XIAMEN UNIVERSITY

COMPUTER GRAPHICS

Lab 1: Hello OpenGL

Dr. Zhonggui Chen
School of Informatics, Xiamen University
http://graphics.xmu.edu.cn

What is OpenGL?

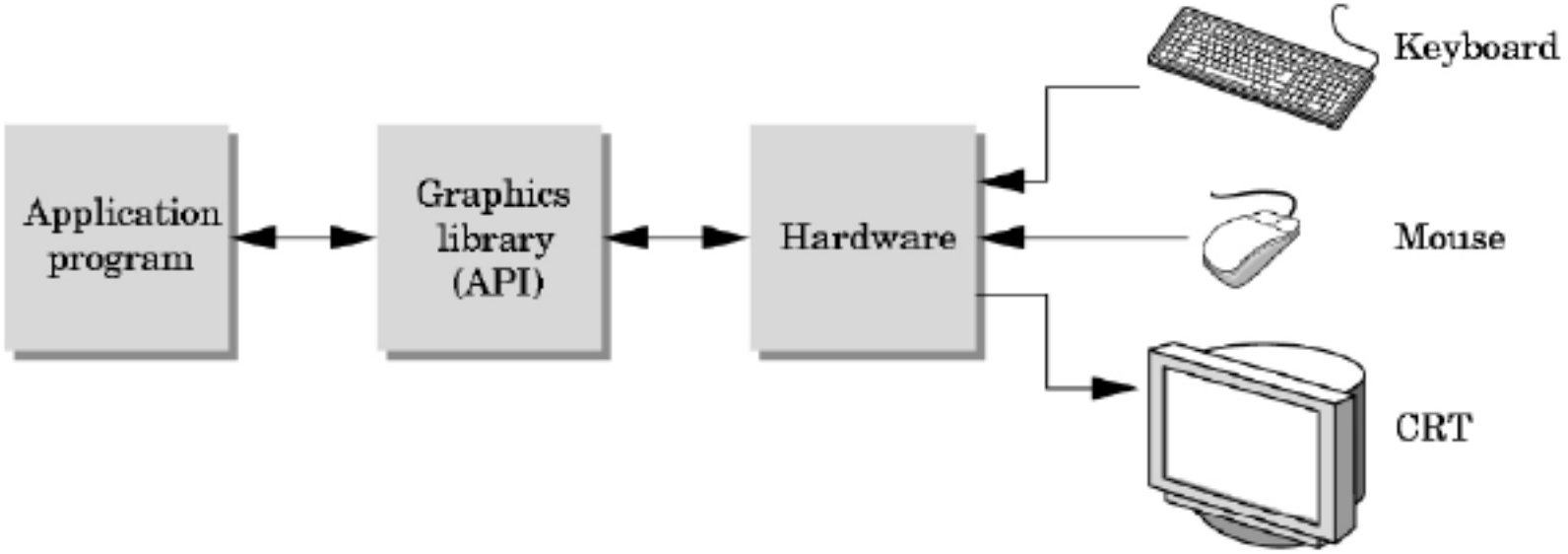
- Low-level, cross-platform graphics library (API) for 2D and 3D interactive Graphics.
- First version in 1992; now: 4.6 (2019)
- Managed by Khronos Group (non-profit)
- API is governed by Architecture Review Board (part of Khronos)!

Where is OpenGL used?

- CAD
- VR
- Scientific Visualization!
- Simulations
- Video games

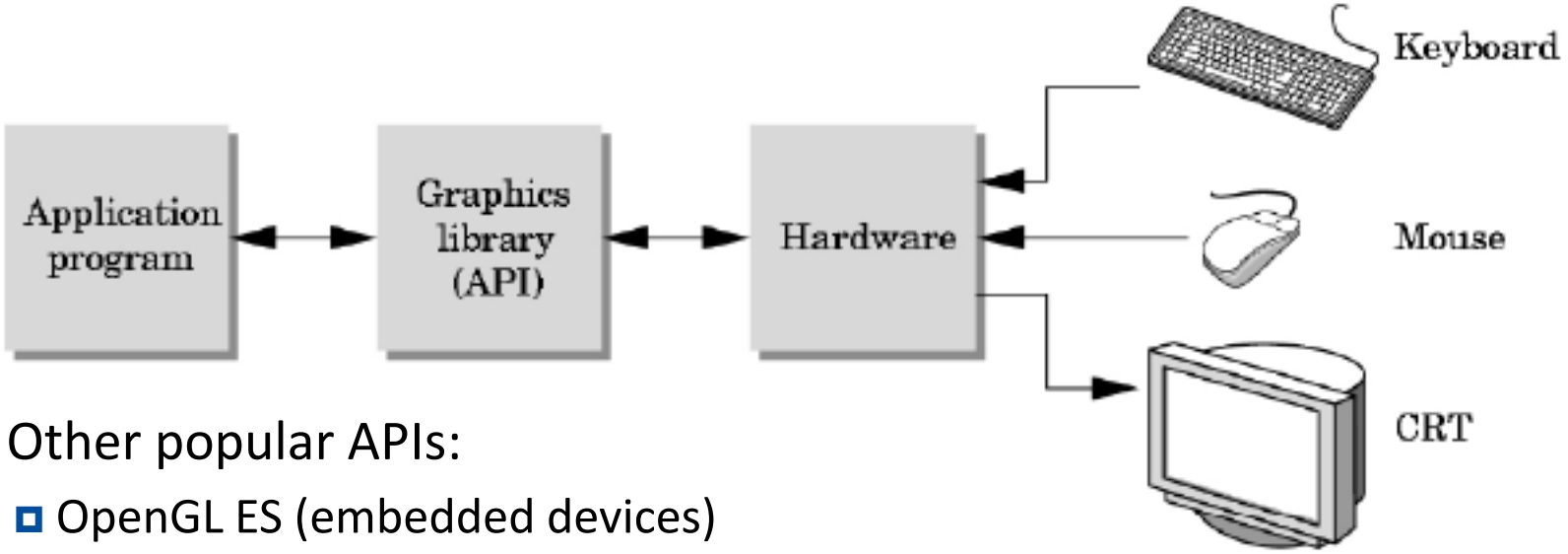
Graphics Library (API)

- **Interface** between Application and Graphics Hardware



Graphics Library (API)

- **Interface** between Application and Graphics Hardware



- Other popular APIs:
  - OpenGL ES (embedded devices)
  - WebGL (web browser)
    - https://webglsamples.org/
  - Direct3D (Microsoft)
  - Vulkan (next generation of OpenGL)

OpenGL库

- OpenGL核心库(OpenGL Core Library)
  - 函数名gl开头
  - Windows: opengl32.dll (WINDOWS\SYSTEM32)
  - 大多数Unix/Linux系统：GL库 (libGL.a)
- OpenGL实用库(OpenGL Utility Library, GLU)
  - OpenGL的一部分，函数名以glu开头
    - Windows: glu32.dll
  - 利用OpenGL实用库提供一些功能，避免重复编写代码
  - 二次曲面、NURBS、多边形网格化等

GLUT

- OpenGL实用工具库（OpenGL Utility Toolkit Library, GLUT）
  - 提供所有窗口系统的共同功能
    - 创建窗口
    - 从鼠标和键盘获取输入
    - 菜单
    - 事件驱动
- 代码可以在平台间移植，但是GLUT缺乏一些现代GUI的控件和功能
  - 无滚动条
  - 可用FLTK、SDL

http://www.opengl.org/resources/libraries/glut/

freeglut

- GLUT库已经很久没有更新
  - 可以和OpenGL 3.1一起使用
  - 有些功能不能使用，因为需要废弃的函数
- Freeglut是类似GLUT的开源扩展
  - 增加的功能
  - 上下文检查

GLEW

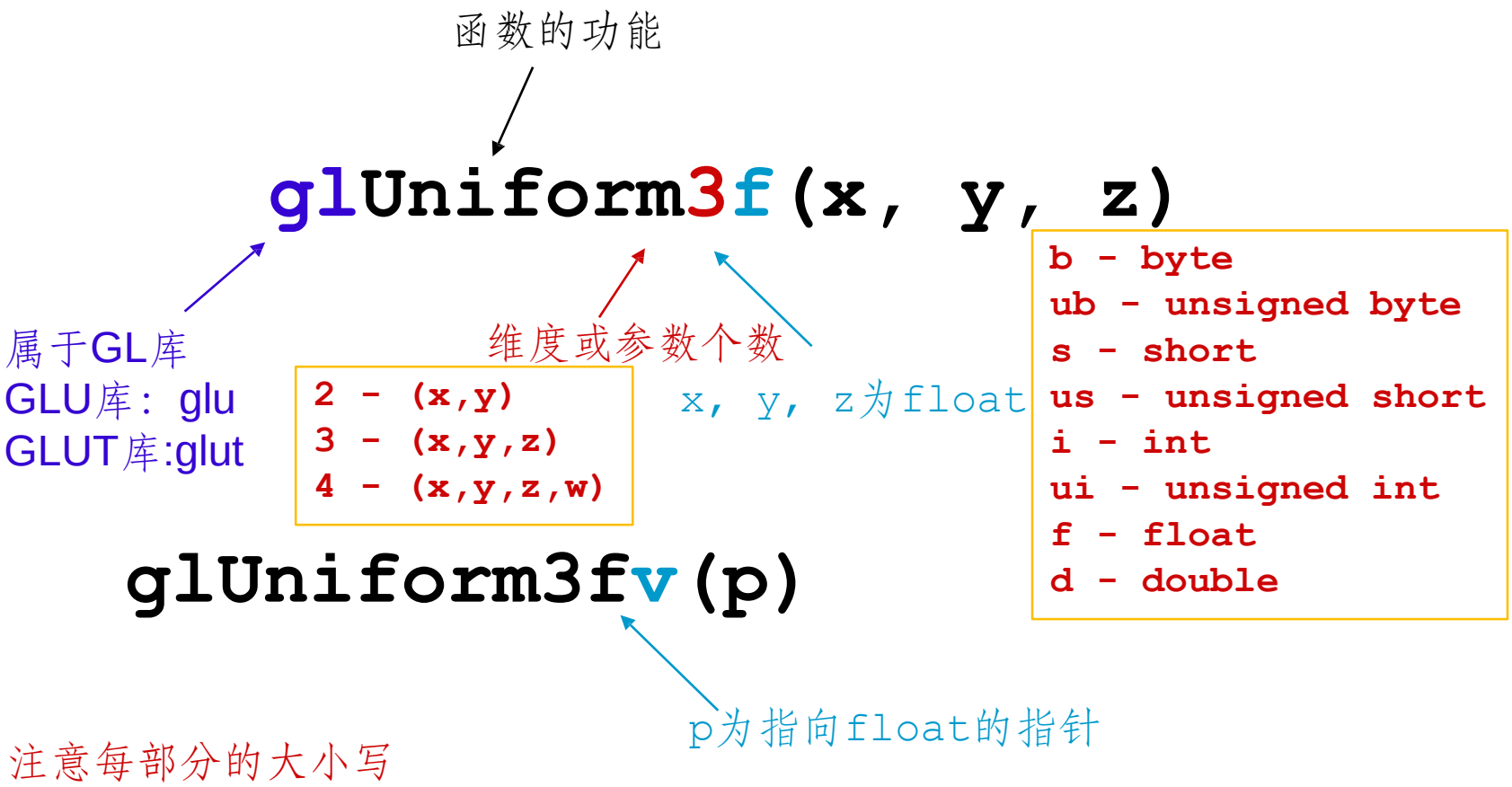
- OpenGL Extension Wrangler Library：跨 平台的开源OpenGL扩展加载库
- 使得调用特定系统支持的OpenGL扩展功 能更简单
- 对于windows代码来说，避免直接调用实 体入口
- 应用程序只需要包含glew.h头文件，并调 用glewInit()即可

OpenGL is cross-platform

- Same code works with little/no modifications
- Implementations:

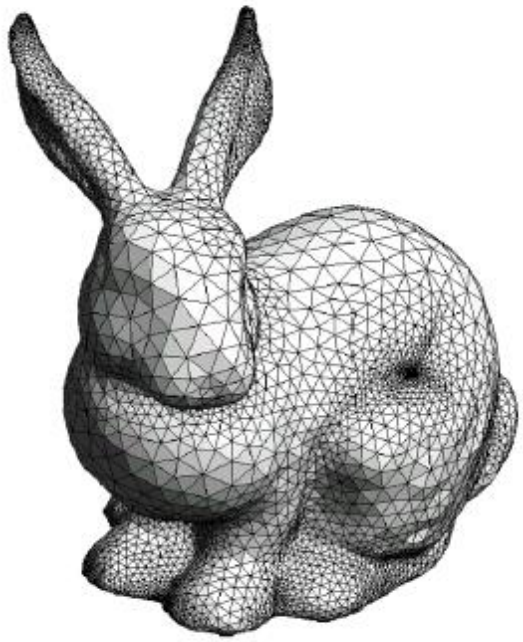
```
#if defined(WIN32) || defined(linux)
  #include <GL/gl.h>
  #include <GL/glu.h>
  #include <GL/glut.h>
#elif defined(__APPLE__)
  #include <OpenGL/gl.h>
  #include <OpenGL/glu.h>
  #include <GLUT/glut.h>
#endif
```

OpenGL函数名称的格式



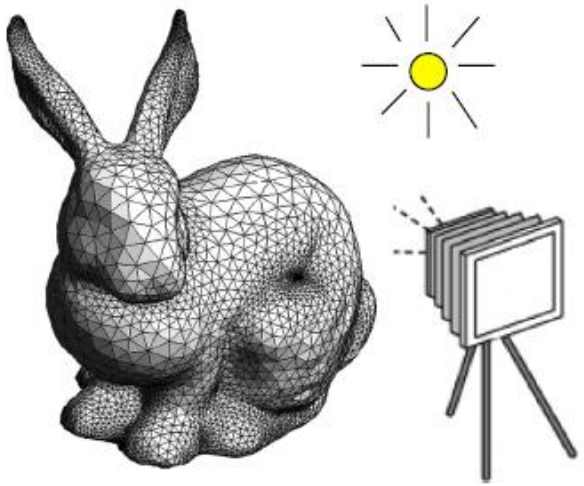
How does OpenGL work

- **From the programmer's point of view:**
  - Specify **geometric objects**
  - Describe **object properties**
    - Color
    - Material

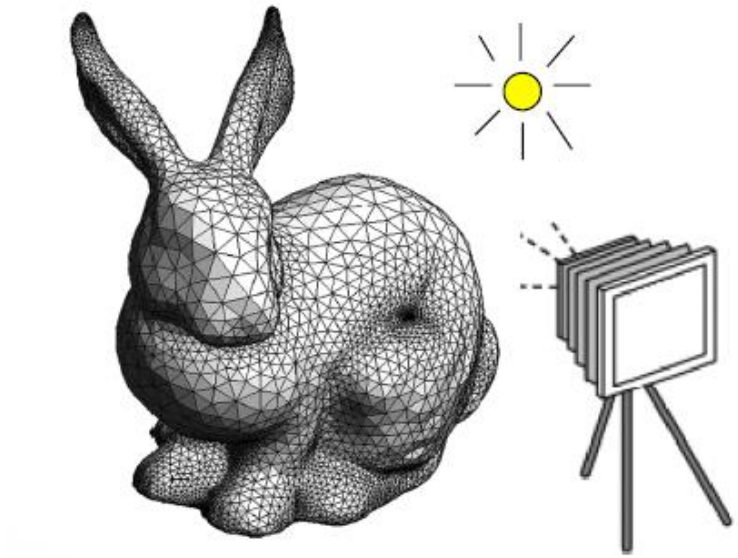


How does OpenGL work(continued)

- **Define how objects should be viewed!**
  - where is the camera?
  - what type of camera?
- **Specify light sources!**
  - where, what kind?
- **Move camera or objects around for animation**



The result



Scene

The result

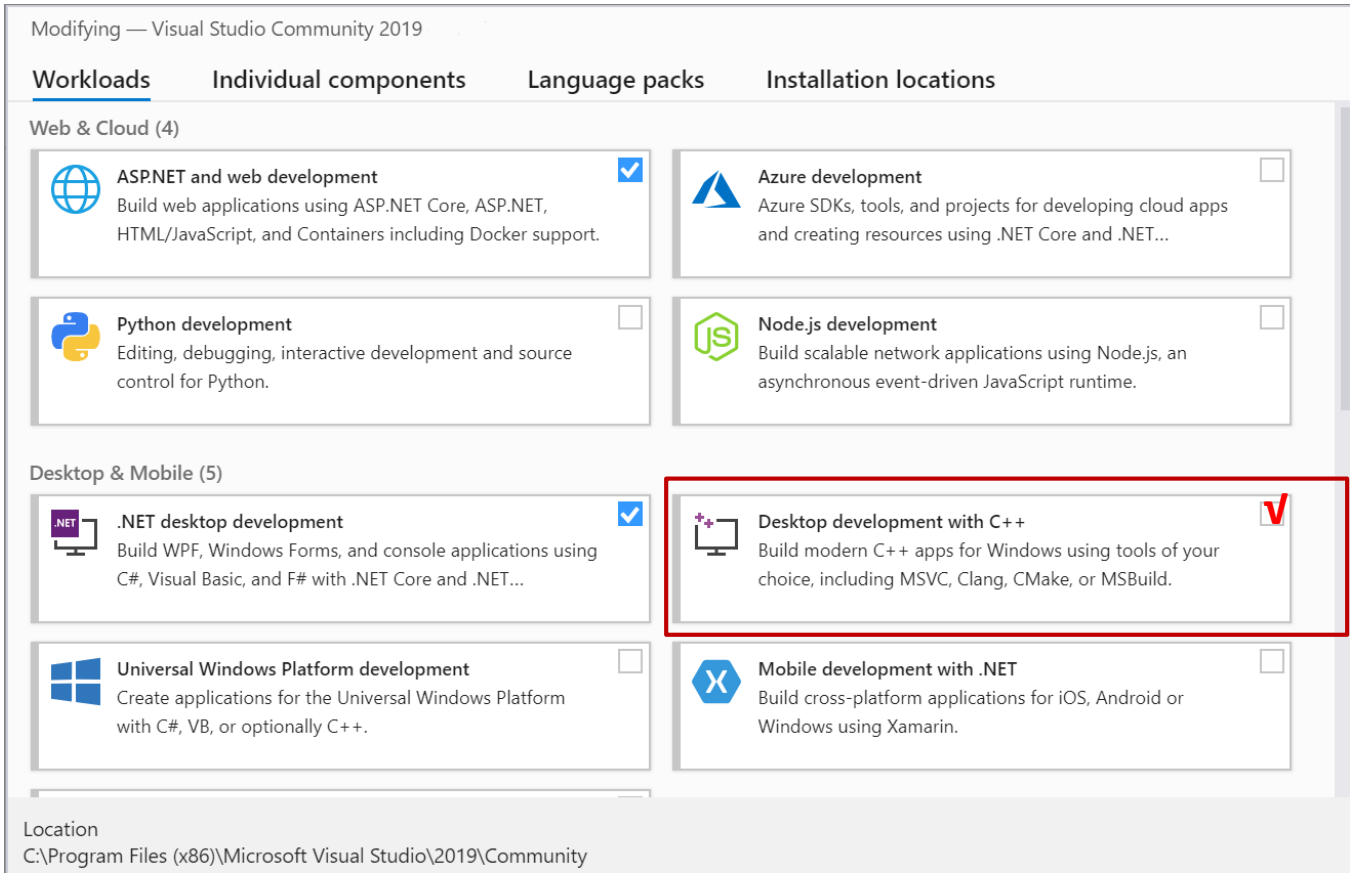
Set up programming environment

- Windows 11
- Microsoft Visual Studio Community 2019 (MSVS)
- C/C++

- Alternatives
  - Linux + Code::Blocks
  - MacOS + Xcode

Install Microsoft Visual Studio Community 2019

- Download it from the following line and follow the installation steps https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16

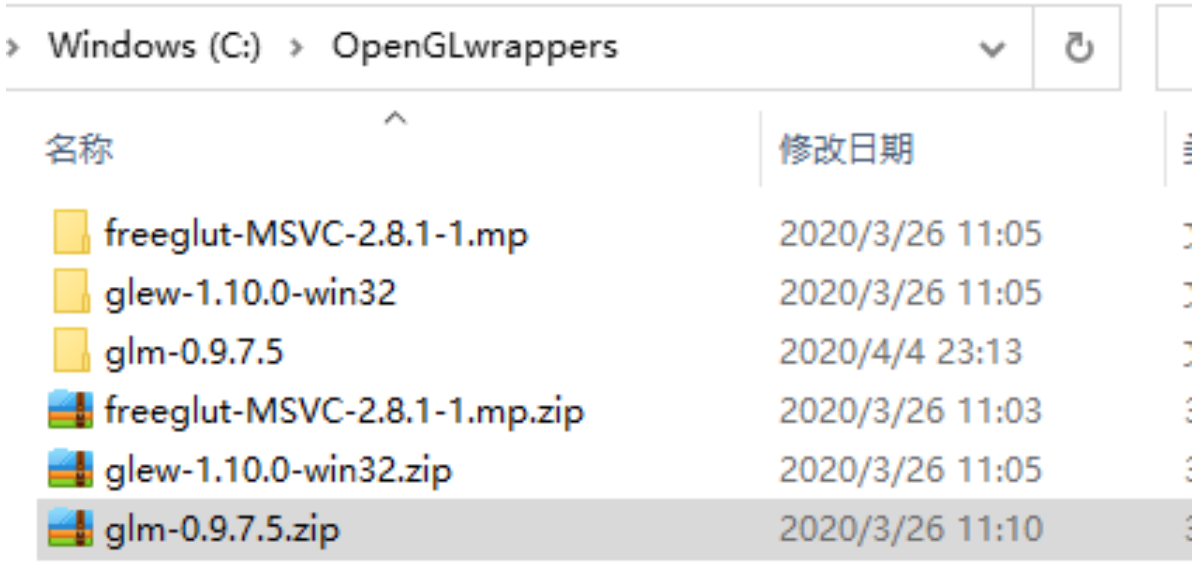


Install Helper Libraries

- Create a folder called OpenGLwrappers in the C: drive
  - So this folder is C:\OpenGLwrappers

Install Helper Libraries

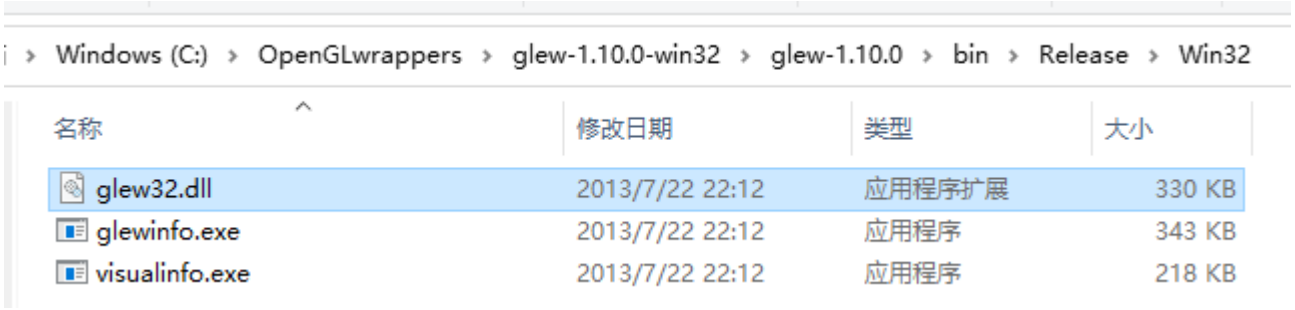
- Download and unzip the following files in \OpenGLwrappers
  - FreeGLUT: http://files.transmissionzero.co.uk/software/development/GLUT/older/freeglut-MSVC-2.8.1-1.mp.zip
  - GLEW: https://sourceforge.net/projects/glew/files/glew/1.10.0/glew-1.10.0-win32.zip/download
  - GLM: https://github.com/g-truc/glm/releases/download/0.9.7.5/glm-0.9.7.5.zip



Install Helper Libraries

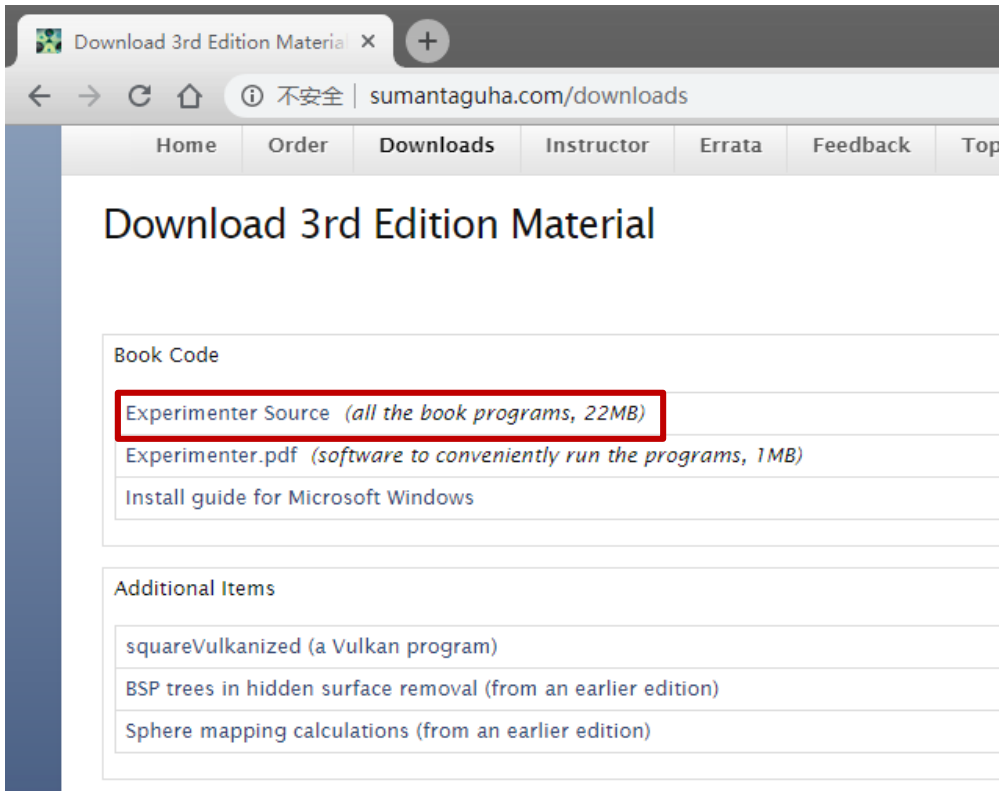
- Copy freeglut.dll from C:\OpenGLwrappers\freeglut-MSVC-2.8.1-1.mp\freeglut\bin to C:\Windows\SysWOW64.

- Copy glew32.dll from C:\OpenGLwrappers\glew-1.10.0-win32\glew-1.10.0\bin\Release\Win32 to C:\Windows\SysWOW64.



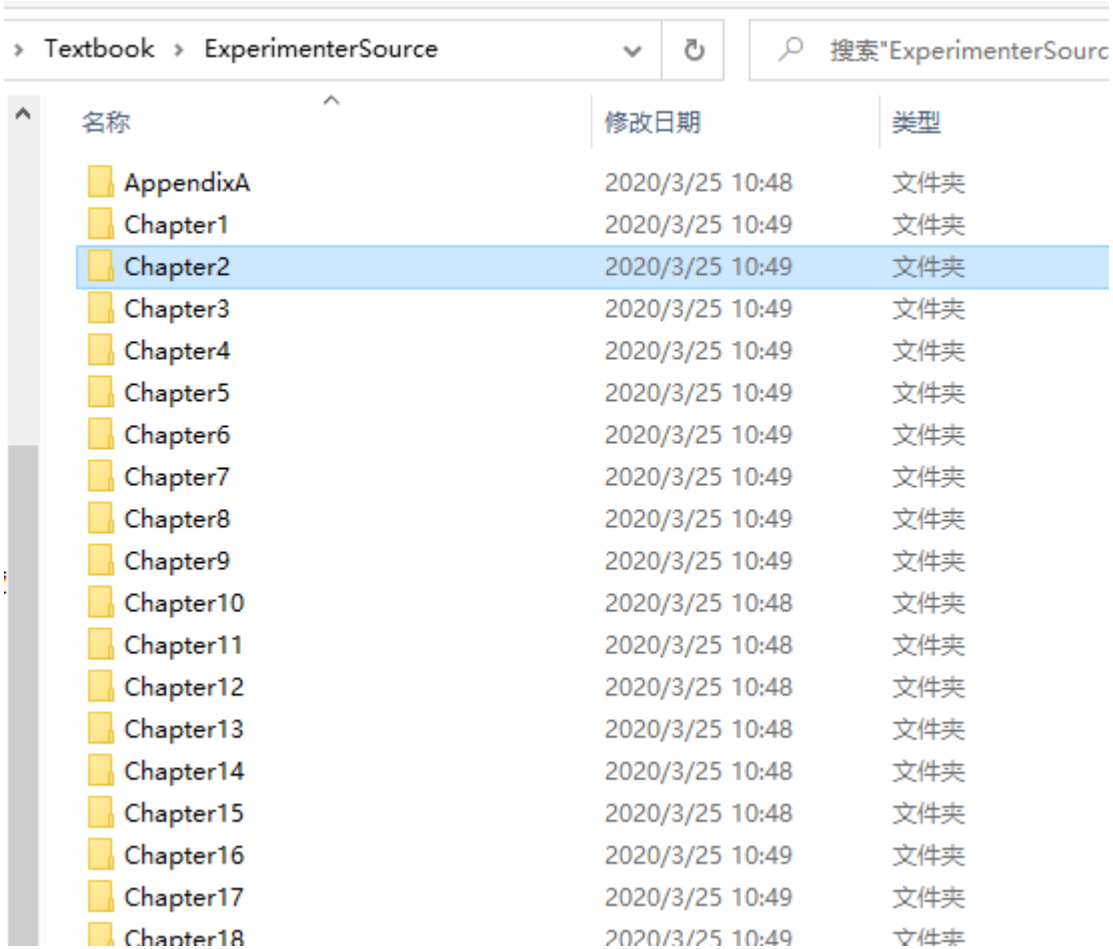
Test the Environment

- Download the ExperimenterSource.zip from the book's website http://www.sumantaguha.com/downloads



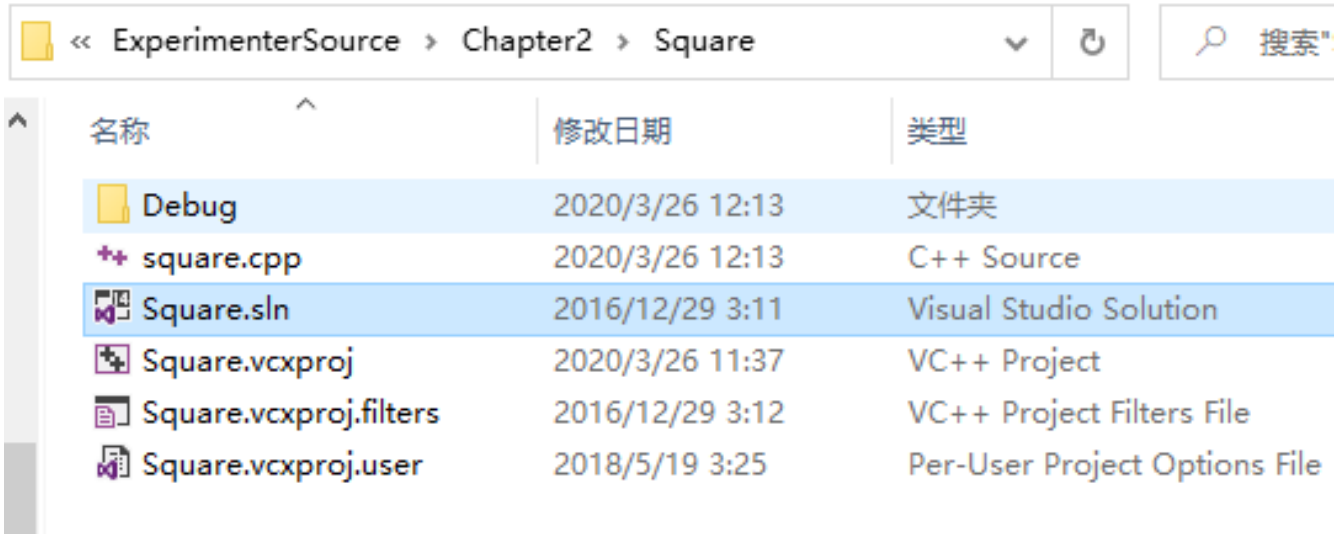
Test the Environment

- Unzip ExperimenterSource.zip to a folder



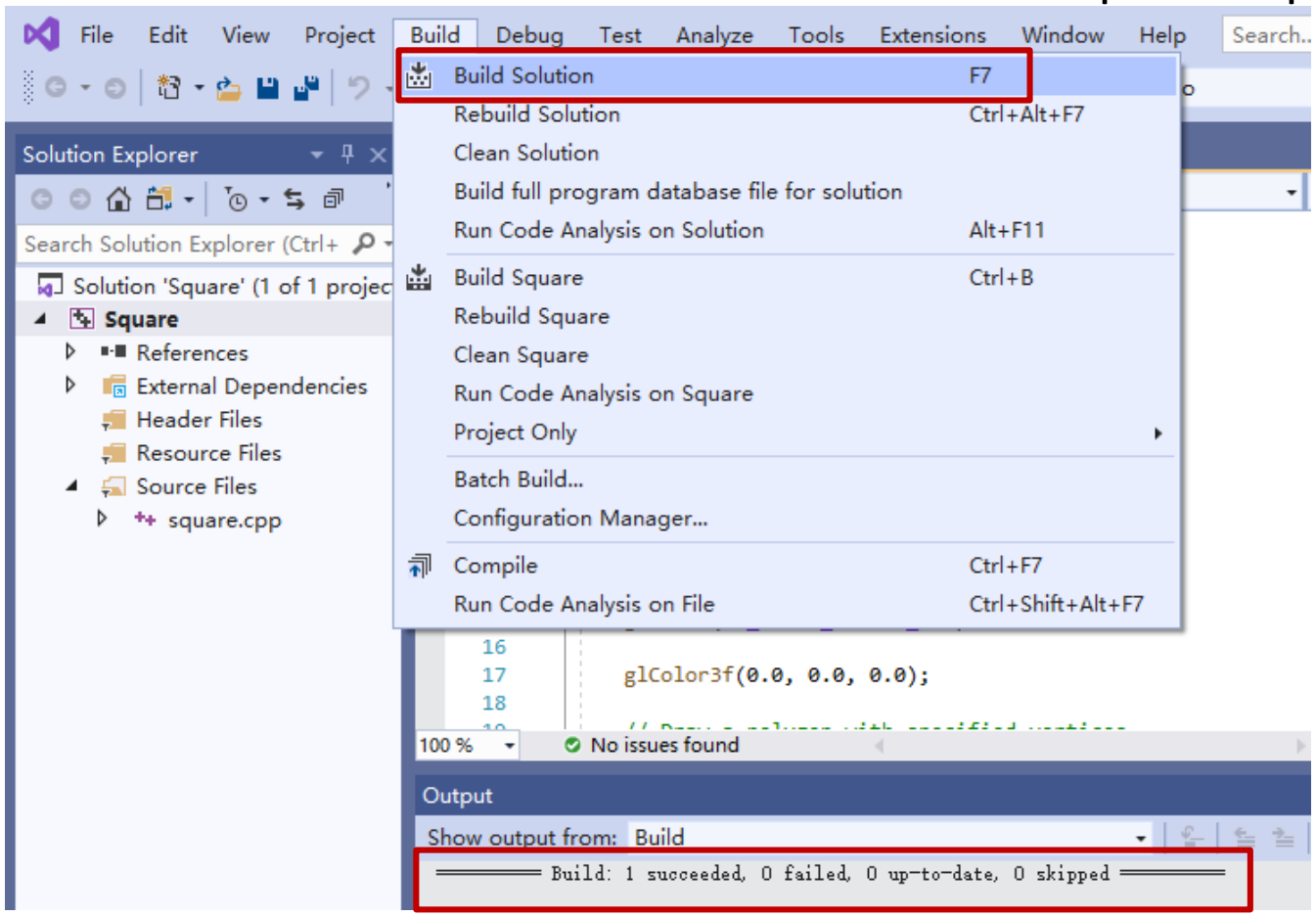
Test the Environment

- Unzip ExperimenterSource.zip to a folder
- Go to the subfolder ExperimenterSource\Chapter2
- Open the Square.sln file in Visual Studio



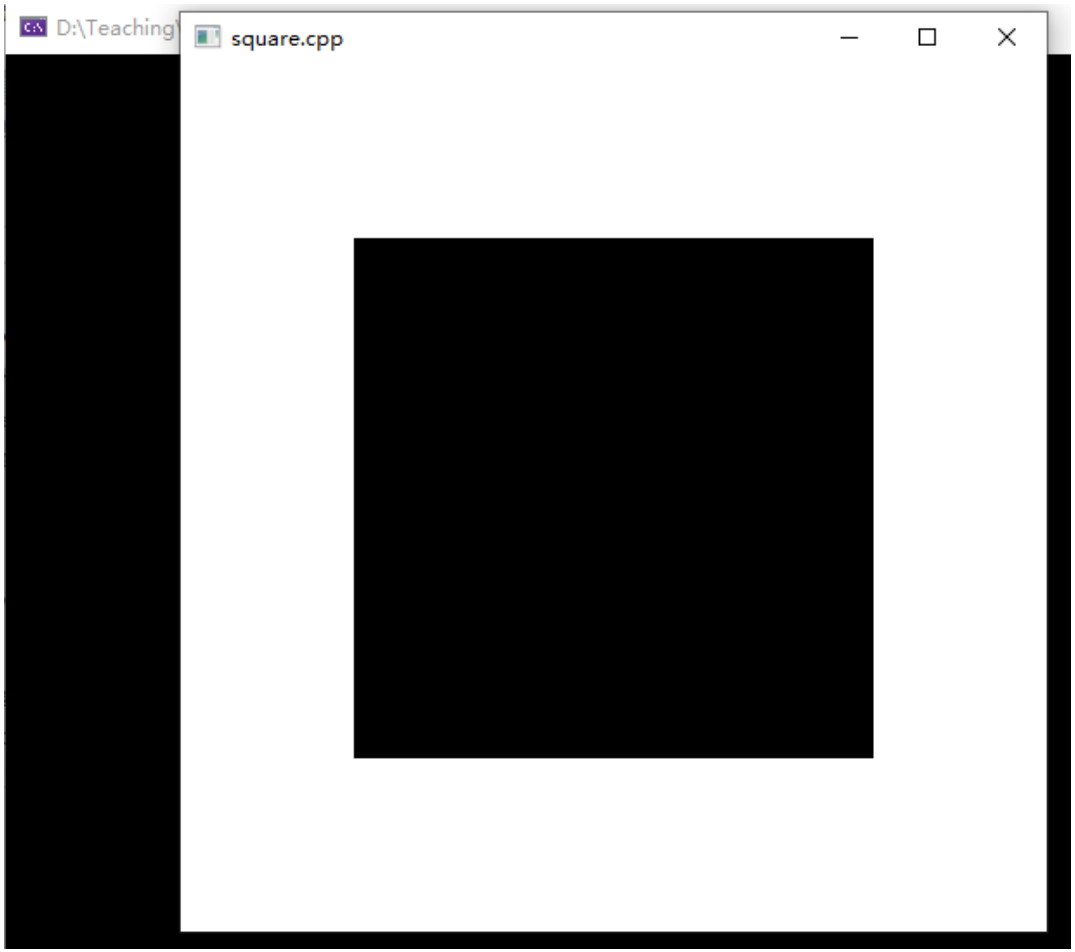
Test the Environment

- Open the Square.sln file in Visual Studio
- Click Build -> Build Solution in Visual Studio to compile the project



Test the Environment

- On the tool bar Click Debug -> Start Debugging to run



A Glimpse of OpenGL Program

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitContextVersion(4, 3);
    glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("square.cpp");
    glutDisplayFunc(drawScene);
    glutReshapeFunc(resize);
    glutKeyboardFunc(keyInput);
    glewExperimental = GL_TRUE;
    glewInit();
    setup();
    glutMainLoop();
}
```

A Glimpse of OpenGL Program

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitContextVersion(4, 3);
    glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);
```

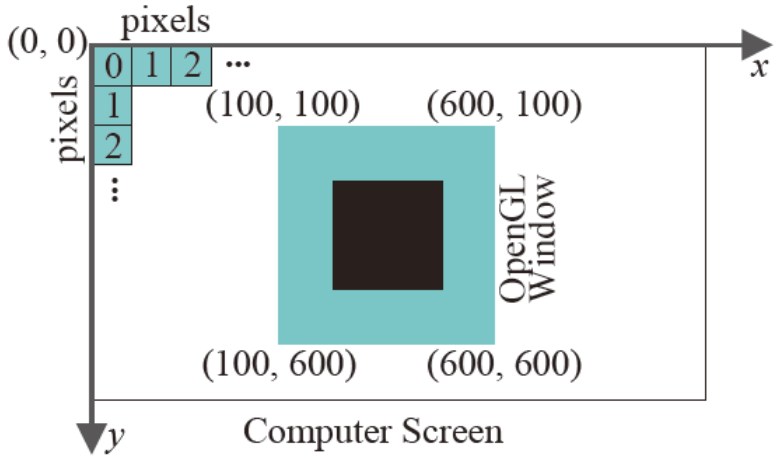
If the program compiles successfully but cannot run, replace glutInitContextVersion(4, 3) with glutInitContextVersion(3, 3) or even glutInitContextVersion(2, 1) instead.

A Glimpse of OpenGL Program

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitContextVersion(4, 3);
    glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("square.cpp");
    glutDisplayFunc(drawScene);
    glutReshapeFunc(resize);
    glutKeyboardFunc(keyInput);
    glewExperimental = GL_TRUE;
    glewInit();
    setup();
    glutMainLoop();
}
```

A Glimpse of OpenGL Program

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitContextVersion(4, 3);
    glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("square.cpp");
    glutDisplayFunc(drawScene);
    glutReshapeFunc(resize);
    glutKeyboardFunc(keyInput);
    glewExperimental = GL_TRUE;
    glewInit();
    setup();
    glutMainLoop();
}
```



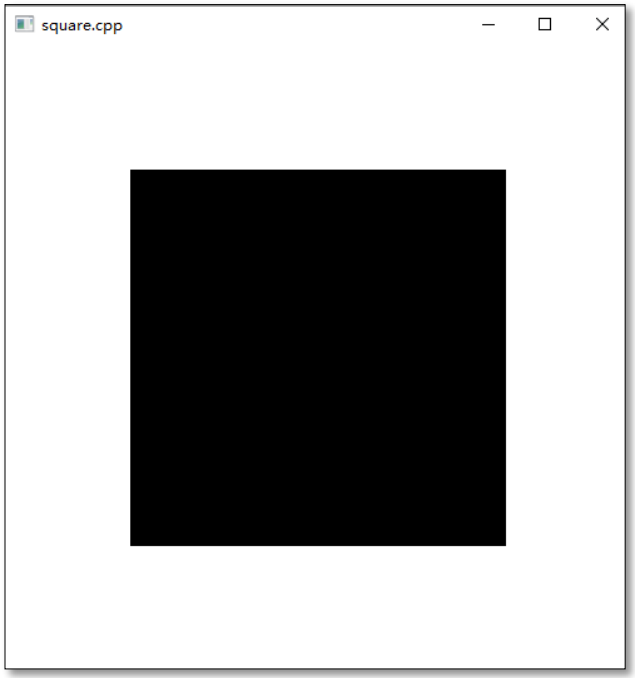
A Glimpse of OpenGL Program

```
// Drawing routine.
void drawScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(0.0, 0.0, 0.0);

    // Draw a polygon with specified vertices.
    glBegin(GL_POLYGON);
    glVertex3f(20.0, 20.0, 0.0);
    glVertex3f(80.0, 20.0, 0.0);
    glVertex3f(80.0, 80.0, 0.0);
    glVertex3f(20.0, 80.0, 0.0);
    glEnd();

    glFlush();
}
```



Tasks

1. Compile and run Square.sln in the directory \ExperimenterSource\Chapter2
2. Change the title of the window to “Hello from YOUR NAME”
3. Change the position of the window.
4. Change the size of the window.
5. Draw a red triangle or other interesting shapes.

- 把程序运行截屏上传到QQ群相册