, 텍스처 데이터를 조합하여 실제 화면 (픽셀 그리드)으로 변환하는 과정, 프레임버퍼(영상 메모리)에 실제 색상 픽셀을 채워 넣는다.
- 프레임 버퍼 : **디스플레이**(모니터)에 출력될 **픽셀 데이터**(색상, 깊이, 스텐실 등)를 저장하는 **메모리 공간**



glut

- 초기 OpenGL 학습환경에서 운영체제마다 다른 **윈도우·이벤트 처리 코드**를 간단히 통일해주기 위해 만들어짐
- glutInit(), glutCreateWindow(), glutMainLoop() 등을 통해 창 생성
- 콜백 함수(glutDisplayFunc(), glutKeyboardFunc(), glutMouseFunc())로 렌더링·입력 이벤트 처리
- 플랫폼 독립적인 방식으로 OpenGL 애플리케이션을 빠르게 개발 가능
- 원조 GLUT(스톨맨 저작권 관련)는 업데이트가 멈췄고, **FreeGLUT**나 **OpenGLUT** 같은 포크가 주로 사용
 - Freeglut : #include <GL/glut.h>를 하면 <GL/gl.h> 등이 자동으로 포함
- 상용/고급 애플리케이션에서는 **GLFW**, **SDL** 등이 대체재로 많이 쓰임



OpenGL 설치



freeglut



전체

이미지

동영상

쇼핑

뉴스

짧은 동영상

웹

: 더보기



FreeGLUT

<https://freeglut.sourceforge.net>



The freeglut Project :: About

GLUT (and hence **freeglut**) takes care of all the system-specific chores required for creating windows, initializing OpenGL contexts, and handling input events, ...

Freeglut download

Download Leader badge A free OpenGL utility toolkit, the open ...

Installation Instructions

Build and install freeglut. This is as simple as: make sudo make ...

[sourceforge.net](#) 검색결과 더보기 »



OpenGL 설치

MSVC(Microsoft Visual C++)

: c++ 컴파일러, Visual Studio 내에서 쓰임

The [freelut source code repository](#) is now hosted on github. So you can also send us pull requests or report bugs through. Large contributions are best suited to the mailing list.

Downloads...

Below are file links for the freelut project. README files are included. Have fun!

Stable Releases

Stable releases are available on our [releases page on github](#), where you can find more information about each release.

- [freelut 3.6.0 \(sourceforge mirror\)](#) [Released: 12 June 2024]
- [freelut 3.4.0 \(sourceforge mirror\)](#) [Released: 7 October 2022]
- [freelut 3.2.2 \(sourceforge mirror\)](#) [Released: 6 February 2022]
- [freelut 3.2.1 \(sourceforge mirror\)](#) [Released: 29 September 2019]
- [freelut 3.2.0 \(sourceforge mirror\)](#) [Released: 16 September 2019]
- [freelut 3.0.0 \(sourceforge mirror\)](#) [Released: 7 March 2015]
- [freelut 2.8.1 \(sourceforge mirror\)](#) [Released: 5 April 2013]
- [freelut 2.8.0 \(sourceforge mirror\)](#) [Released: 2 January 2012]
- [freelut 2.6.0 \(sourceforge mirror\)](#) [Released: 27 November 2009]
- [freelut 2.4.0 \(sourceforge mirror\)](#) [Released: 9 June 2005]
- [freelut 2.2.0 \(sourceforge mirror\)](#) [Released: 12 December 2003]
- [freelut 2.0.1 \(sourceforge mirror\)](#) [Released: 23 October 2003]

Pre-Compiled Packages

The freelut project does not distribute pre-compiled or otherwise system-specific packages for freelut. We only release distributions include some version of freelut, which might provide a more convenient way to obtain and use freelut.

- GNU/Linux distributions such as Debian, Arch, Ubuntu, RedHat and so on, all distribute some version of freelut. !
freelut3-dev.
- FreeBSD includes freelut in its [ports system](#). as does NetBSD.
- Windows users using the msys2/mingw system can install freelut with the command: `pacman -S mingw-w64-x86_64-freelut`
- MacOS X users can use the [homebrew](#) package manager to install freelut with the command: `brew install freelut`
- Another option for windows users is [Martin Payne's Windows binaries \(MSVC and MinGW\)](#)

If you have problems with these packages, please contact their maintainers, we probably can't help.



OpenGL 설치

- windows
- visual studio 2019

freelut-MSVC-3.0.0-2.mp.zip



bin



include



lib



Copying.txt



Readme.txt

<https://www.transmissionzero.co.uk/software/freelut-devel/>

Transmission Zero

freelut Windows Development Libraries

[Home](#)[Radio](#)[Audio](#)[Computing](#)[Software](#)[Gaming](#)[Food & Drink](#)[Contact](#)[News](#)[Facebook](#)

Introduction

Whilst at the University of Essex, I took a module called "Interactive Computer Graphics" (or EE222 as we referred to it). Half of the course consisted of using POV-Ray to create images, and then putting them together to make a high quality animation. The other half of the course consisted of programming real-time interactive graphics using C and OpenGL, with the help the GLUT library [freelut](#). I went on to do my final year project, creating a simulation in C++ and OpenGL. It was the first time I had ever written a real application, and I still have a soft spot for GLUT.

For my university project, I used Nate Robins' GLUT for Win32 project for the Windows build, and freelut for the Linux build. Windows freelut binary packages are somewhat hard to find, particularly for the MinGW compiler, so I've created suitable packages and put them online. You can use them to build freelut applications using Microsoft's Visual C++ and MinGW.

Get the Sample Code

If you're downloading one of the packages on this page, you'll probably want some sample code to try it out. If you have Git installed, you can get the sample code for a simple "Hello World" GLUT application by running `git clone https://github.com/TransmissionZero/Hello-GLUT.git`. Alternatively you can [download a Hello GLUT source release](#). You are also encouraged to visit that link to star and watch the repository if you find it useful.

freelut 3.0.0 MSVC Package

This package contains 32 and 64 bit Windows DLLs, import libraries, and header files, allowing freelut applications to be written using Microsoft's Visual C++. The package was built from source code using Visual Studio 2013. The DLL is binary compatible with both my MinGW DLL, and the GLUT for Win32 DLL provided by Nate Robins. The DLL has been tested on Windows 98 SE, Windows ME, Windows 2000, Windows XP, Windows Vista, Windows 7 (64 bit), and Windows 8 (64 bit).

[Download freelut 3.0.0 for MSVC](#) (with [PGP signature](#) and [PGP key](#))

freelut 3.0.0 MinGW Package

This package contains 32 and 64 bit Windows DLLs, import libraries, static libraries, and header files, allowing freelut applications to be compiled using the MinGW compiler. The package was built from source code using MinGW. The DLL is binary compatible with both my MSVC DLL, and the GLUT for Win32 DLL provided by Nate Robins. The DLL has been tested on Windows 98 SE, Windows ME, Windows 2000, Windows XP, Windows Vista, Windows 7 (64 bit), and Windows 8 (64 bit).

The package should also work with the various IDEs which offer frontends to MinGW / gcc, for example [Eclipse](#), [Dev-C++](#), and [Code::Blocks](#). However, you'll need to consult the manual for your IDE for instructions on how to do this.

[Download freelut 3.0.0 for MinGW](#) (with [PGP signature](#) and [PGP key](#))

For further details on using freelut with MinGW, see my [GLUT MinGW article](#).

Older freelut Packages



If you would prefer to download the older 2.4.0, 2.6.0 or 2.8.x packages, they can be found in the [older GLUT packages directory](#). If you need to support Windows NT 4.0, you will need to download one of these packages as freelut ≥2.8.0 is not compatible with this platform.




Visual Studio 2019

최근 파일 열기(R)



▲ 오늘

-  **Project1.sln** 25-02-11(화) 02:15
C:\Wclass\W2025_01_DAW4th_computer_graphics\WProject1
-  **Project1.sln** 25-02-11(화) 02:13
C:\Wclass\W2025_01_DAW4th_computer_graphics\WHello-GLUT-rel-v1.0\WProject1

▲ 이번 달

-  **Thetis_VS2022.sln** 25-01-16(목) 10:41
C:\WUsers\Wyoow\Downloads\WThetis-2.10.3.5\WProject Files\WSource

▲ 오래됨

-  **Project1.sln** 24-05-08(수) 23:17
C:\Wresearches\Wmammogram\WdicomToTiff
-  **Project1.sln** 24-05-08(수) 23:10
C:\Wresearches\Wmammogram\WdicomTopng

시작



리포지토리 복제(C)

GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기



프로젝트 또는 솔루션 열기(P)

로컬 Visual Studio 프로젝트 또는.sln 파일 열기



로컬 폴더 열기(F)

폴더 내에서 탐색 및 코드 편집



새 프로젝트 만들기(N)

시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

[코드를 사용하지 않고 계속\(W\) →](#)



새 프로젝트 만들기

최근 프로젝트 템플릿(R)

빈 프로젝트

C++



템플릿 검색(Alt+S)(S)



모든 언어(L)

모든 플랫폼(P)

모든 프로젝트 형식(T)



ASP.NET 웹 애플리케이션(.NET Framework)

ASP.NET 애플리케이션을 만드는 데 사용되는 프로젝트 템플릿입니다. ASP.NET Web Forms, MVC 또는 Web API 애플리케이션을 만들 수 있으며 ASP.NET의 기타 여러 기능을 추가할 수 있습니다.

Visual Basic

Windows

클라우드

웹



빈 프로젝트

Windows용 C++를 사용하여 처음부터 시작합니다. 시작 파일을 제공하지 않습니다.

C++

Windows

콘솔



콘솔 앱

Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.

C++

Windows

콘솔



CMake 프로젝트

.sln 또는 .vcxproj 파일에 종속되지 않은 최신 플랫폼 간 C++ 앱을 빌드하세요.

C++

Windows

Linux

콘솔



Windows 데스크톱 마법사

마법사를 사용하여 고유한 Windows 앱을 만드세요.

뒤로(B)

다음(N)



새 프로젝트 구성

빈 프로젝트

C++

Windows

콘솔

프로젝트 이름(I)

Project6

위치(L)

C:\Users\Wyoos\source\repos

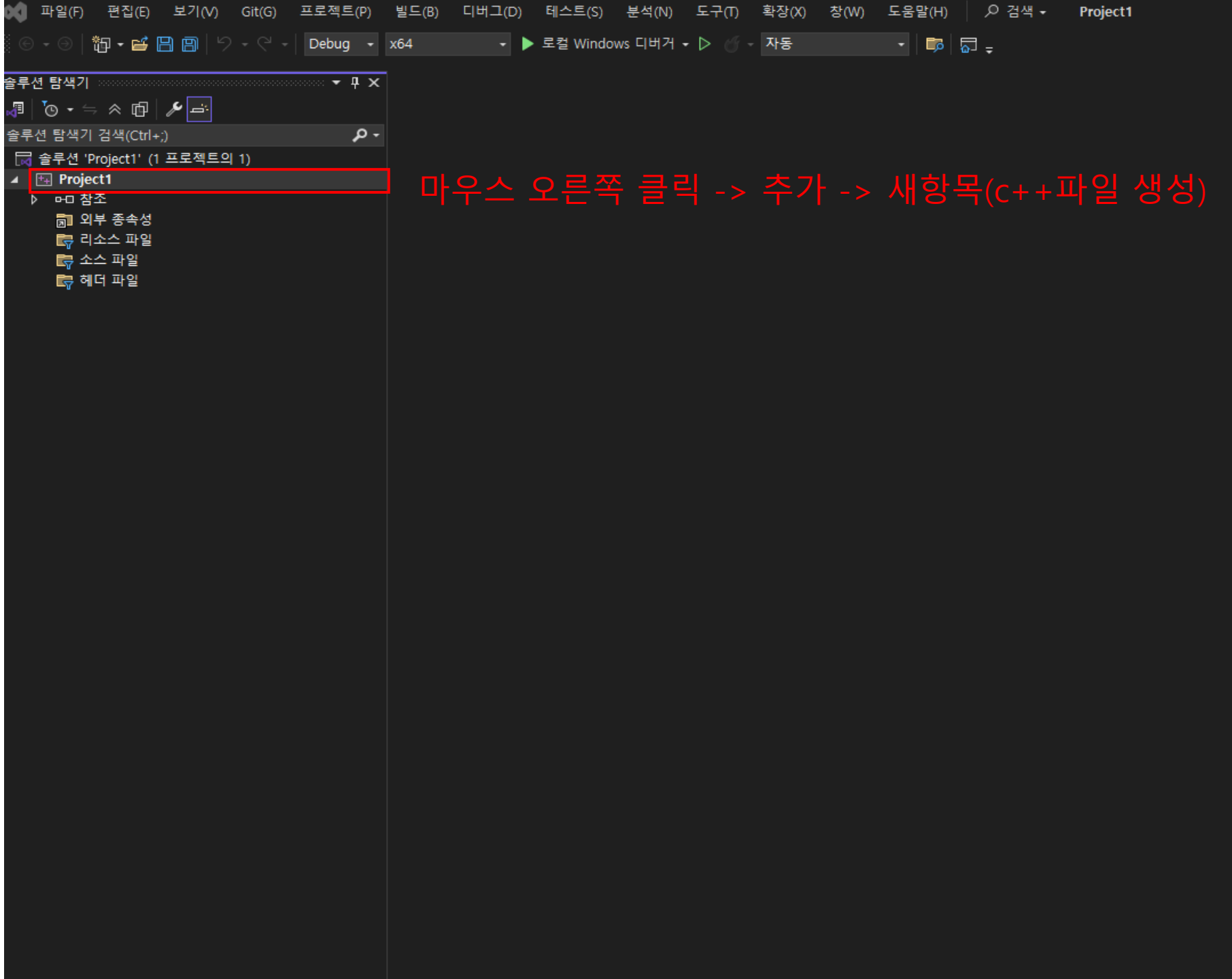
솔루션 이름(M) ⓘ

Project6

☒ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

뒤로(B)

만들기(C)





솔루션 탐색기

솔루션 탐색기 검색(Ctrl+)

솔루션 'Project1' (1 프로젝트의 1)

Project1

참조

외부 종속성

리소스 파일

소스 파일

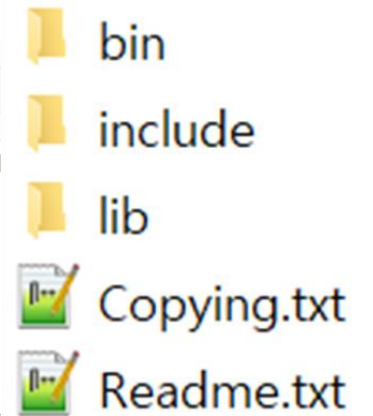
헤더 파일

마우스 오른쪽 클릭 -> 속성



OpenGL 설치

freeglut-MSVC-3.0.0-2.mp.zip



freeglut 압축파일 폰 폴더에서
include 폴더까지의 전체경로

C:\classW2025_01_DAW4th_computer_graphics\freeglut\include;%(\AdditionalInclude

C:\classW2025_01_DAW4th_computer_graphics\freeglut\lib*.*;%(\AdditionalLibraryDirectories)

freeglut.lib;%(\AdditionalDependencies)

Project1 속성 페이지

구성(C): 모든 구성

플랫폼(P): 활성(x64)

구성 속성

일반
고급
디버깅
VC++ 디렉터리

C/C++

일반

추가 포함 디렉터리
추가 #using 디렉터리
추가 BMI 디렉터리
추가 모듈 종속성
추가 헤더 단위 종속성
소스의 모듈 종속성 검사
include 지시문을 import 지시문으로 변환

아니요

아니요

C/C++

일반
최적화
전처리기
코드 생성
언어
미리 컴파일된 헤더
출력 파일
찾아보기 정보
외부 include 지시문
고급
모든 옵션
명령줄

시작 배너 표시 안 함
가져오기 라이브러리 무시
출력 등록
사용자 단위 리디렉션
추가 라이브러리 디렉터리
라이브러리 종속성 링크
라이브러리 종속성 입력 사용
링크 상태
DLL 바인딩 방지
링커 경고를 오류로 처리
강제 파일 출력
핫 패치 가능한 이미지 만들기
섹션 특성 지정

예(/NOLOGO)

아니요

아니요

아니요

예

아니요

구성 속성

일반
고급
디버깅
VC++ 디렉터리
C/C++
일반
최적화
전처리기
코드 생성
언어
미리 컴파일된 헤더
출력 파일
찾아보기 정보
외부 include 지시문
고급
모든 옵션
명령줄

링커

일반

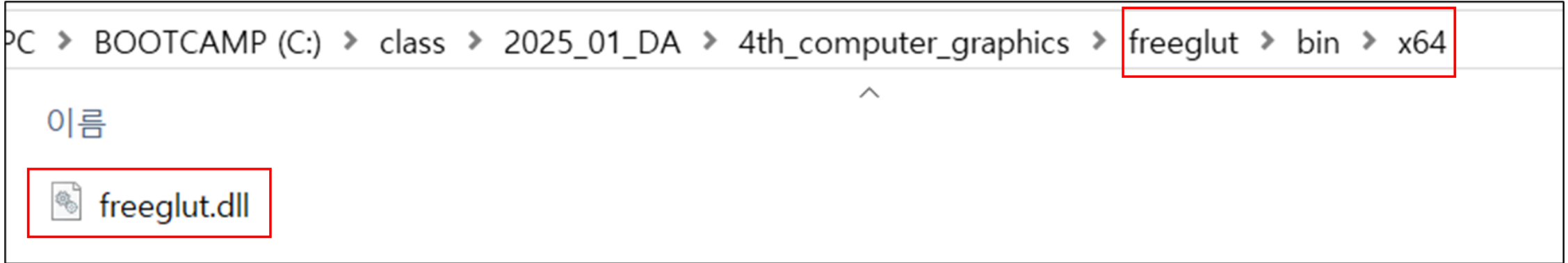
입력

추가 종속성
모든 기본 라이브러리 무시
특정 기본 라이브러리 무시
모듈 정의 파일
어셈블리에 모듈 추가
관리되는 리소스 파일 포함
강제 기호 참조
지연 로드된 DLL
어셈블리와 리소스 링크



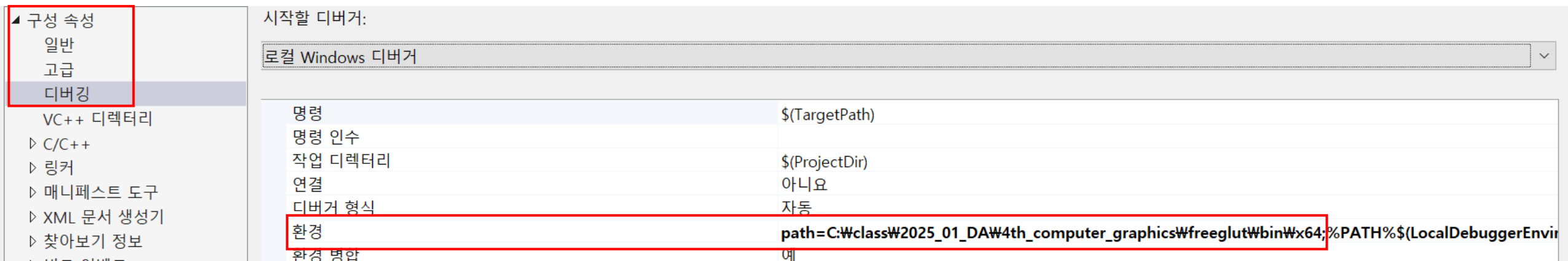
OpenGL 설치

■ .dll을 해당 프로젝트 폴더에 복사



■ 혹은, 프로젝트 속성 > 디버깅 > 환경에

- path=dll경로;%PATH%
- 예) path=C:\class\2025_01_DA\4th_computer_graphics\freeglut\bin\x64;%PATH%





데이터 타입 및 함수 이름

■ 데이터 타입

| OpenGL 데이터 타입 | 설명 | 대응되는 C 데이터 타입 | 접두어 |
|------------------------------------|---------------------|----------------|-----|
| GLbyte | 8비트 정수 | signed char | b |
| GLshort | 16비트 정수 | short | s |
| GLint, GLsizei | 32비트 정수 | long | l |
| GLfloat, GLclampf | 32비트 실수 | float | f |
| GLdouble, GLclampd | 64비트 실수 | double | d |
| GLubyte, GLboolean | 8비트 부호없는 정수 | unsigned char | ub |
| GLushort | 16비트 부호없는 정수 | unsigned short | us |
| GLunit, <u>GLenum</u> , GLbitfield | <u>32비트 부호없는 정수</u> | unsigned long | ui |

■ 함수 이름

