CSCI 3260
Principles of Computer Graphics

--------Tutorial 1

XU Jiaqi

About this course

- XU Jiaqi (jqxu@cse.cuhk.edu.hk)
  Office: SHB 1024
  Office hour: Friday 3:30pm-5:30pm
- **Tutorial hours:**
  Monday 3:30pm-4:15pm
  Thursday 5:30pm-6:15pm
- **Tutorial notes:**
  Blackboard system
- **Program language:**
  OpenGL, C++

Basic schedule

	Announce on	Due on
➢ Assignment 1	12/9 (Mon)	2/10 (Sun)
➢ Assignment 2		
➢ Course project		
➢ Mid-term exam		
➢ Final exam		

Outline

- Introduction to OpenGL
- Setup OpenGL environment

Introduction

Introduction

OpenGL (Open Graphics Library):

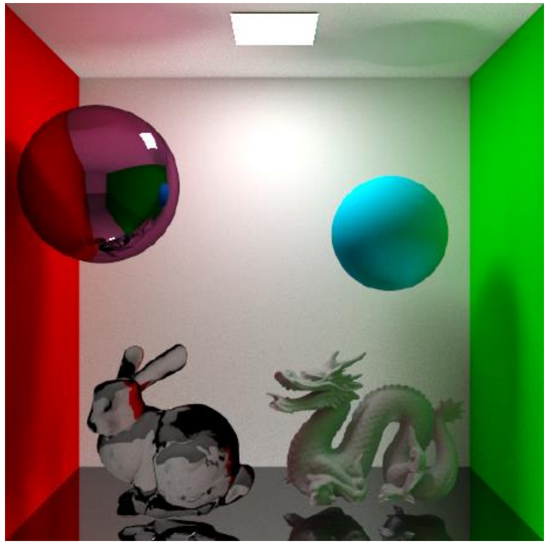
- A cross-language, cross-platform application programming interface (**API**) for rendering 2D and 3D graphics & communicate with graphic hardware (GPU)
- Official website: *https://www.opengl.org/*
- Silicon Graphics Inc. (SGI) developed OpenGL in 1991, and the latest version is OpenGL 4.6.
- Widely used in computer-aided design, virtual reality, visualization, games, etc.
- Low-level implementation.

OpenGL vs. DirectX:

- Both 2D/3D graphics API
- OpenGL is multi-platform; DX is for Windows only
- DX is a more complicated API (powerful for sound and video)

Introduction

Examples programmed in OpenGL:



Introduction

More about OpenGL:

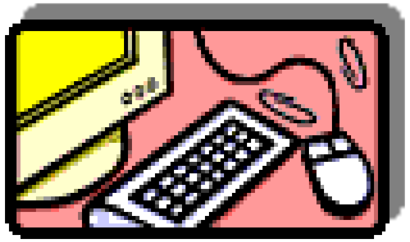
- WebGL (Web Graphics Library)
  - A JavaScript API for rendering interactive 3D computer graphics and 2D graphics within any compatible web browser.
  - Multi-platform
  - WebGL is widely supported in modern browsers, including desktop browsers and mobile browsers, such as Google Chrome, Safari, Firefox, Microsoft Edge, etc.
- WebGL samples: http://webglsamples.org/

- Other online rendering tools:
  - Shadertoy, etc.

Introduction

What OpenGL (solely) doesn’t do:

- OpenGL (solely) cannot deal with commands for performing windowing/ event system are provided



To develop an interactive graphics application, other OpenGL related utility libraries are required.

Introduction

OpenGL related libraries:

- **UI library**

  (help create and manage windows, as well as handle joystick, keyboard and mouse input)

  - OpenGL Utility Toolkit Library (GLUT) ---- no longer maintained
  - GLFW (*http://www.glfw.org/*)
  - FreeGLUT (*http://freeglut.sourceforge.net/*)

- **Extension library (query and load OpenGL extensions)**
  - OpenGL Extension Wrangler Library (GLEW) (*http://glew.sourceforge.net/*)

- **Mathematical library**
  - OpenGL Mathematics (GLM) (*https://glm.g-truc.net/0.9.9/index.html*)

Introduction

OpenGL related libraries:

- **UI library**

  (help create and manage windows, as well as handle joystick, keyboard and mouse input)

  - OpenGL Utility Toolkit Library (GLUT) ---- no longer maintained
  - GLFW (*http://www.glfw.org/*)
  - FreeGLUT (*http://freeglut.sourceforge.net/*)

- **Extension library (query and load OpenGL extensions)**

  - OpenGL Extension Wrangler Library (GLEW) (*http://glew.sourceforge.net/*)

- **Mathematical library**

  - OpenGL Mathematics (GLM) (*https://glm.g-truc.net/0.9.9/index.html*)

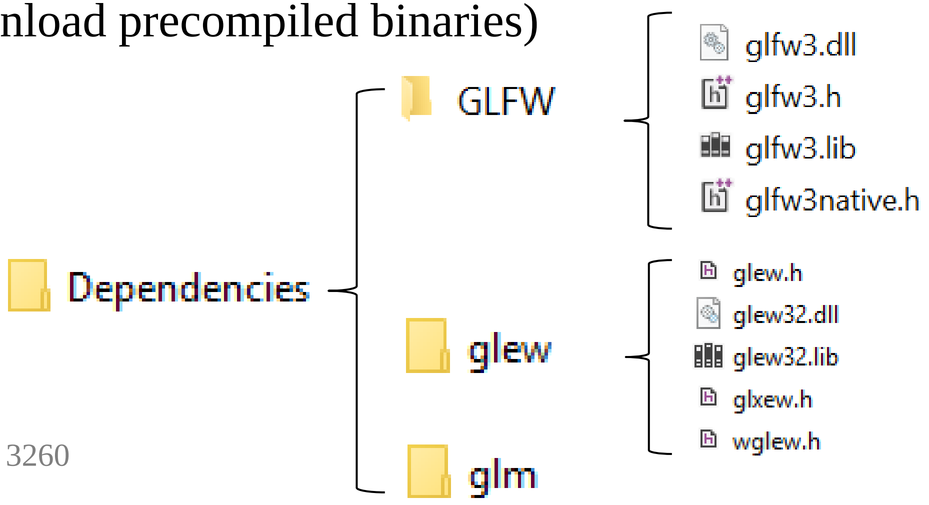
Setup OpenGL (for Windows)

Setup OpenGL environment (for Windows):

- Programming language: OpenGL & C++ (VS2022 Community is recommended)
- GLFW & GLEW & GLM (provided)

Resources:

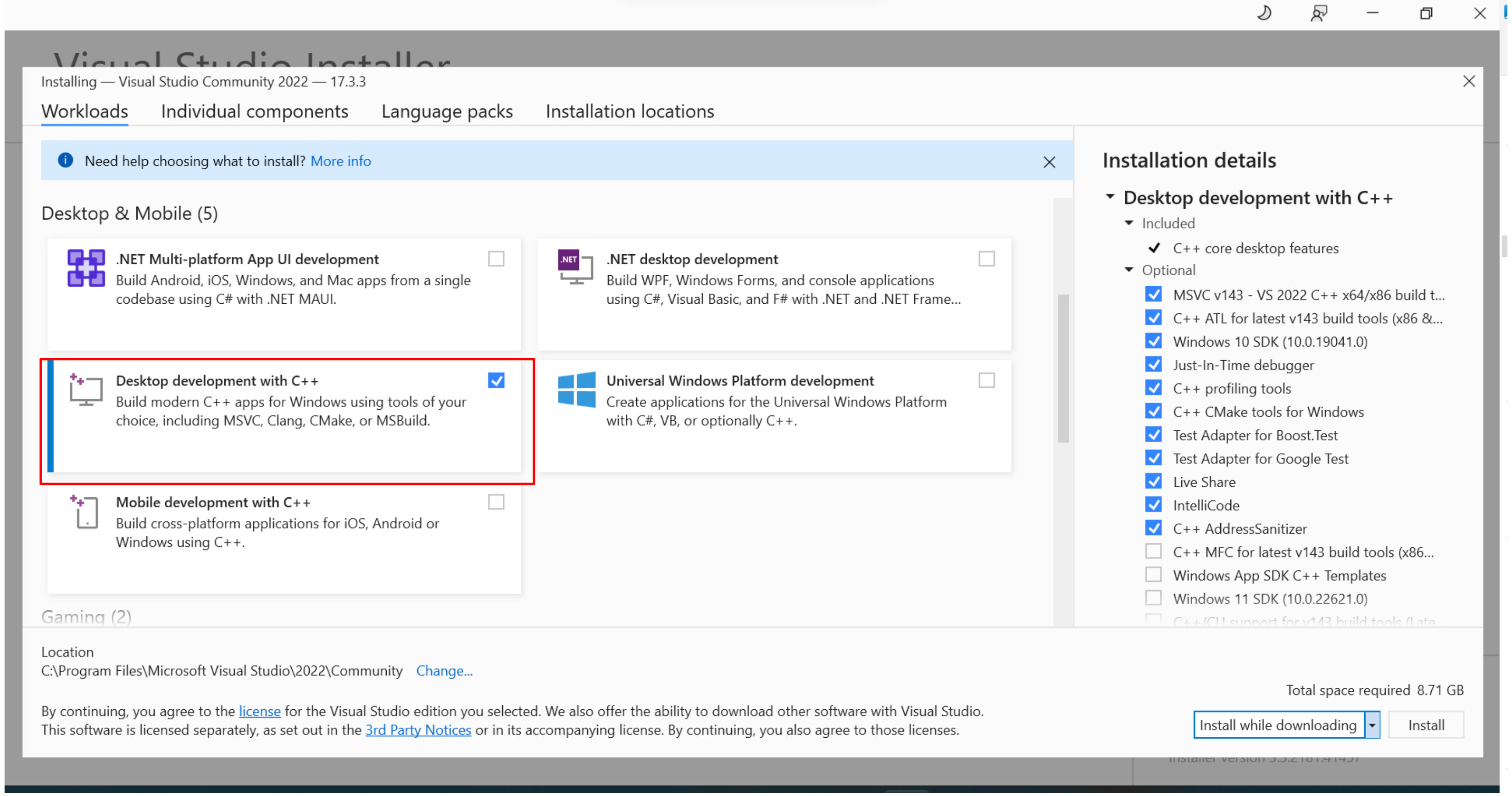
- VS2022: https://visualstudio.microsoft.com/downloads/
- GLEW: http://glew.sourceforge.net/ (download precompiled binaries)
- GLFW: https://www.glfw.org/download.html (download precompiled binaries)
- GLM: https://glm.g-truc.net/0.9.9/



Dependencies
- GLFW
  - glfw3.dll
  - glfw3.h
  - glfw3.lib
  - glfw3native.h
- glew
  - glew.h
  - glew32.dll
  - glew32.lib
  - glxew.h
  - wglew.h
- glm

Setup OpenGL (for Windows)

Install Visual Studio



Installing — Visual Studio Community 2022 — 17.3.3

Workloads | Individual components | Language packs | Installation locations

Need help choosing what to install? More info

Desktop & Mobile (5)

☐ **.NET Multi-platform App UI development**
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.

☐ **.NET desktop development**
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

☑ **Desktop development with C++**
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

☐ **Universal Windows Platform development**
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

☐ **Mobile development with C++**
Build cross-platform applications for iOS, Android or Windows using C++.

Gaming (2)

Installation details

Desktop development with C++

Included

✓ C++ core desktop features

Optional

- ☑ MSVC v143 - VS 2022 C++ x64/x86 build t...
- ☑ C++ ATL for latest v143 build tools (x86 &...
- ☑ Windows 10 SDK (10.0.19041.0)
- ☑ Just-In-Time debugger
- ☑ C++ profiling tools
- ☑ C++ CMake tools for Windows
- ☑ Test Adapter for Boost.Test
- ☑ Test Adapter for Google Test
- ☑ Live Share
- ☑ IntelliCode
- ☑ C++ AddressSanitizer
- ☐ C++ MFC for latest v143 build tools (x86...
- ☐ Windows App SDK C++ Templates
- ☐ Windows 11 SDK (10.0.22621.0)

Location
C:\Program Files\Microsoft Visual Studio\2022\Community Change...

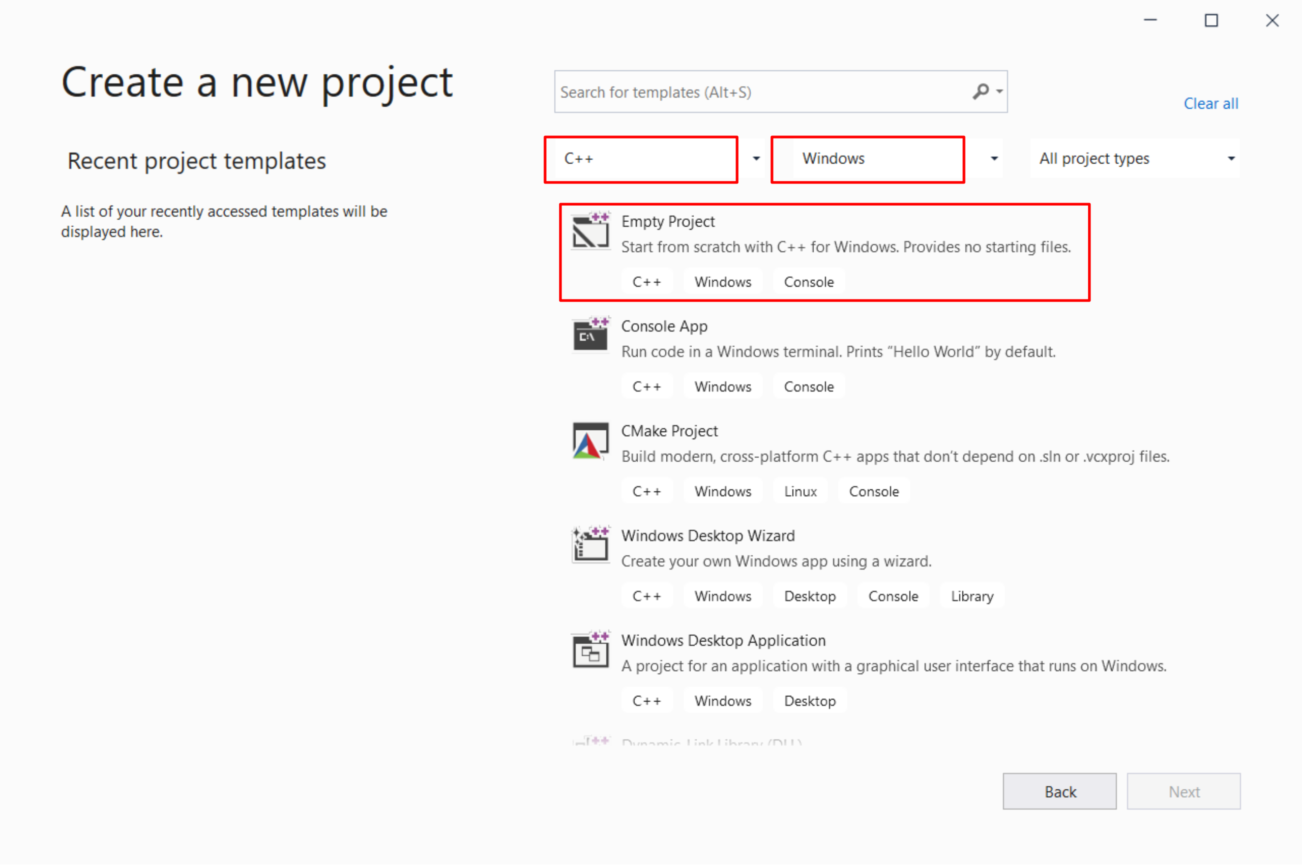
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 8.71 GB

Install while downloading | Install

Setup OpenGL (for Windows)

Create a new project



Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

Search for templates (Alt+S)

Clear all

C++ | Windows | All project types

Empty Project
Start from scratch with C++ for Windows. Provides no starting files.
C++ Windows Console

Console App
Run code in a Windows terminal. Prints "Hello World" by default.
C++ Windows Console

CMake Project
Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.
C++ Windows Linux Console

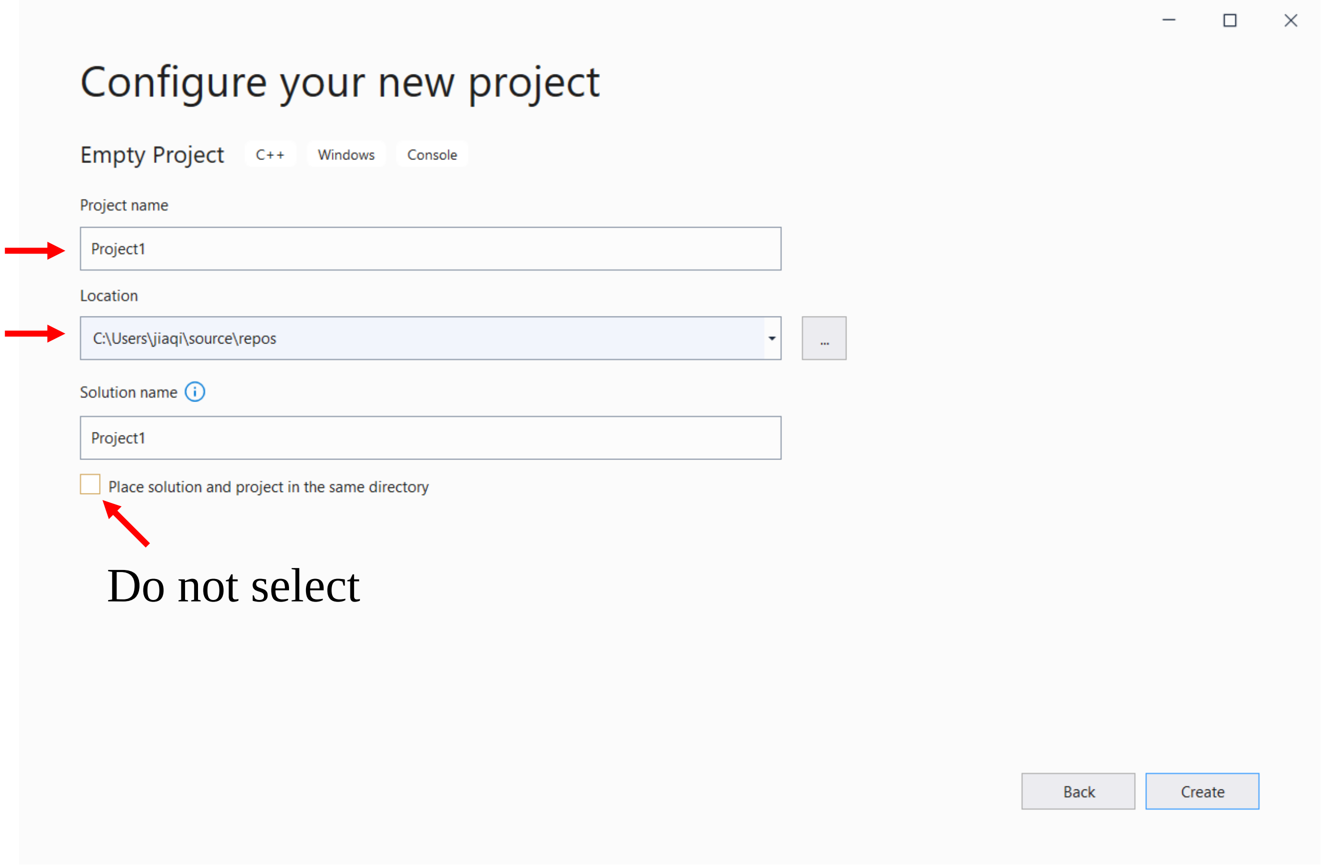
Windows Desktop Wizard
Create your own Windows app using a wizard.
C++ Windows Desktop Console Library

Windows Desktop Application
A project for an application with a graphical user interface that runs on Windows.
C++ Windows Desktop

Dynamic-Link Library (DLL)

Setup OpenGL (for Windows)

Configure the project



Configure your new project

Empty Project C++ Windows Console

Project name

Project1

Location

C:\Users\jiaqi\source\repos

Solution name

Project1

□ Place solution and project in the same directory

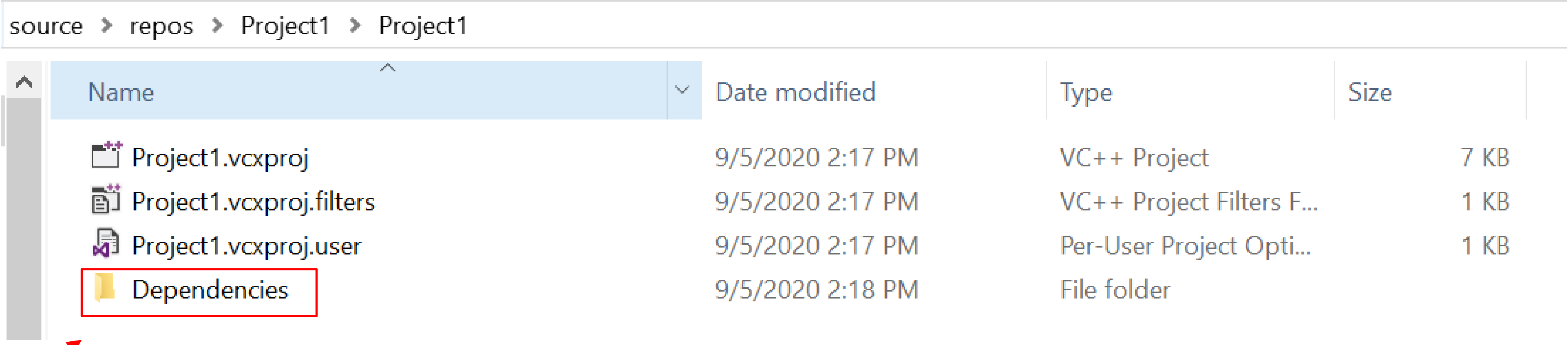
Give your project a name

Project location

Do not select

Setup OpenGL (for Windows)

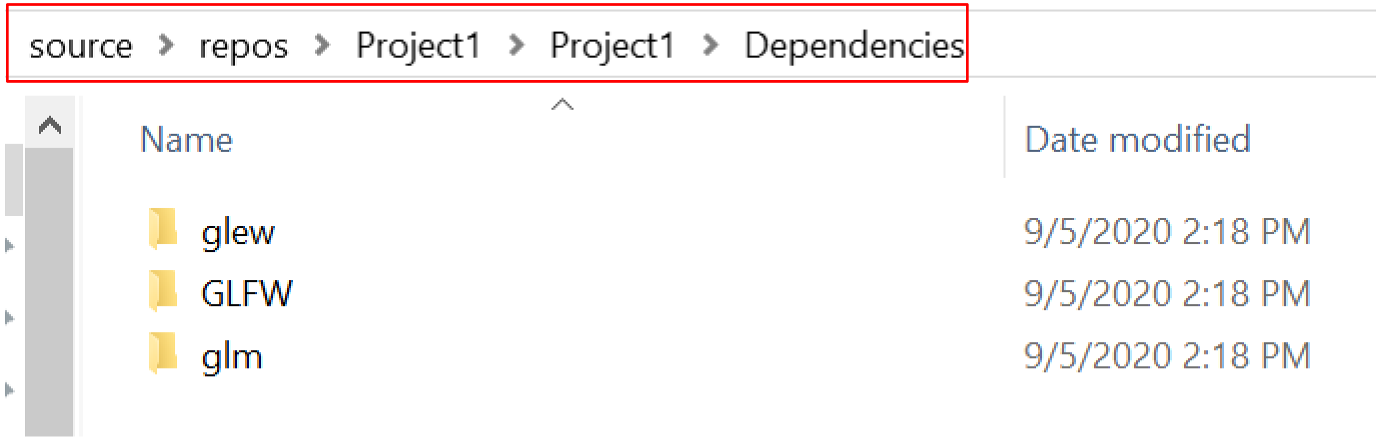
Copy “Dependencies” folder into your project folder.



source › repos › Project1 › Project1

Name	Date modified	Type	Size
Project1.vcxproj	9/5/2020 2:17 PM	VC++ Project	7 KB
Project1.vcxproj.filters	9/5/2020 2:17 PM	VC++ Project Filters F...	1 KB
Project1.vcxproj.user	9/5/2020 2:17 PM	Per-User Project Opti...	1 KB
Dependencies	9/5/2020 2:18 PM	File folder	

main.cpp
glsl code should in the same folder

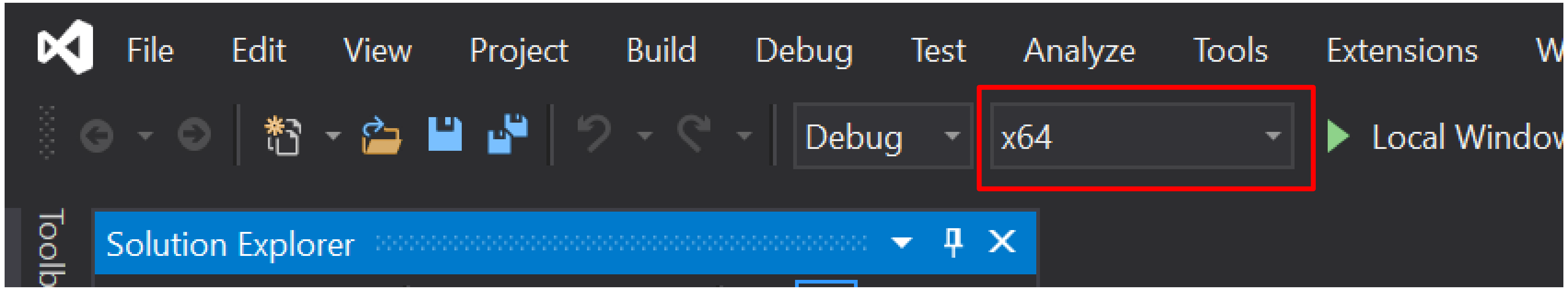


source › repos › Project1 › Project1 › Dependencies

Name	Date modified
glew	9/5/2020 2:18 PM
GLFW	9/5/2020 2:18 PM
glm	9/5/2020 2:18 PM

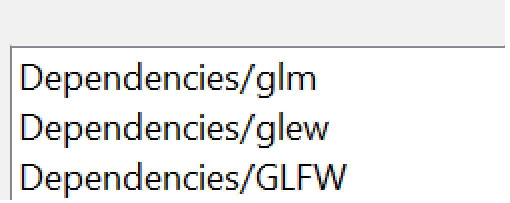
Setup OpenGL (for Windows)

Make sure your platform is ***x64***!



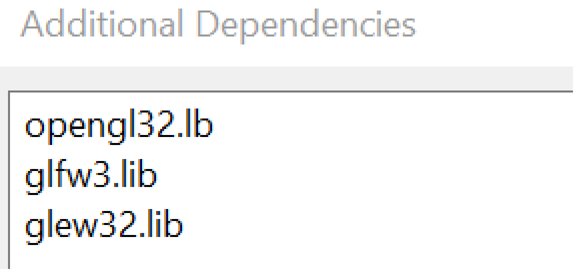
Setup OpenGL (for Windows)

1. Right click the project name ⟶ ***Properties***
2. ***Linker ⟶ General ⟶ Additional Library Directories***
3. Add the “GLFW” & “glew” & “glm” folder



Dependencies/glm
Dependencies/glew
Dependencies/GLFW

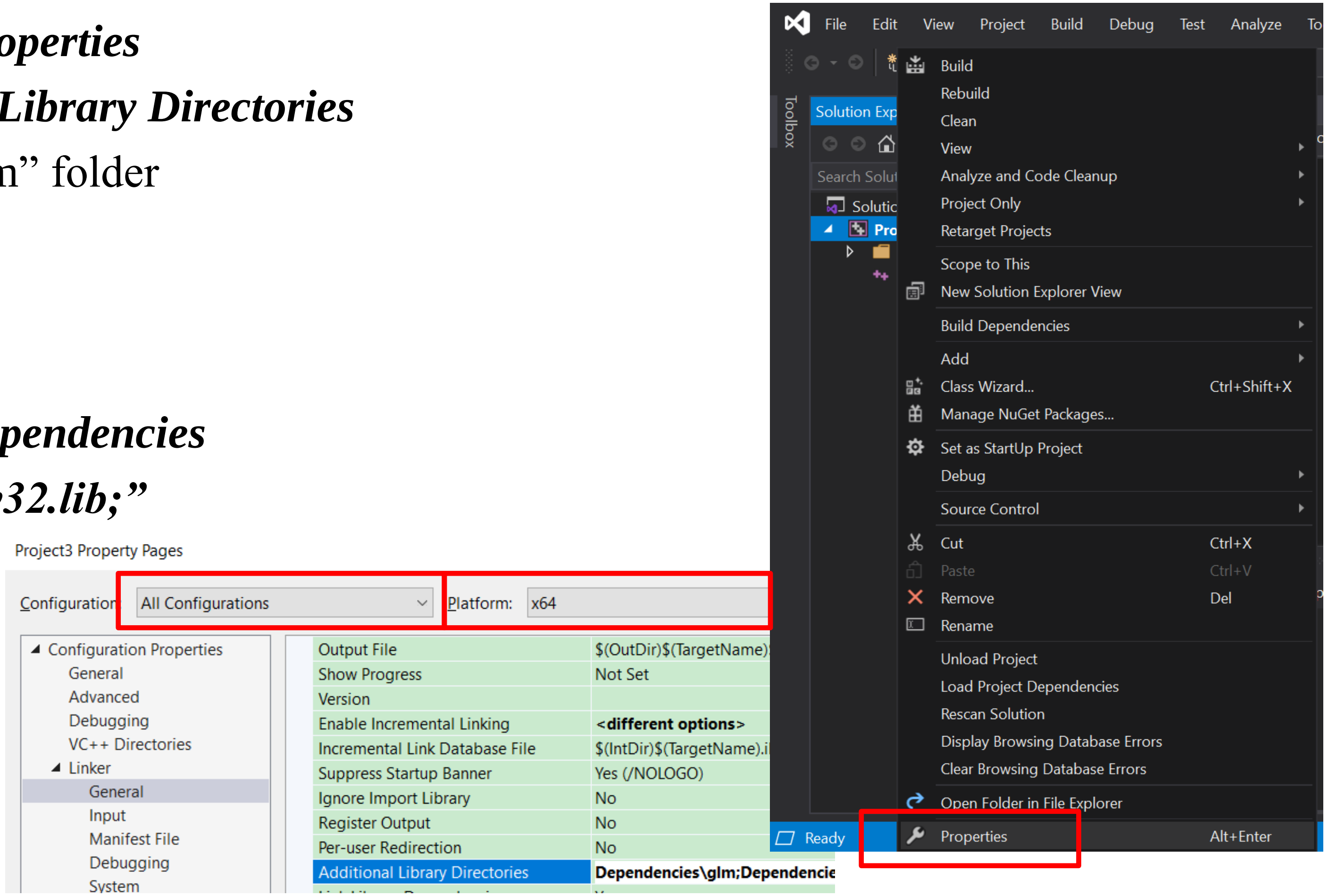
4. ***Linker ⟶ Input ⟶ Additional Dependencies***
5. Add ***“opengl32.lib; glfw3.lib; glew32.lib;”***



Additional Dependencies

opengl32.lb
glfw3.lib
glew32.lib

6. Press “***Apply***” & “***OK***”



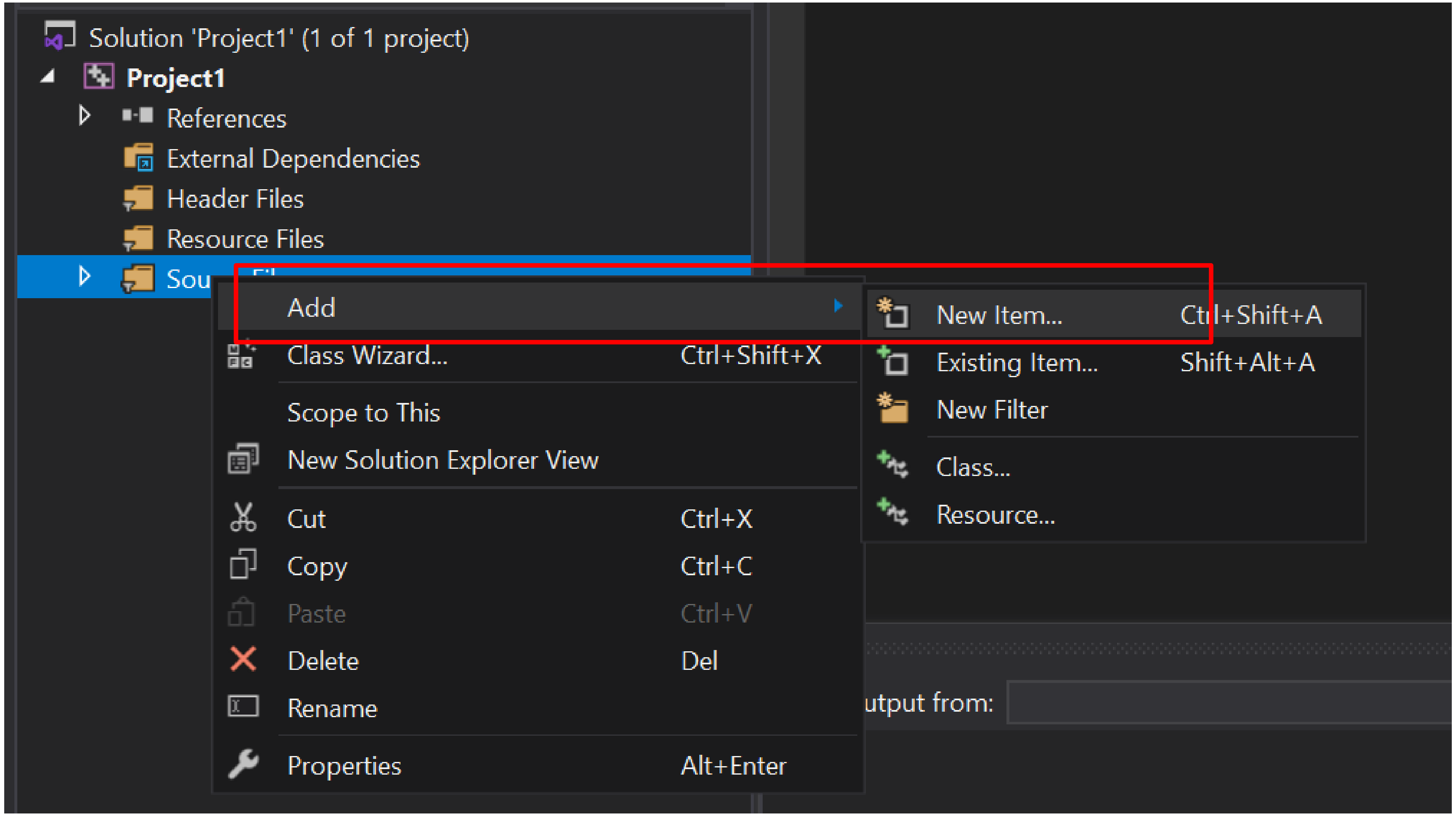
Project3 Property Pages

Configuration: All Configurations   Platform: x64

Output File	$(OutDir)$(TargetName)
Show Progress	Not Set
Version	
Enable Incremental Linking	**<different options>**
Incremental Link Database File	$(IntDir)$(TargetName).i
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**Dependencies\glm;Dependencie**

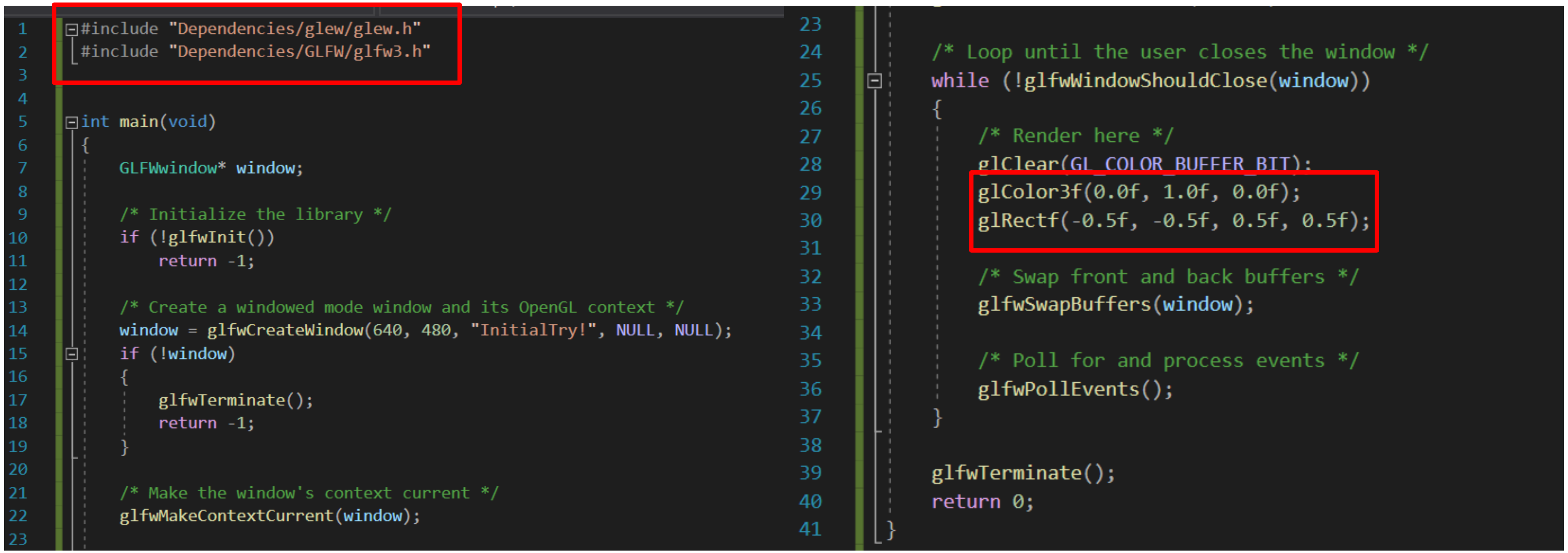
Setup OpenGL (for Windows)

Right click *Source Files* ⇒ *Add* ⇒ *New Item* to add a *main.cpp*.



Setup OpenGL (for Windows)

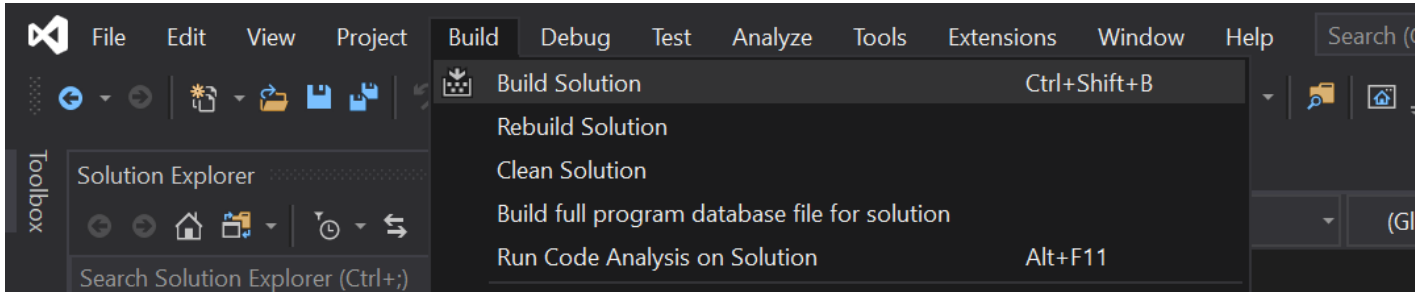
- Type this short code to test whether the OpenGL environment is setup successfully.
- Refer to https://www.glfw.org/documentation.html.
- Modify according to the red box.



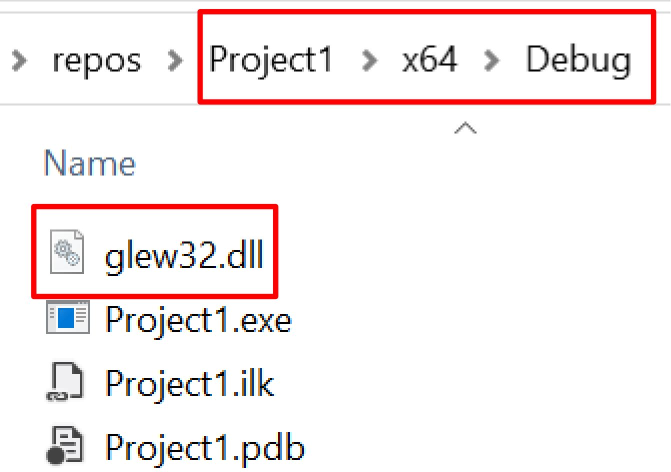
```
#include "Dependencies/glew/glew.h"
#include "Dependencies/GLFW/glfw3.h"


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, "InitialTry!", 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);
        glColor3f(0.0f, 1.0f, 0.0f);
        glRectf(-0.5f, -0.5f, 0.5f, 0.5f);

        /* Swap front and back buffers */
        glfwSwapBuffers(window);

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

    glfwTerminate();
    return 0;
}
```

Setup OpenGL (for Windows)

- Press *Build* ⇒ *Build Solution*.


File
Edit
View
Project
Build
Debug
Test
Analyze
Tools
Extensions
Window
Help
Search (
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Build full program database file for solution
Run Code Analysis on Solution Alt+F11
Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+;)
(Gl

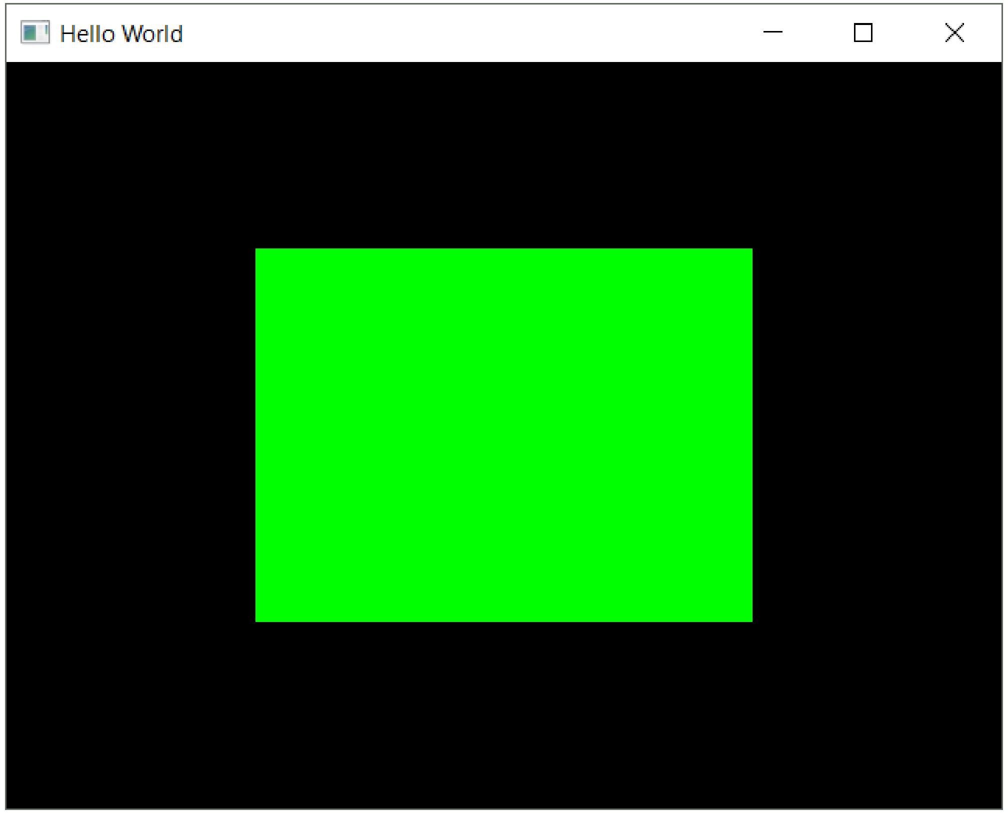
- If build successfully, the final step is to copy *glew32.dll* (in the *Dependencies* subfolder) to **the Debug folder that contains .exe item** (for later tutorial & assignment submission use).



- Press *F*5 to see the output.

Setup OpenGL (for Windows)

- Output image



Setup OpenGL (for MacOS)

1. See the self-study material.

2. YouTube video:
   https://www.youtube.com/watch?v=Tz0dq2krCW8&list=PLRtjMdoYXLf6zUMDJVRZYV-6g6n62vet8&index=1

3. Links (Chinese):
   https://www.cnblogs.com/yinxiangnan-charles/p/5002293.html
   http://blog.shenyuanluo.com/OpenGLEnvironment.html

Summary

You should know:

- What is OpenGL & related utility toolkit.
- How to setup OpenGL environment on your own computer.

Next tutorial

- Introduction to basic OpenGL programming.