Programming Assignment 1

2026 Computer Graphics

Computer Graphics Laboratory

2026/03/23

Seongil Kim

Contents

- Prerequisites
    - Setup the assignment
    - Submission
    - How to comment commit id
    - Policies
- Assignment Overview
    - Introduction to OpenGL Core Profile
    - Introduction to Shaders
    - Tasks
    - Tips for tasks
    - Additional resources
    - Demonstration

Prerequisites

Setup the assignment

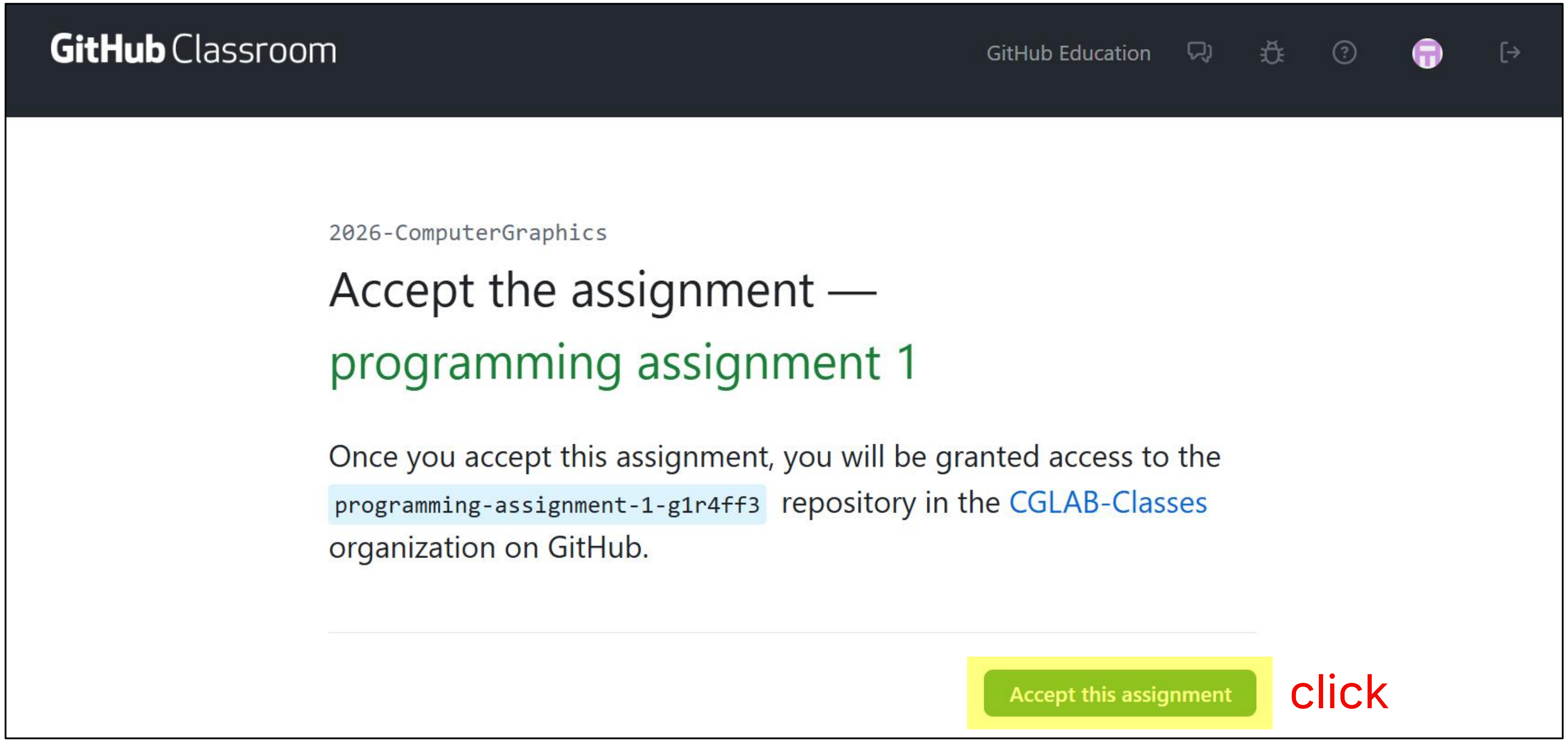
- Minimum requirements
  - C++ compiler
    - MSVC 19.30+ (VS2022+)
    - GCC 11+
    - Apple Clang 14+
  - CMake 3.22+
  - Git

- For Windows users, Visual Studio 2019 is not supported because it's not supported by Microsoft as of 2026. Use Visual Studio 2026 instead.

- For your convenience, there is a setup script for this assignment
  - Windows: setup.ps1 (powershell)
  - Linux/macOS: setup.sh (bash)

Setup the assignment

1. Take the assignment
   - Login github
   - Follow https://classroom.github.com/a/mfIgAFDE
   - Accept the assignment



2026-ComputerGraphics

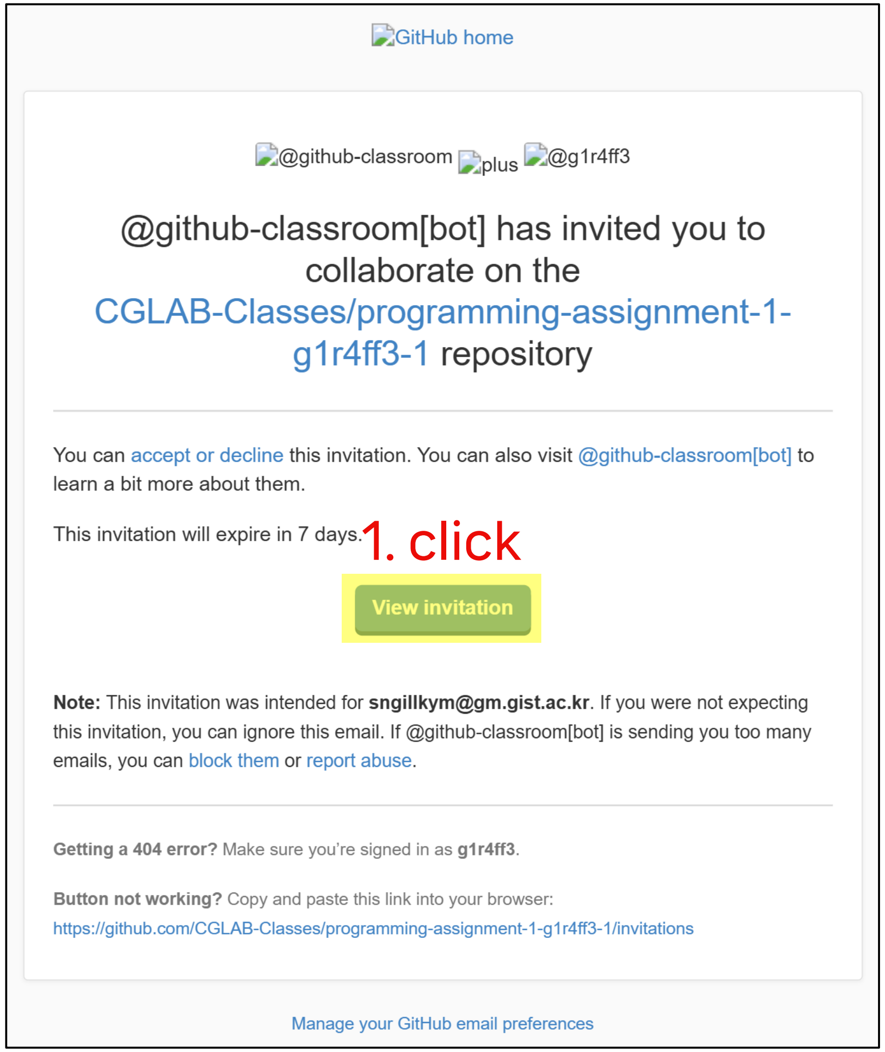
Accept the assignment — programming assignment 1

Once you accept this assignment, you will be granted access to the `programming-assignment-1-g1r4ff3` repository in the CGLAB-Classes organization on GitHub.

click

Setup the assignment

2. Accept the invitation from github-classroom sent to your email



GitHub home

@github-classroom plus @g1r4ff3

@github-classroom[bot] has invited you to collaborate on the CGLAB-Classes/programming-assignment-1-g1r4ff3-1 repository

You can accept or decline this invitation. You can also visit @github-classroom[bot] to learn a bit more about them.

This invitation will expire in 7 days.

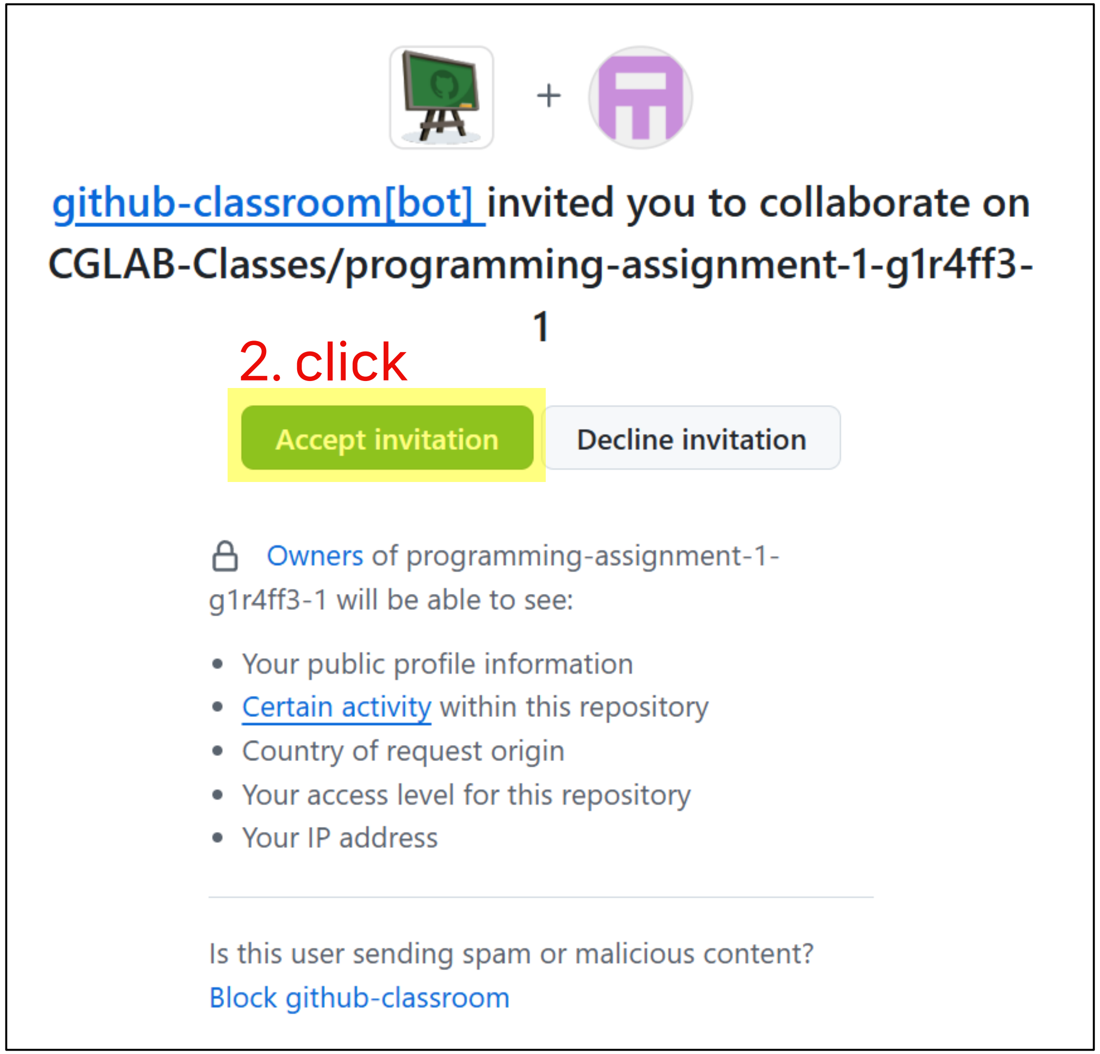
1. click

Note: This invitation was intended for **sngillkym@gm.gist.ac.kr**. If you were not expecting this invitation, you can ignore this email. If @github-classroom[bot] is sending you too many emails, you can block them or report abuse.

Getting a 404 error? Make sure you're signed in as **g1r4ff3**.

Button not working? Copy and paste this link into your browser:
https://github.com/CGLAB-Classes/programming-assignment-1-g1r4ff3-1/invitations

Manage your GitHub email preferences



github-classroom[bot] invited you to collaborate on CGLAB-Classes/programming-assignment-1-g1r4ff3-1

2. click

Owners of programming-assignment-1-g1r4ff3-1 will be able to see:

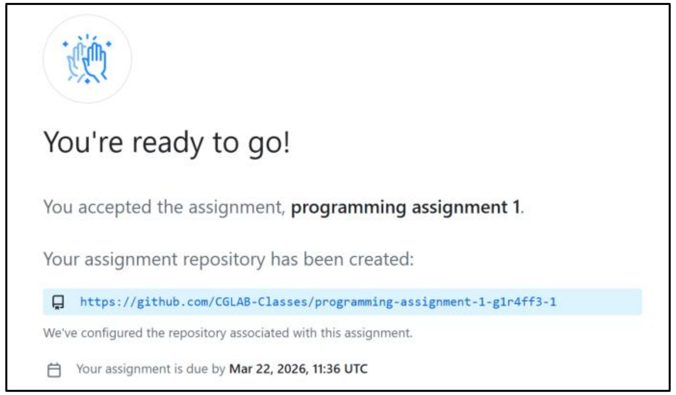
- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?

Block github-classroom

Setup the assignment

3. After the invitation is accepted, you can access your assignment repository in CGLAB-Classes organization



You're ready to go!

You accepted the assignment, **programming assignment 1**.

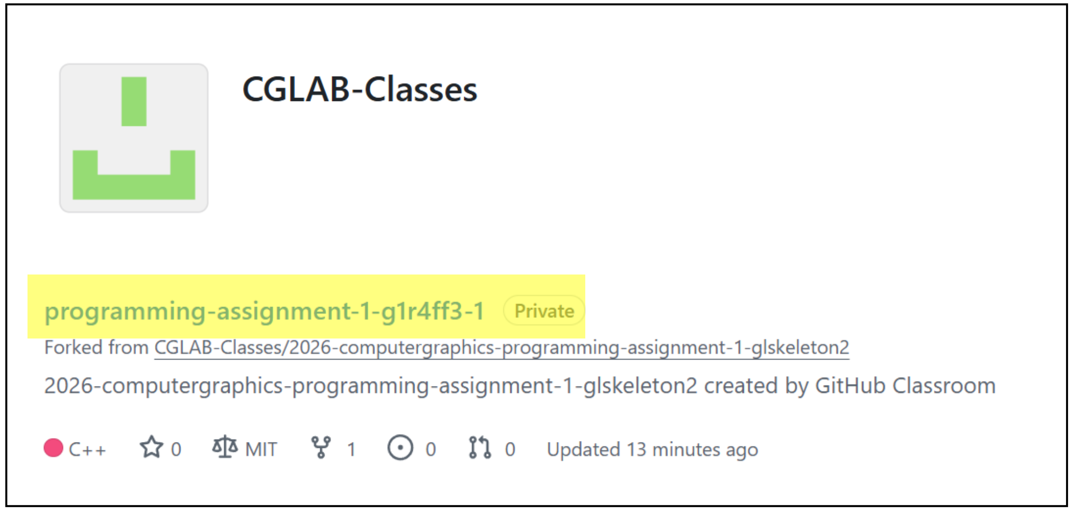
Your assignment repository has been created:

https://github.com/CGLAB-Classes/programming-assignment-1-g1r4ff3-1

We've configured the repository associated with this assignment.

Your assignment is due by **Mar 22, 2026, 11:36 UTC**

4. Go to your assignment repository



CGLAB-Classes

programming-assignment-1-g1r4ff3-1 Private

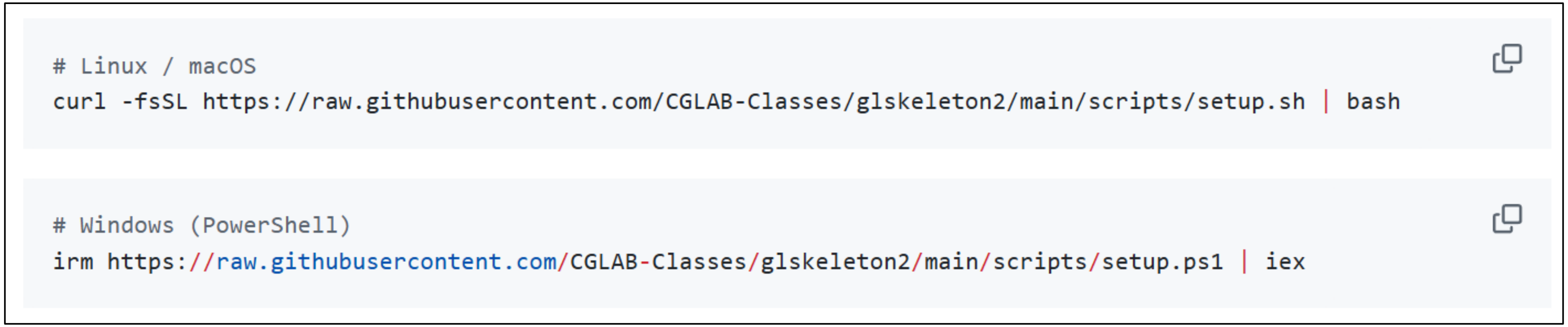
Forked from CGLAB-Classes/2026-computergraphics-programming-assignment-1-glskeleton2

2026-computergraphics-programming-assignment-1-glskeleton2 created by GitHub Classroom

● C++ ☆ 0 MIT 1 0 0 Updated 13 minutes ago

Setup the assignment

5. According to your OS, download and run the setup script in terminal by following commands



```
# Linux / macOS
curl -fsSL https://raw.githubusercontent.com/CGLAB-Classes/glskeleton2/main/scripts/setup.sh | bash
```

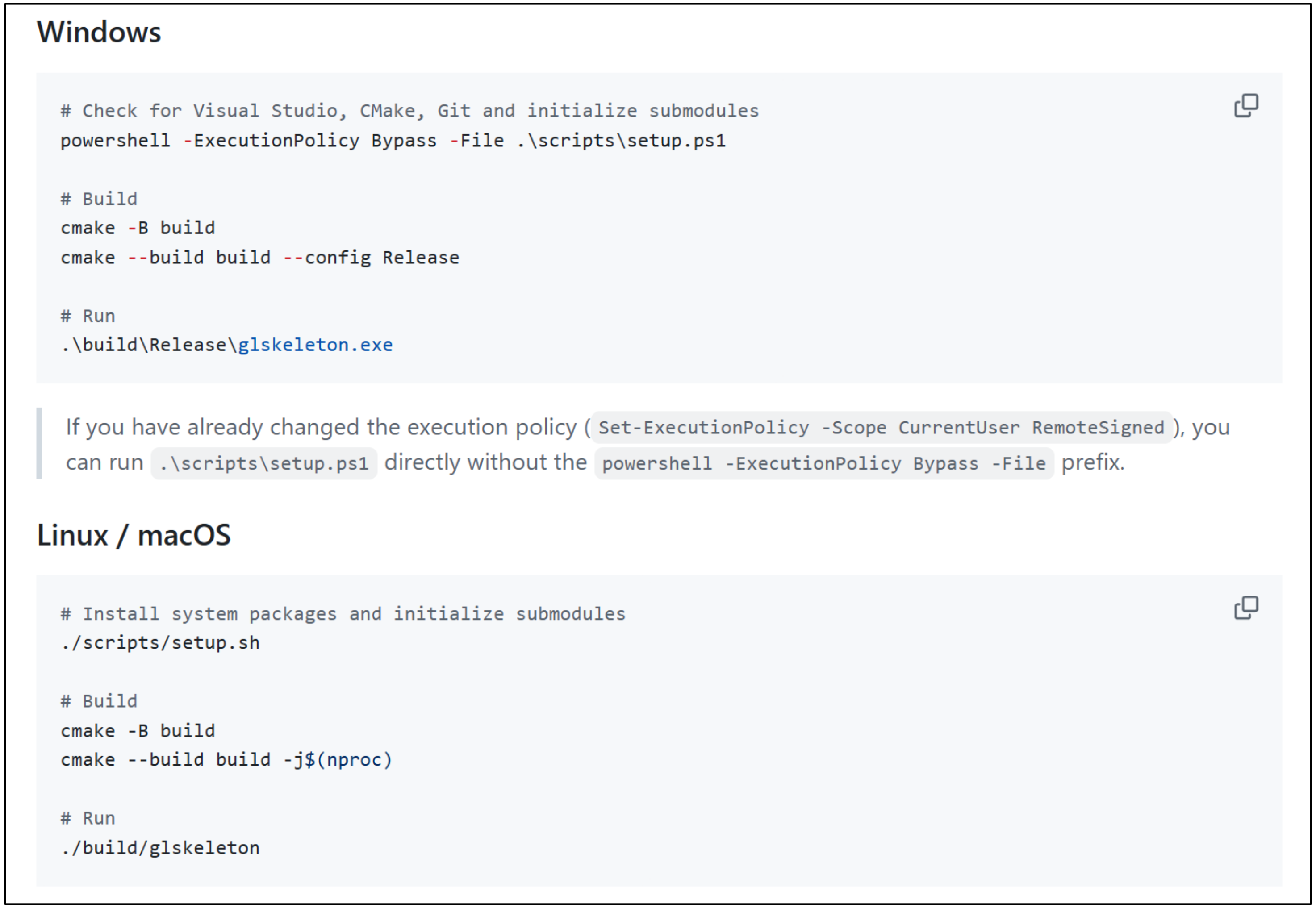
```
# Windows (PowerShell)
irm https://raw.githubusercontent.com/CGLAB-Classes/glskeleton2/main/scripts/setup.ps1 | iex
```

- The script will check your system whether it meets the requirements
- If one of them don't exist, the script will install default location to your system

6. Clone your assignment repository after setup
   - git clone --recursive <your pa link>

Setup the assignment

7. According to your OS, follow instructions below to build and run



Windows

```
# Check for Visual Studio, CMake, Git and initialize submodules
powershell -ExecutionPolicy Bypass -File .\scripts\setup.ps1

# Build
cmake -B build
cmake --build build --config Release

# Run
.\build\Release\glskeleton.exe
```

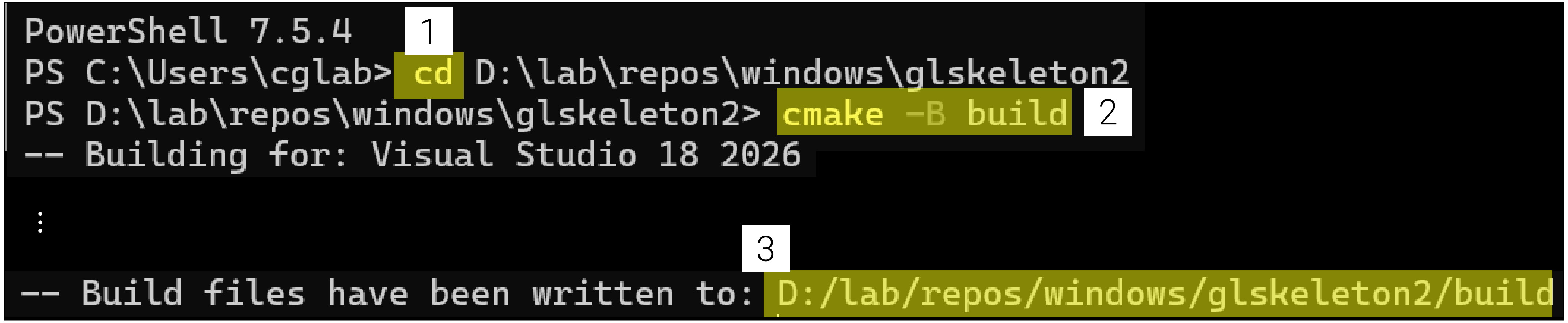
> If you have already changed the execution policy (`Set-ExecutionPolicy -Scope CurrentUser RemoteSigned`), you can run `.\scripts\setup.ps1` directly without the `powershell -ExecutionPolicy Bypass -File` prefix.

Linux / macOS

```
# Install system packages and initialize submodules
./scripts/setup.sh

# Build
cmake -B build
cmake --build build -j$(nproc)

# Run
./build/glskeleton
```

Setup the assignment

- How to build and run for Windows/Visual Studio users in GUI after the setup is complete

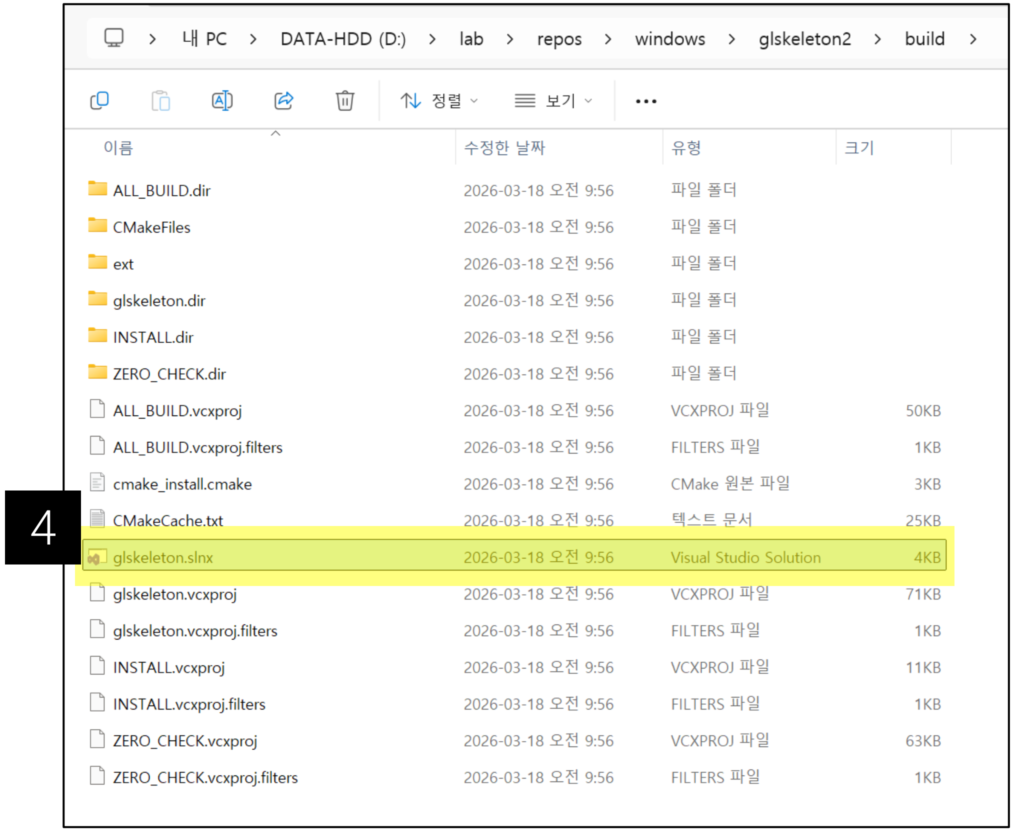


```
PowerShell 7.5.4
PS C:\Users\cglab> cd D:\lab\repos\windows\glskeleton2
PS D:\lab\repos\windows\glskeleton2> cmake -B build
-- Building for: Visual Studio 18 2026

 ⋮

-- Build files have been written to: D:/lab/repos/windows/glskeleton2/build
```

1. Go to your local repository in terminal: cd <root_path>
2. Make a build system: cmake –B build
3. Find and open the build directory

Setup the assignment

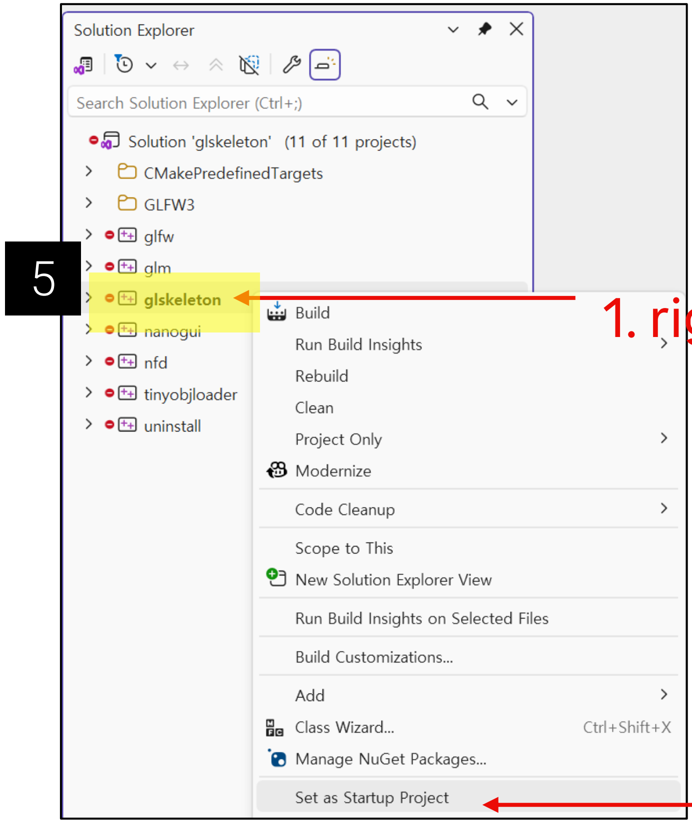
4



내 PC > DATA-HDD (D:) > lab > repos > windows > glskeleton2 > build >

이름	수정한 날짜	유형	크기
ALL_BUILD.dir	2026-03-18 오전 9:56	파일 폴더	
CMakeFiles	2026-03-18 오전 9:56	파일 폴더	
ext	2026-03-18 오전 9:56	파일 폴더	
glskeleton.dir	2026-03-18 오전 9:56	파일 폴더	
INSTALL.dir	2026-03-18 오전 9:56	파일 폴더	
ZERO_CHECK.dir	2026-03-18 오전 9:56	파일 폴더	
ALL_BUILD.vcxproj	2026-03-18 오전 9:56	VCXPROJ 파일	50KB
ALL_BUILD.vcxproj.filters	2026-03-18 오전 9:56	FILTERS 파일	1KB
cmake_install.cmake	2026-03-18 오전 9:56	CMake 원본 파일	3KB
CMakeCache.txt	2026-03-18 오전 9:56	텍스트 문서	25KB
glskeleton.slnx	2026-03-18 오전 9:56	Visual Studio Solution	4KB
glskeleton.vcxproj	2026-03-18 오전 9:56	VCXPROJ 파일	71KB
glskeleton.vcxproj.filters	2026-03-18 오전 9:56	FILTERS 파일	1KB
INSTALL.vcxproj	2026-03-18 오전 9:56	VCXPROJ 파일	11KB
INSTALL.vcxproj.filters	2026-03-18 오전 9:56	FILTERS 파일	1KB
ZERO_CHECK.vcxproj	2026-03-18 오전 9:56	VCXPROJ 파일	63KB
ZERO_CHECK.vcxproj.filters	2026-03-18 오전 9:56	FILTERS 파일	1KB

5



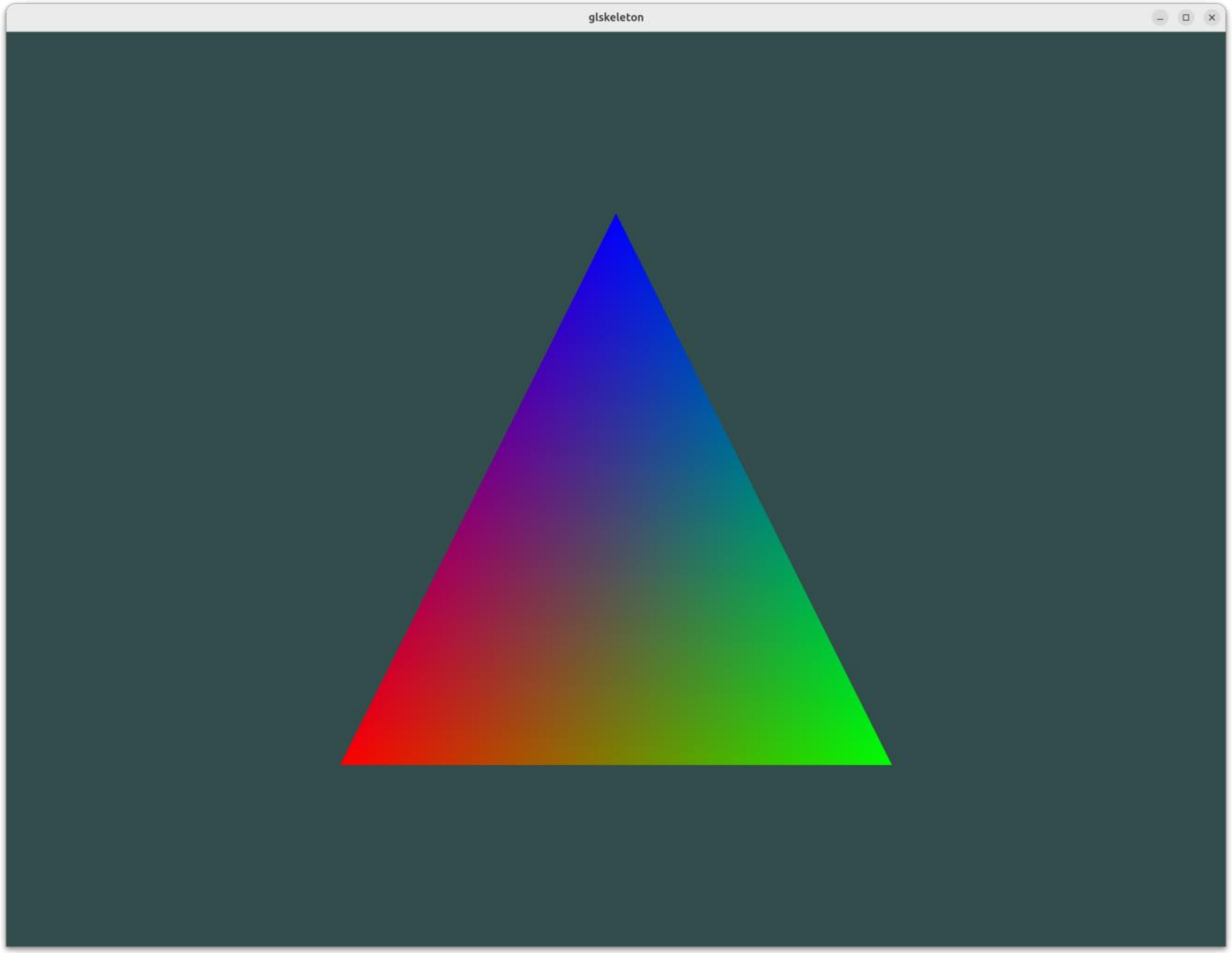
1. right click

2. left click

4. Open the solution file
5. Set **glskeleton** as Startup Project

Setup the assignment

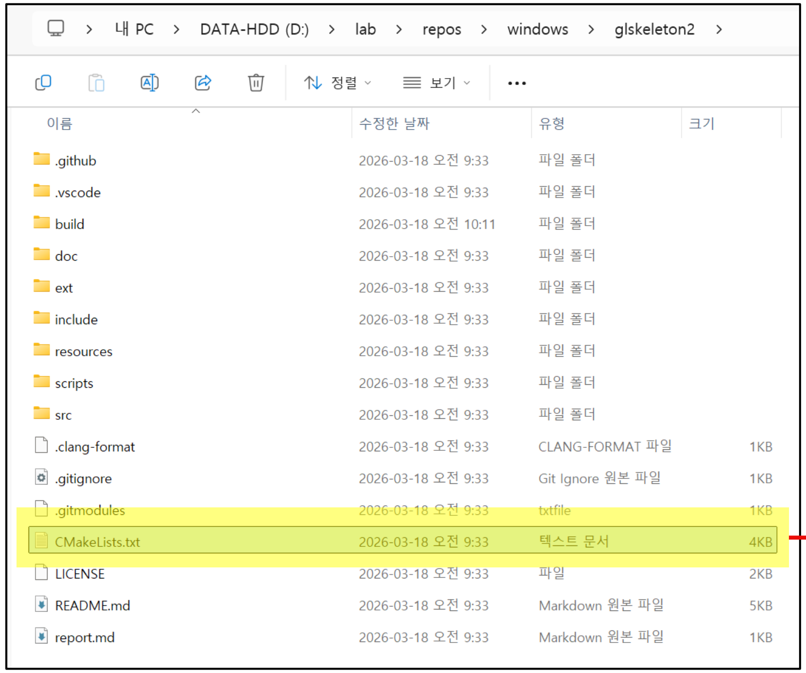
- After that, hit F5 (or Ctrl + F5 for Release mode) then you will see a triangle!



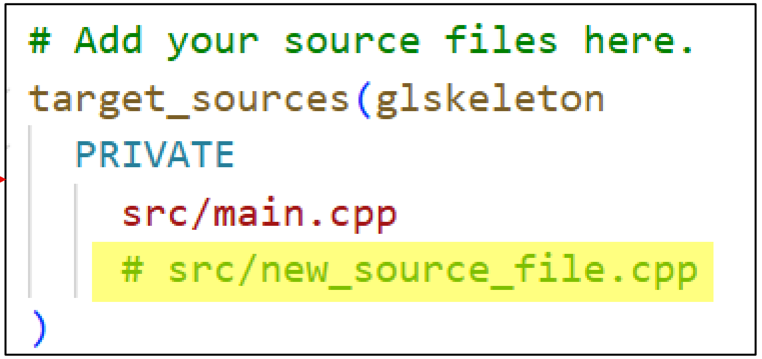
Setup the assignment

- Whatever your OS is, you can add a new source file into your repository using CMake as follows

1. Add a path to the new source file in <root_path>/CMakeLists.txt


내 PC › DATA-HDD (D:) › lab › repos › windows › glskeleton2 ›
정렬
보기
...

이름	수정한 날짜	유형	크기
.github	2026-03-18 오전 9:33	파일 폴더	
.vscode	2026-03-18 오전 9:33	파일 폴더	
build	2026-03-18 오전 10:11	파일 폴더	
doc	2026-03-18 오전 9:33	파일 폴더	
ext	2026-03-18 오전 9:33	파일 폴더	
include	2026-03-18 오전 9:33	파일 폴더	
resources	2026-03-18 오전 9:33	파일 폴더	
scripts	2026-03-18 오전 9:33	파일 폴더	
src	2026-03-18 오전 9:33	파일 폴더	
.clang-format	2026-03-18 오전 9:33	CLANG-FORMAT 파일	1KB
.gitignore	2026-03-18 오전 9:33	Git Ignore 원본 파일	1KB
.gitmodules	2026-03-18 오전 9:33	txtfile	1KB
CMakeLists.txt	2026-03-18 오전 9:33	텍스트 문서	4KB
LICENSE	2026-03-18 오전 9:33	파일	2KB
README.md	2026-03-18 오전 9:33	Markdown 원본 파일	5KB
report.md	2026-03-18 오전 9:33	Markdown 원본 파일	1KB



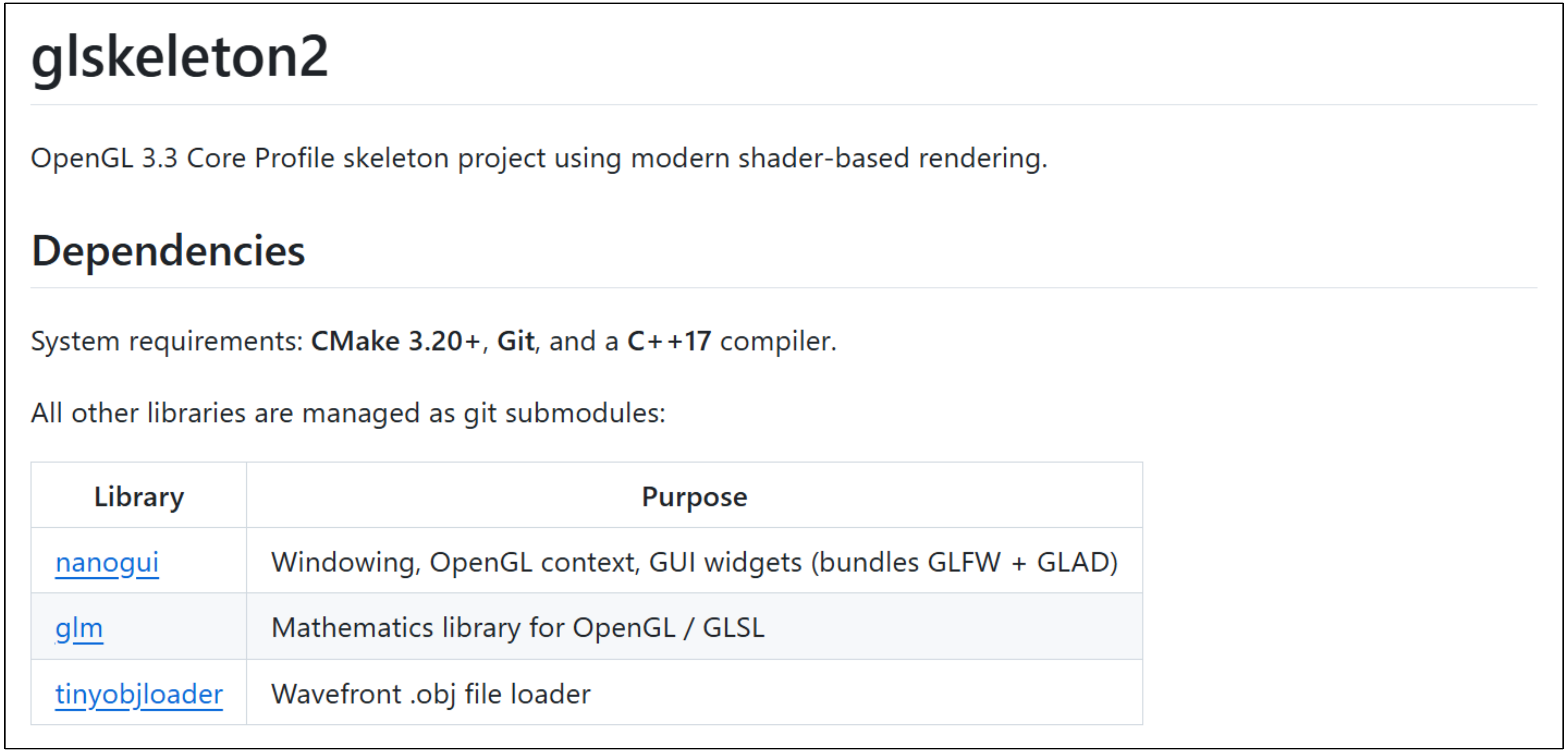
```
# Add your source files here.
target_sources(glskeleton
  PRIVATE
    src/main.cpp
    # src/new_source_file.cpp
)
```

2. Create a build system again to reflect changes ➔ cmake –B build

3. Continue your work

Setup the assignment

- Further detailed explanations, refer to README.md in your repo



glskeleton2

OpenGL 3.3 Core Profile skeleton project using modern shader-based rendering.

Dependencies

System requirements: **CMake 3.20+**, **Git**, and a **C++17** compiler.

All other libraries are managed as git submodules:

Library	Purpose
nanogui	Windowing, OpenGL context, GUI widgets (bundles GLFW + GLAD)
glm	Mathematics library for OpenGL / GLSL
tinyobjloader	Wavefront .obj file loader

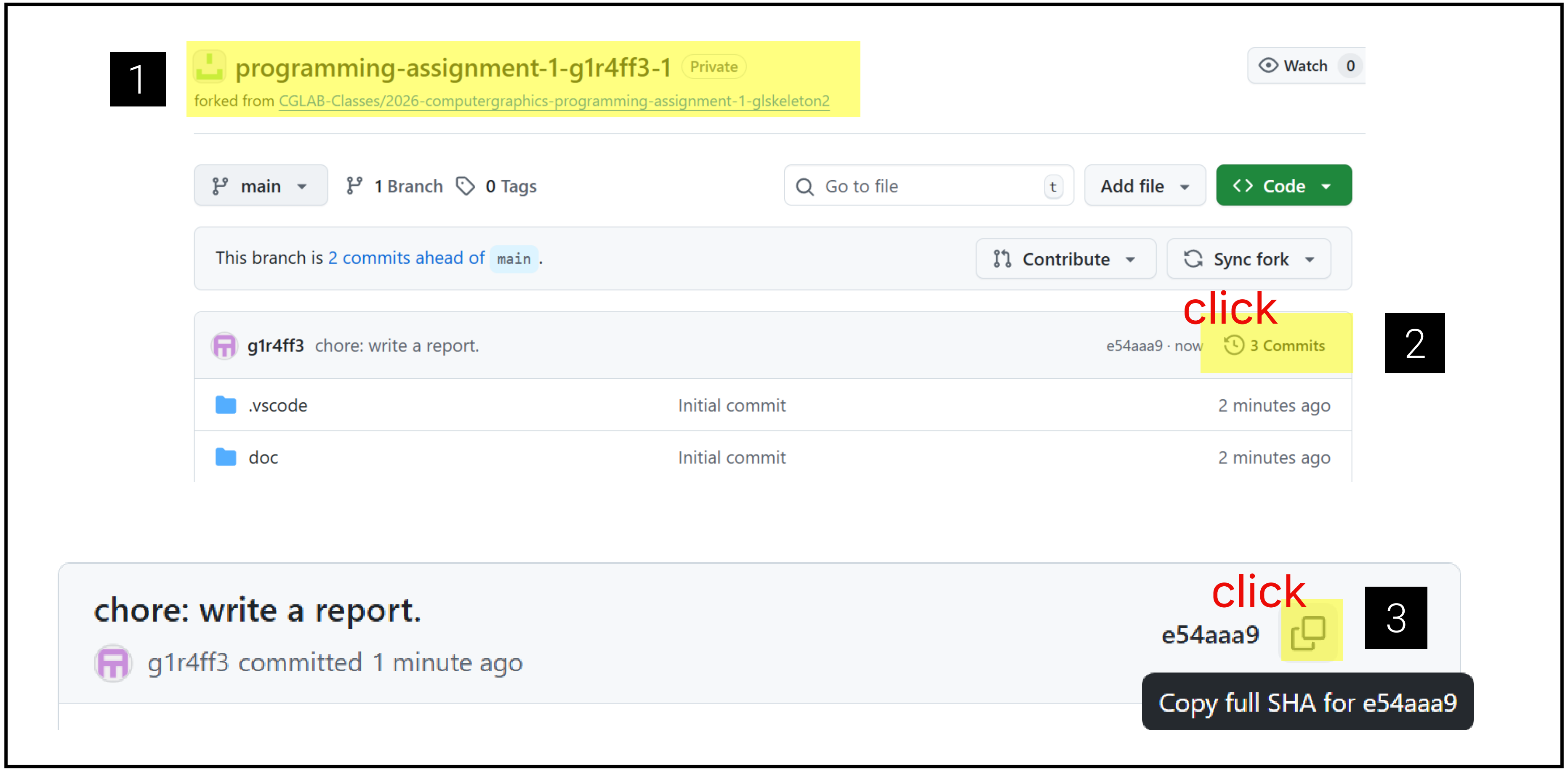
Submission

- Deadline: 23:59:59, Sunday, April 5th, 2026 (KST, +0900)
  - GitHub server clock

- To submit your assignment, two things must be done BEFORE deadline.
  - You must push your commit to your repository
  - You must comment the last commit id (SHA-1 hash) in github issue board

- The last commit in the issue board BEFORE deadline will be considered as submitted assignment.
  - Local timestamp in your commit will be ignored. GitHub server timestamp used instead

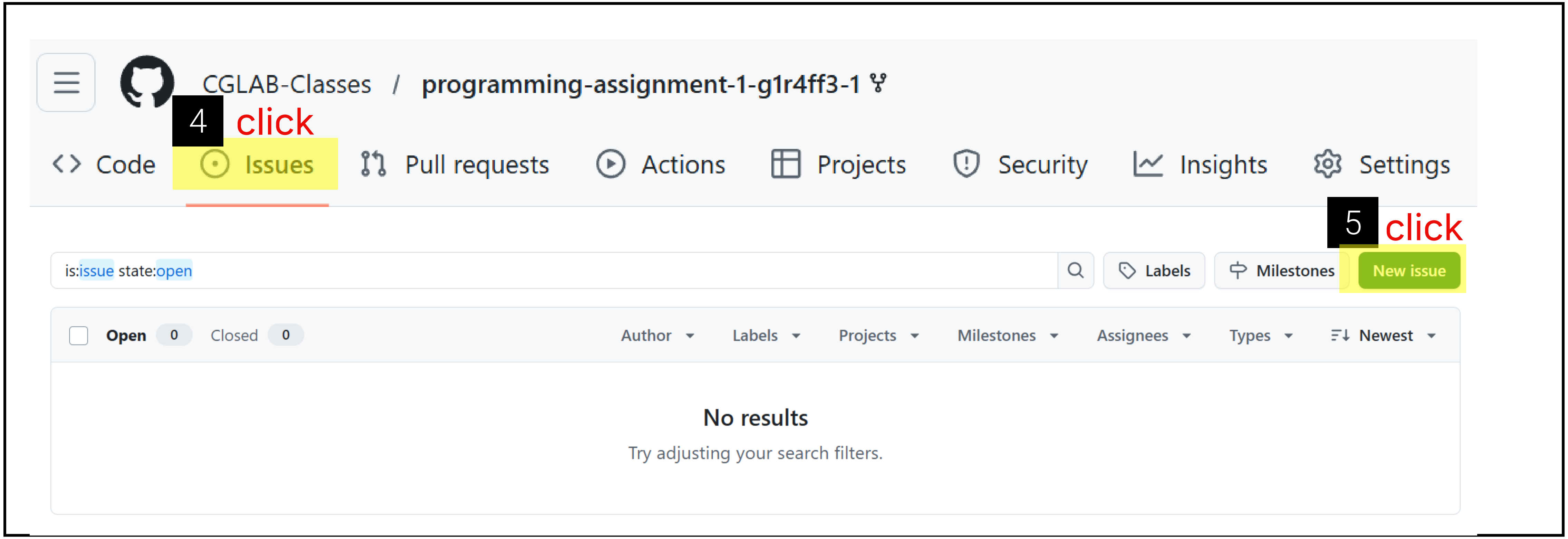
How to comment a commit id



1
programming-assignment-1-g1r4ff3-1
Private
forked from CGLAB-Classes/2026-computergraphics-programming-assignment-1-glskeleton2
Watch 0
main
1 Branch
0 Tags
Go to file
t
Add file
Code
This branch is 2 commits ahead of main .
Contribute
Sync fork
click
2
g1r4ff3 chore: write a report.
e54aaa9 · now
3 Commits
.vscode
Initial commit
2 minutes ago
doc
Initial commit
2 minutes ago
chore: write a report.
g1r4ff3 committed 1 minute ago
click
e54aaa9
3
Copy full SHA for e54aaa9

1. Go to your assignment repository
2. Click "Commits"
3. Click copy button of your last commit

How to comment a commit id



CGLAB-Classes / programming-assignment-1-g1r4ff3-1

4 click

5 click

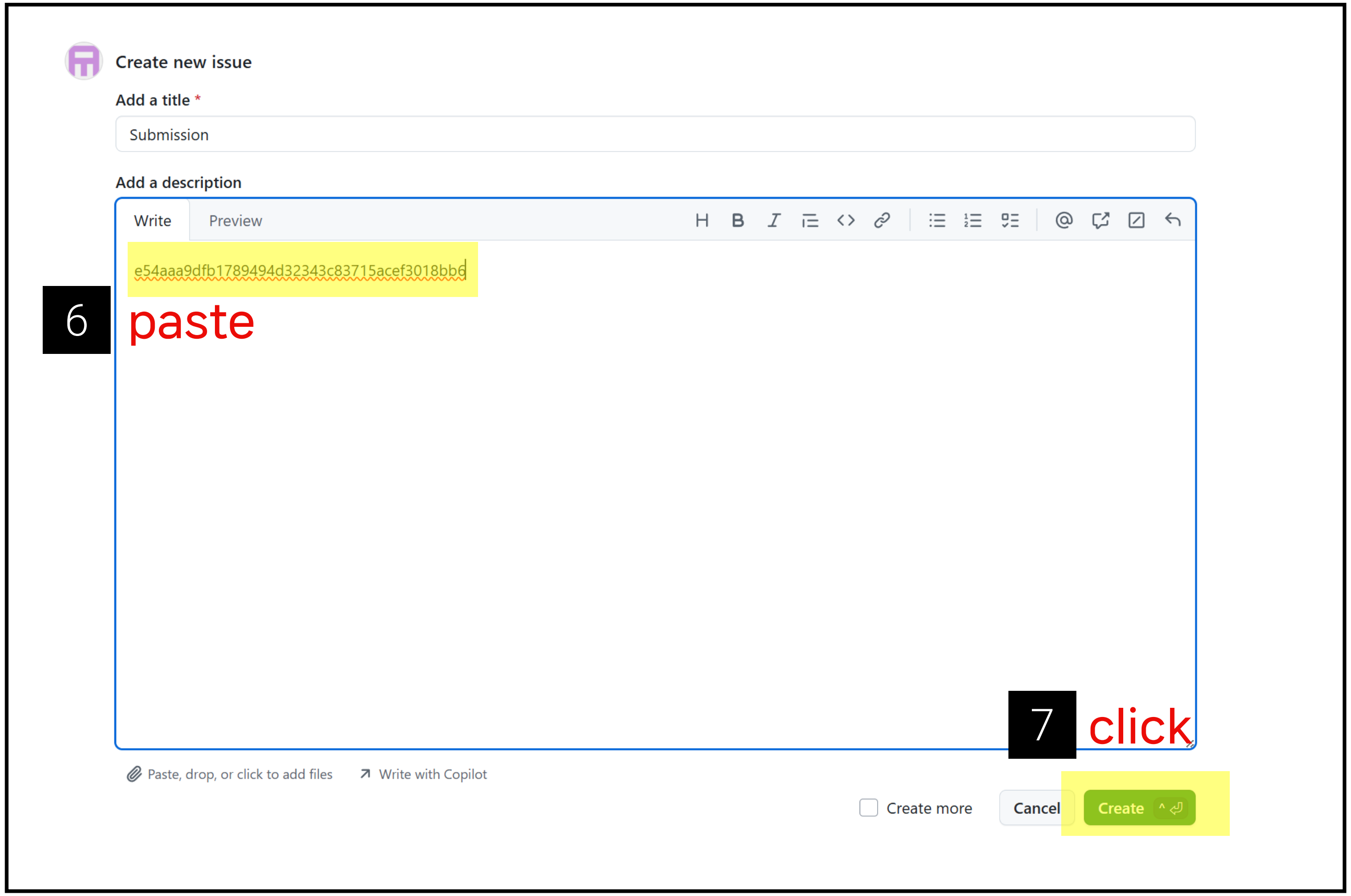
is:issue state:open

No results

Try adjusting your search filters.

4. Go to issues tab
5. Click "New Issue"

How to comment a commit id



Create new issue

Add a title *

Submission

Add a description

e54aaa9dfb1789494d32343c83715acef3018bb6

6. Paste your latest commit id (ctrl-v)
7. Click "Create"

Policies

- In following cases, your grade for this PA will be "0"
  - Late submission
    - late push after deadline or late last commit id comment on issue board
  - Build/execution failure
  - Making public of your assignment repository

- Your final grade will be "F"
  - Copy (We will run a copy detector)

Assignment Overview

Introduction to OpenGL Core Profile

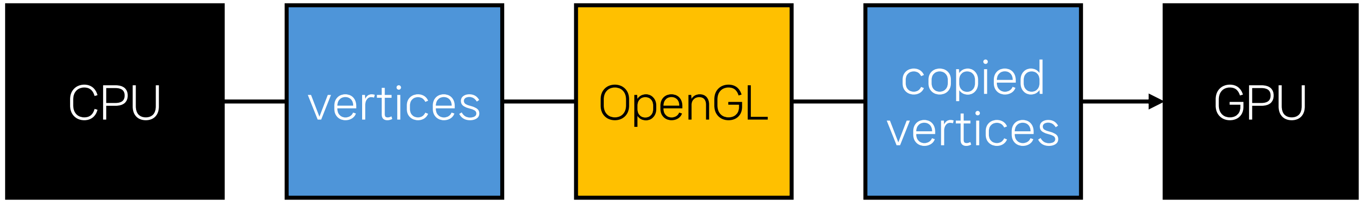
- You have learned what the rendering pipeline in OpenGL is
  - Ex. basic gl* functions are immediate mode (glBegin, glEnd, glColorXf, glVertexXf, …)

- Main objective of this assignment is to understand how modern graphics pipeline processes graphical operations

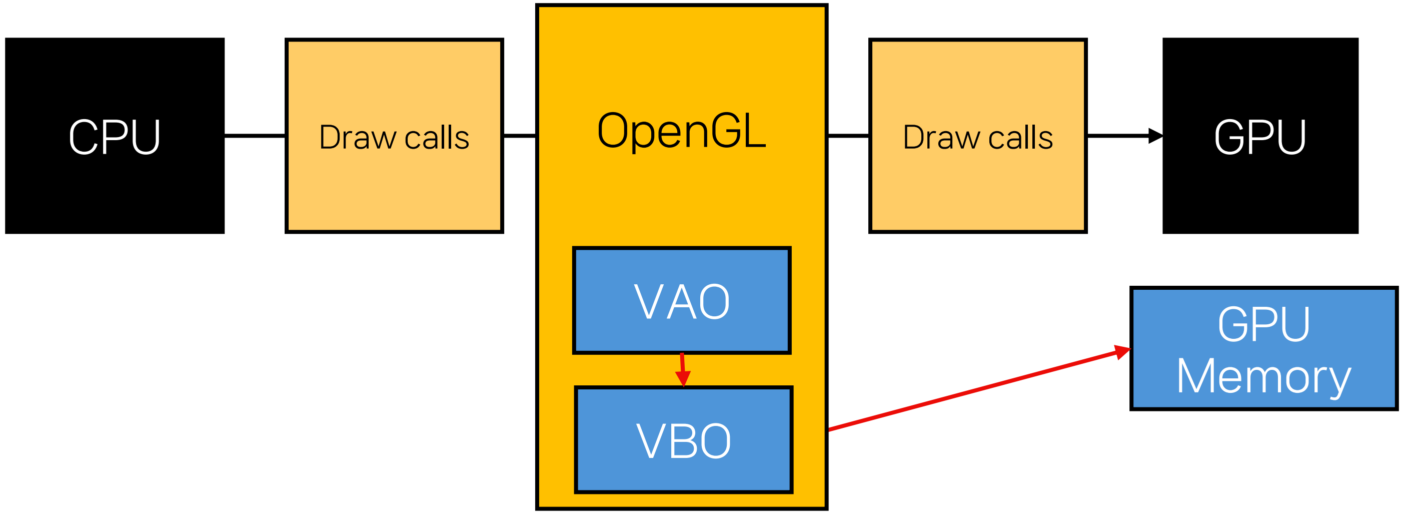
- There is another rendering mode which is more flexible and modern: Core Profile
  - Efficiency: upload scene data on GPU and render scenes in parallel
  - Flexibility: write programs (shaders) to change behaviors of the pipeline

Introduction to OpenGL Core Profile

- In immediate mode, we use glBegin/glEnd to load vertex data on GPU from CPU
  - In each frame, copy vertices in CPU and load them on GPU
  - This inherently blocks executing commands in parallel for waiting CPU

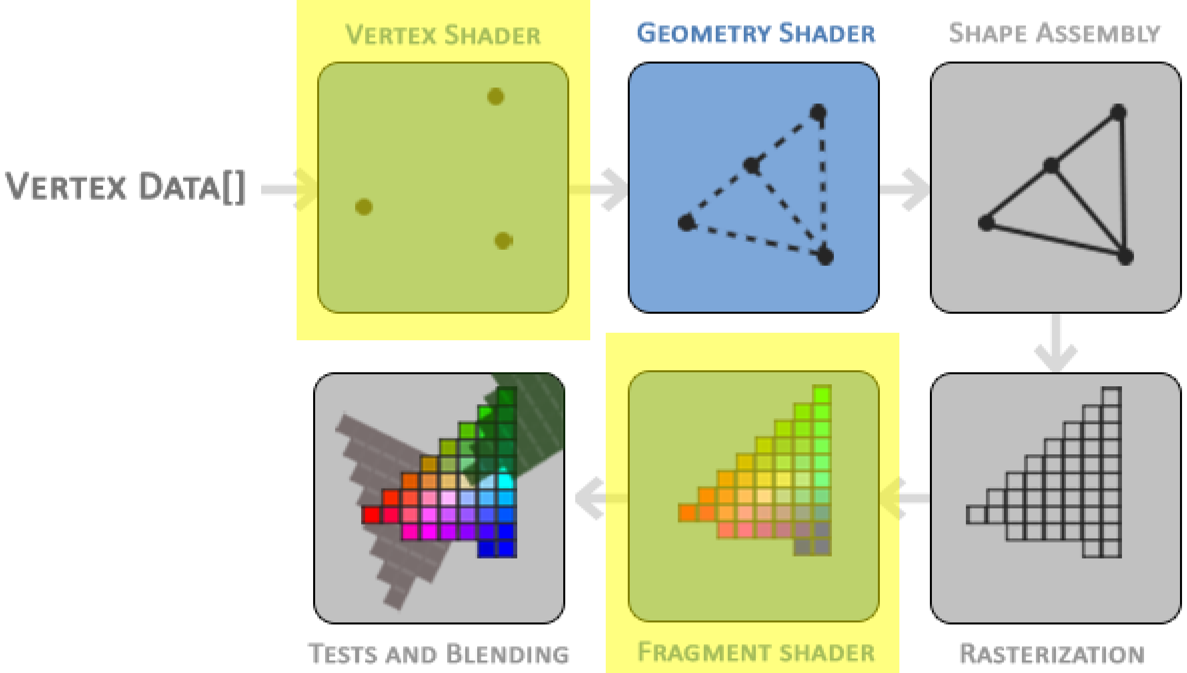


- In core profile mode, we use VBO/VAO to load vertex data on GPU from CPU if needed
  - Before rendering, copy vertices in CPU and load them on GPU (same above)
  - While rendering, we can use loaded data, and OpenGL can handle the data freely not waiting the CPU



Introduction to Shaders

- Just like small C programs lived in the rendering pipeline
  - Have programming language which is different according to graphics API
  - Transform inputs to outputs generally
  - Isolated in that they are not allowed to communicate with each other
- Most important two types of shaders: vertex and fragment
  - Vertex shader: process input vertices (ex. transformation)
  - Fragment shader: process final output colors to be seen



Tasks

1. Understand OpenGL in core profile mode, and practice modern OpenGL [5 Points]
   - Rotate your triangle with respect to time [2 Points]
   - Change your triangle color with respect to time [3 Points]
2. Abstract geometry loading and drawing process to Mesh class [8 Points]
   - Design your own Mesh class (mesh.h, mesh.cpp) [4 Points]
   - Draw bunny.obj same as task 1 alongside your triangle, which is located at *resources* directory [4 Points]
3. Write a report [2 Points]
   - Write your name, student id, GitHub username in report.md [1 Points]
   - Attach at least two result images in report.md [1 Points]

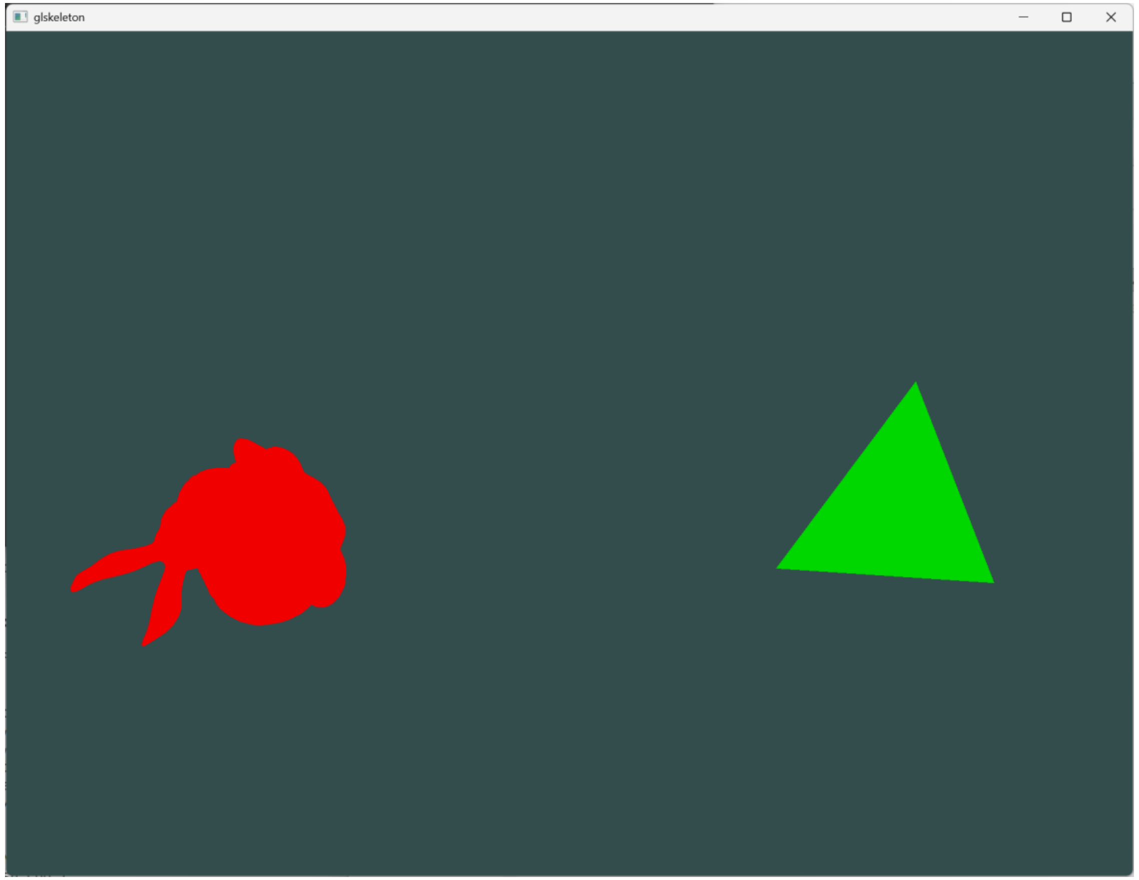
Tips for tasks

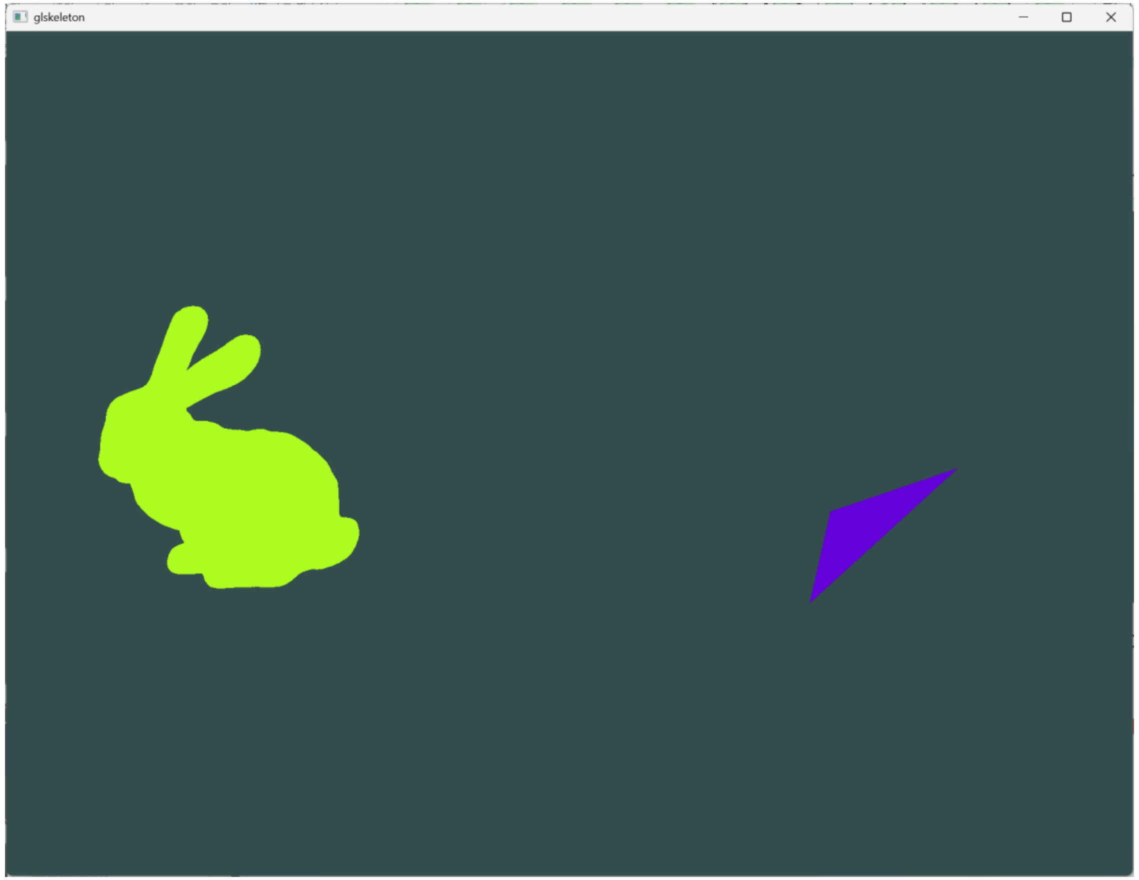
- Task 1
  - Before you start, first try to understand modern OpenGL concepts in the skeleton code
  - You had better read through the early part of OpenGL tutorial like LearnOpenGL
  - Hints
    - **glfwGetTime** function to figure out current time.
    - Use **sin** and **cos** function in <**cmath**> (or you can use **glm::sin, glm::cos** in <**glm/glm.hpp**>)
- Task 2
  - This task aims to check if you can abstract some programming patterns for later assignments
  - Hints
    - Use **tinyobj::LoadObj** in <**tiny_obj_loader.h**> to load the obj mesh
    - https://github.com/tinyobjloader/tinyobjloader#usage

Additional resources

- LearnOpenGL: Modern OpenGL Tutorial
  - https://learnopengl.com/
  - Refer to **Getting Started/{OpenGL, Hello Triangle, Shaders}** for understanding core profile
  - Refer to **Model Loading** to implement your Mesh class
- Obj file format
  - https://www.scratchapixel.com/lessons/3d-basic-rendering/obj-file-format/obj-file-format.html
- Git cheat sheet
  - https://education.github.com/git-cheat-sheet-education.pdf
- Github flavored Markdown cheat sheet
  - https://enterprise.github.com/downloads/en/markdown-cheatsheet.pdf

Demonstration





Rotating triangle and bunny changing their colors with respect to time

Q & A

- Email
  - Seongil Kim: sngillkym@gm.gist.ac.kr
  - Hyunjin Jung: hjjung0810@gm.gist.ac.kr
- Office: 104 Dasan Bldg