

Tutorial 0: Warm-up & Assignment 1

TA: Wu Yiheng, Zhang Zhangrui, Zhang Siyuan

Agenda

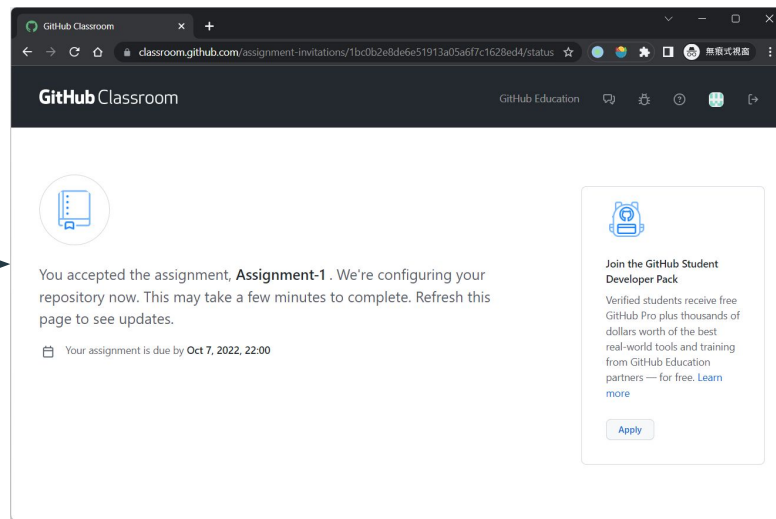
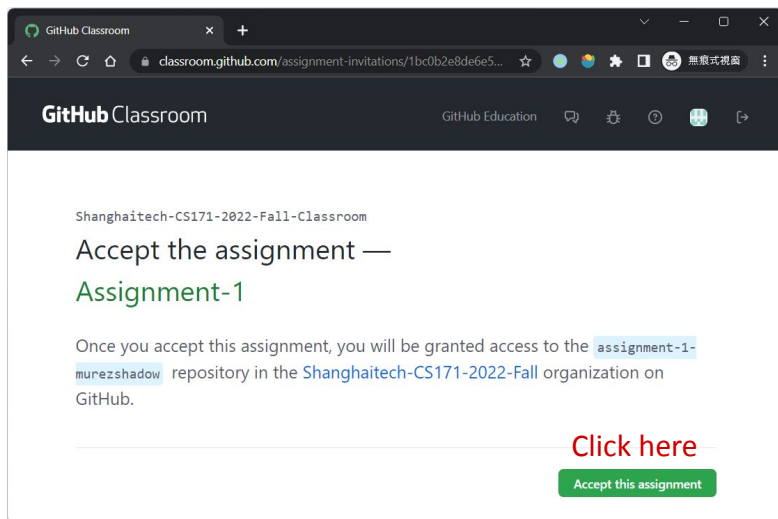
1. Assignment 1 I
2. Setup
3. OepnGL

Assignment, where?



Course Page: <https://faculty.sist.shanghaitech.edu.cn/faculty/liuxp/course/cs171.01/>

Assignment Page: <https://classroom.github.com/a/a8wFqhTv>

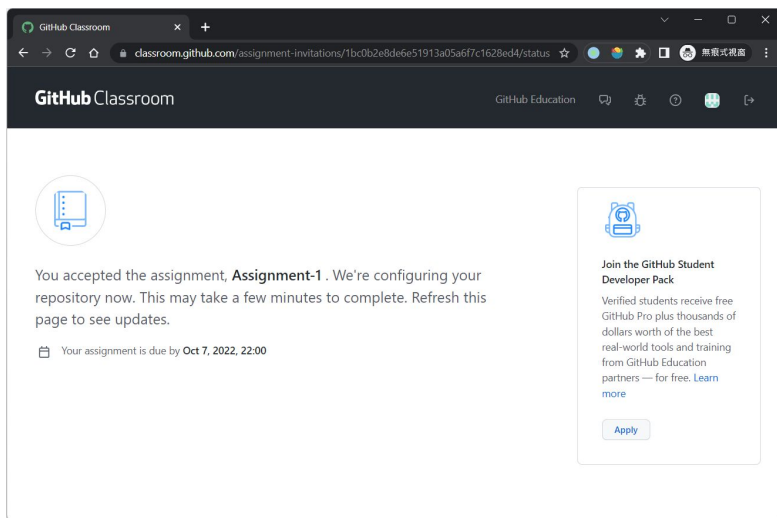


Assignment, where?

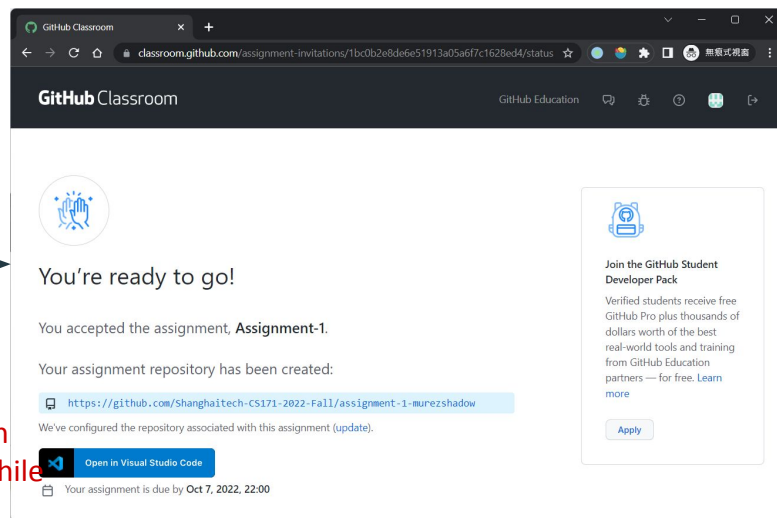


Course Page: <https://faculty.sist.shanghaitech.edu.cn/faculty/liuxp/course/cs171.01/>

Assignment Page: <https://classroom.github.com/a/a8wFqhTv>



refresh
after a while

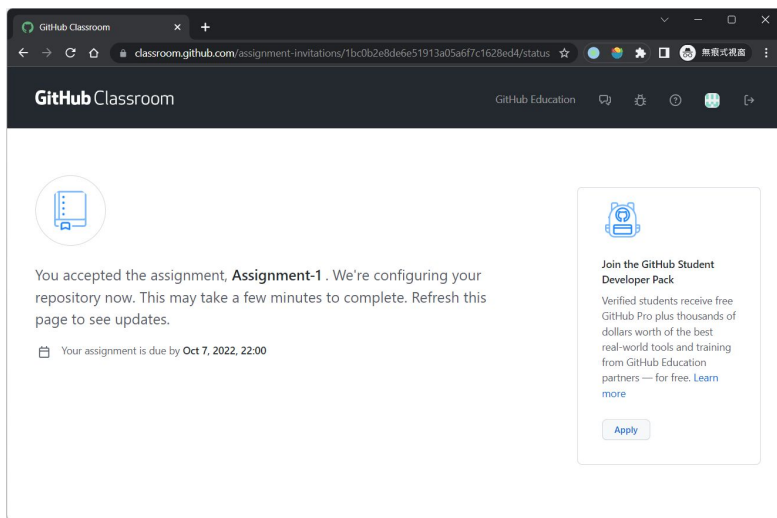


Assignment, where?

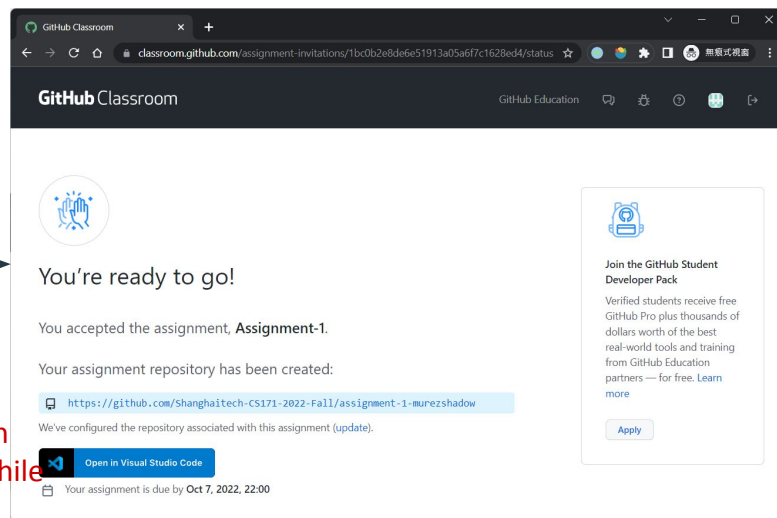


Course Page: <https://faculty.sist.shanghaitech.edu.cn/faculty/liuxp/course/cs171.01/>

Assignment Page: <https://classroom.github.com/a/a8wFqhTv>



refresh
after a while

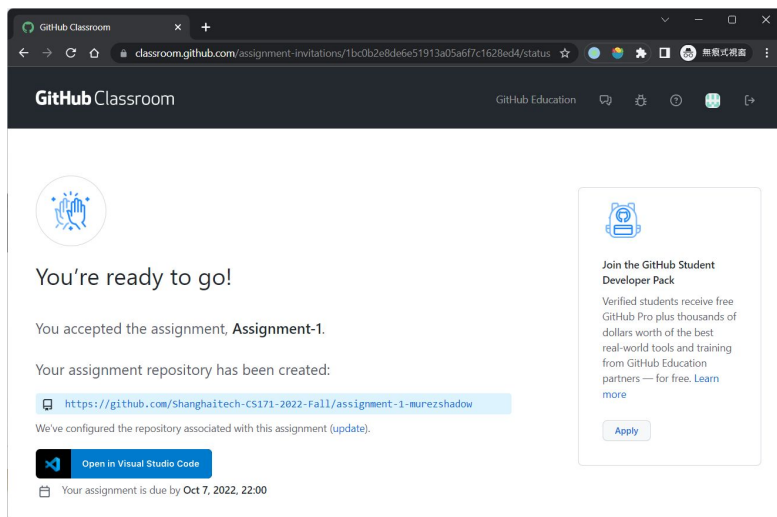


Assignment, where?

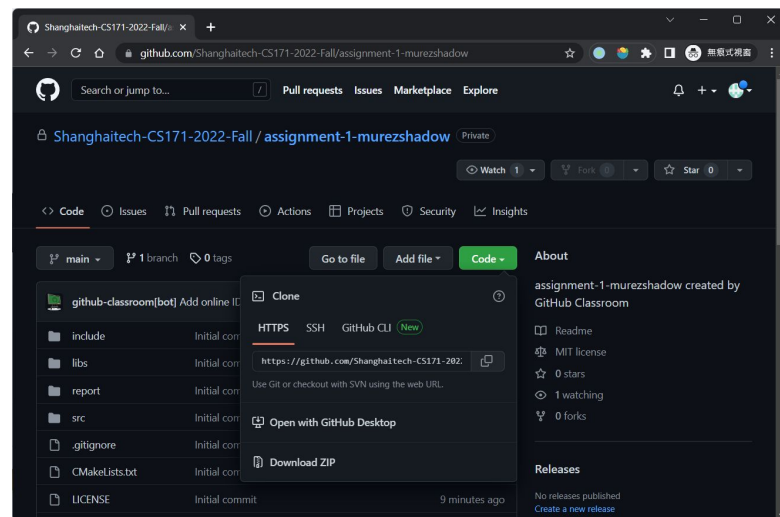


Course Page: <https://faculty.sist.shanghaitech.edu.cn/faculty/liuxp/course/cs171.01/>

Assignment Page: <https://classroom.github.com/a/a8wFqhTv>

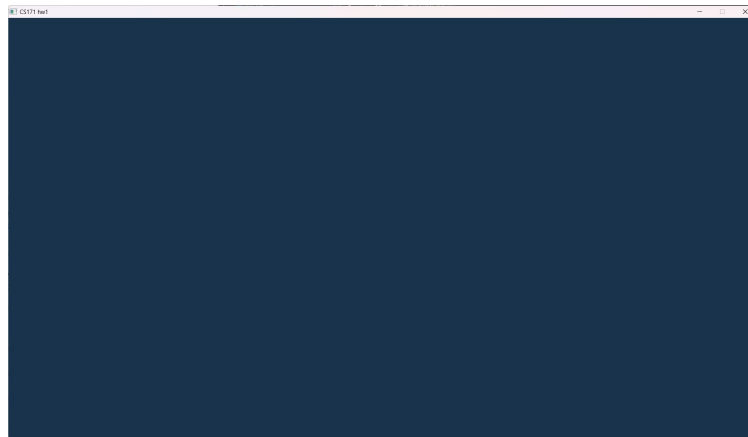
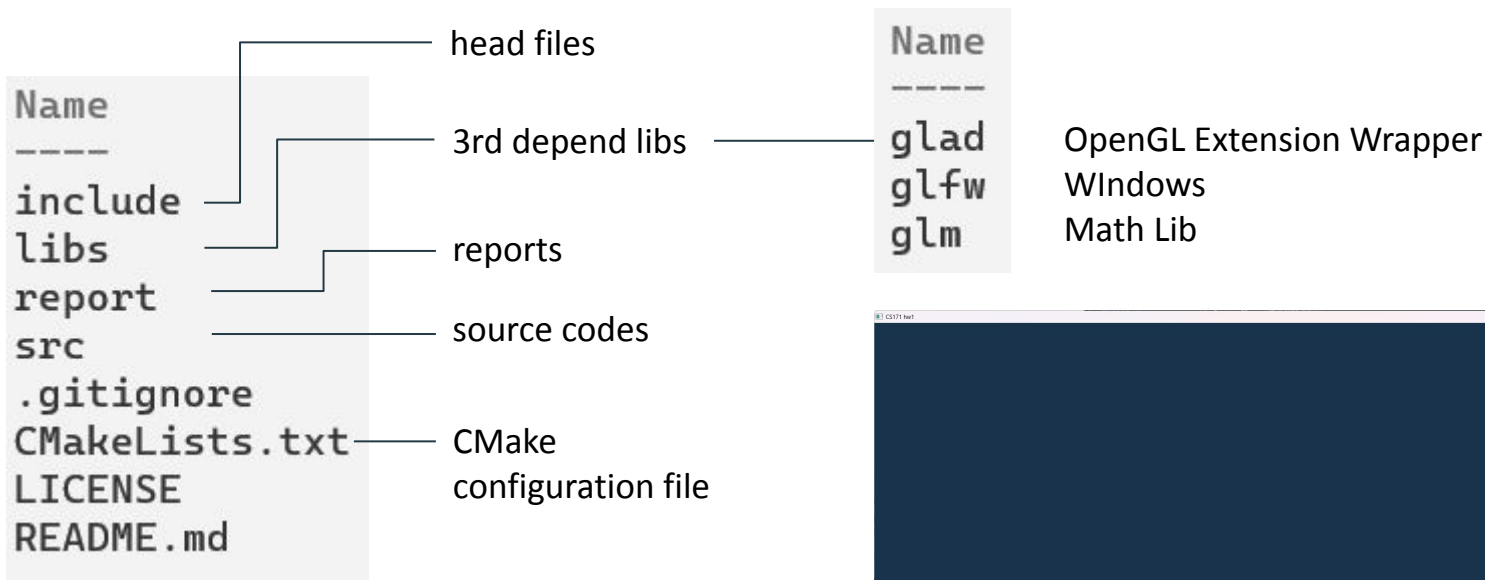


The screenshot shows the GitHub Classroom interface. At the top, the browser address bar displays the URL `classroom.github.com/assignment-invitations/1bc0b2e8de6e51913a05a67c1628ed4/status`. The page header includes the GitHub Classroom logo and the text "GitHub Education". The main content area features a blue circular icon with a graduation cap and the text "You're ready to go!". Below this, it states "You accepted the assignment, Assignment-1." and "Your assignment repository has been created:" followed by the repository URL `https://github.com/Shanghaitech-CS171-2022-Fall/assignment-1-murezhadow`. A button labeled "Open in Visual Studio Code" is visible. At the bottom, it says "Your assignment is due by Oct 7, 2022, 22:00". On the right side, there is a box titled "Join the GitHub Student Developer Pack" with a description of the benefits and an "Apply" button.



The screenshot shows the GitHub repository page for "Shanghaitech-CS171-2022-Fall / assignment-1-murezhadow". The repository is marked as "Private". The page includes navigation links for "Pull requests", "Issues", "Marketplace", and "Explore". The repository statistics show "Watch 1", "Fork 0", and "Star 0". The "Code" button is highlighted, and a dropdown menu is open, showing options to "Clone" (with sub-options for HTTPS, SSH, and GitHub CLI), "Open with GitHub Desktop", and "Download ZIP". The file list on the left includes "include", "libs", "report", "src", ".gitignore", "CMakeLists.txt", and "LICENSE". The right sidebar shows the repository's "About" section, including the license (MIT), stars (0), and releases (No releases published).

Assignment, what?



Assignment Requirements

detailed requirements is on the course page

You are supposed to rendering given **mesh** under **Phong Lighting** in the given OpenGL window, and you can navigate the scene with keyboard and mouse.

Bouns: Changing light type & Geometry Shader.

Pipeline Mode & Shader is a Must.

Take it easy and be creative.



Grading Rule

Warm Up Task will not be graded. And the rest tasks are surely taken into account of the final score.

You can earn **additional** score from **bonus** (maximum: **30%** of the entire score of this assignment.)

NO CHEATING.

Late submission of your assignment will be subject to score deduction based on the rule on the course webpage.

Assignment, when?

DDL: 2022 Oct. 7 22:00 (UTC+8)



You accepted the assignment, **Assignment-1** . We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates.

📅 Your assignment is due by **Oct 7, 2022, 22:00**

Learning Materials

Useful Websites

<https://learnopengl.com/>

<http://www.opengl-tutorial.org/>

<https://www.khronos.org/opengl/wiki/Tutorials>

Book

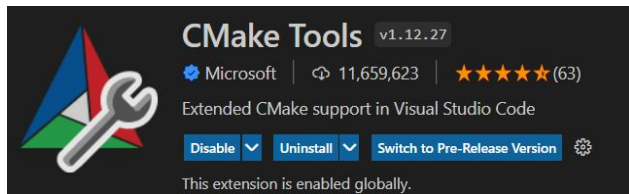
[Physically Based Rendering - From Theory to Implementation](#)

Setup

CMake Project

1. Windows: VS Code / Visual Studio / Clion
2. Linux: VS Code / Clion / Vim / Emacs / manual ...
3. MacOS: VS Code / X Code ? (None of TAs owns a Mac QAQ)

And a demo under Windows for VS Code / Clion / Visual Studio



OpenGL Setup (Window & Extension Wrapper)

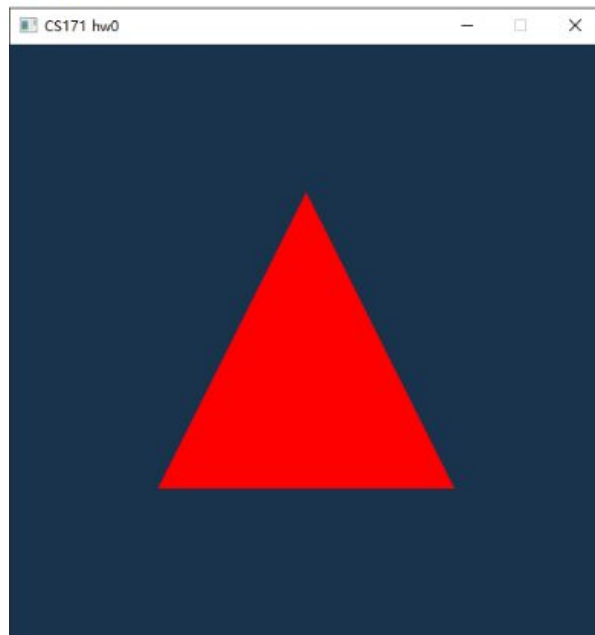
1. Window (窗口管理库)
 - a. 古老产品: glut/freeglut
 - b. 替代品: glfw
2. OpenGL Extension Wrapper (函数加载)
 - a. 古老产品: glew
 - b. 替代品: glad

常见环境配置

1. glfw + glew (因为我们希望大家尽早接触可编程管线模式我们放弃此模板配置 (;'⌣`))
2. glfw + glad (推荐配置, 我们提供的配置, 针对现代OpenGL o(╯▽╰)d)
3. freeglut + glew

OpenGL Coding, Immediate Mode

```
glBegin(GL_TRIANGLES);  
// vertex color (red)  
glColor3f(1.0f, 0.0f, 0.0f);  
// positions  
glVertex2f(-0.5f, -0.5f);  
glVertex2f(0.5f, -0.5f);  
glVertex2f(0.0f, 0.5f);  
glEnd();
```



OpenGL Coding, Immediate v.s. Core

Immediate mode (立即模式):

- 早期的OpenGL 使用
- 固定渲染管线
- 容易使用和理解
- 绘制图形很方便(glBegin & glEnd)
- 大多数功能都被库隐藏起来, 不够灵活, 效率低

Core mode (核心模式):

- 现代OpenGL
- 可编程渲染管线
- 更多的灵活性, 更高的效率
- 更深入的理解 图形编程
- 虽然上手更困难, 但这份努力是值得的

OpenGL is?

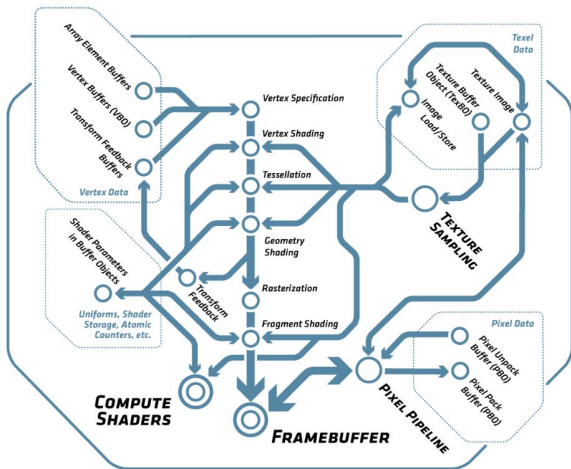
OpenGL是一个状态机

记忆功能:能够记住自己当前的状态。

接收输入:根据输入的内容和自己的状态, 修改自己的状态, 并且可以得到输出。

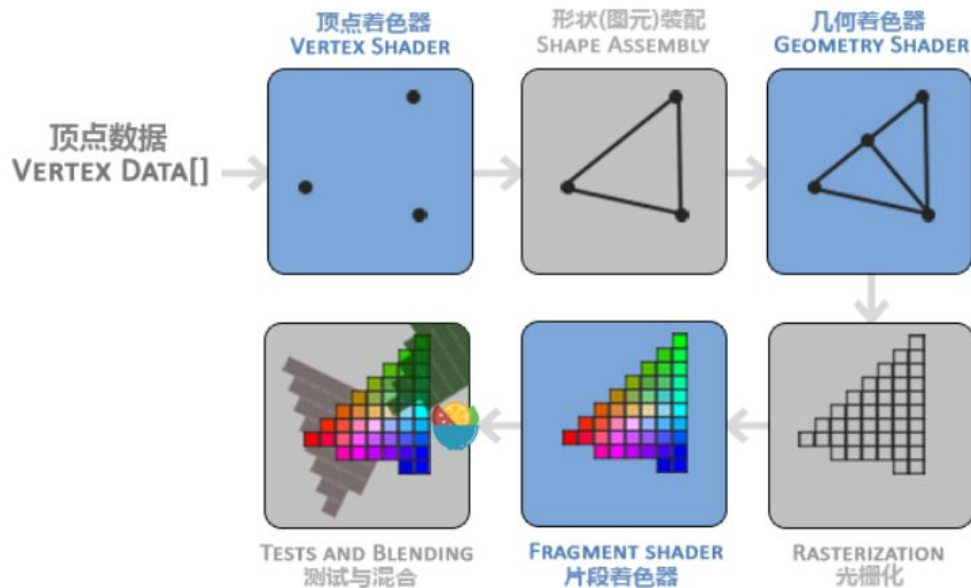
如何操作 OpenGL?

通过绑定对象和选项



```
// 创建对象
unsigned int objectId = 0;
glGenObject(1, &objectId);
// 绑定对象至上下文
glBindObject(GL_WINDOW_TARGET, objectId);
// 设置当前绑定到 GL_WINDOW_TARGET 的对象的一些选项
glSetObjectOption(GL_WINDOW_TARGET, GL_OPTION_WINDOW_WIDTH, 800);
glSetObjectOption(GL_WINDOW_TARGET, GL_OPTION_WINDOW_HEIGHT, 600);
// 将上下文对象设回默认
glBindObject(GL_WINDOW_TARGET, 0);
```


Shader is?



```
#version 330 core
layout (location = 0) in vec3 aPos;
layout (location = 1) in vec3 aColor;
out vec3 ourColor;
void main() {
    gl_Position = vec4(aPos.x, aPos.y, aPos.z, 1.0);
    ourColor = aColor;
}
```



```
#version 330 core
out vec4 FragColor;
in vec3 ourColor;
uniform float getColorByTime;
void main() {
    ourColor = ourColor * getColorByTime;
    FragColor = vec4(ourColor[0], ourColor[1], ourColor[2], 1.0f);
}
```

VBO (Vertex Buffer Object, 顶点缓冲对象)

```
// VBO id
unsigned int VBO;
// 生成 VBO
glGenBuffers(1, &VBO);
// 绑定一个 VBO 对象 (why? 状态机)
glBindBuffer(GL_ARRAY_BUFFER, VBO);
// 将 vertices 数据复制到 VBO
glBufferData(GL_ARRAY_BUFFER,
             sizeof(vertices),
             vertices,
             GL_STATIC_DRAW);
glVertexAttribPointer(0,
                     3,
                     GL_FLOAT,
                     GL_FALSE,
                     6 * sizeof(float),
                     (void *) 0);
```

