# 一、GLFW+GLEW+SOIL+GLM

**窗口及初始化设置**

glfwInit();// Initialize GLFW

//\*\*\*设置GLFW，第一变量设置选项类型，第二变量设置数值

glfwWindowHint(GLFW\_CONTEXT\_VERSION\_MAJOR, 3);//设定OpenGL使用版本3.3

glfwWindowHint(GLFW\_CONTEXT\_VERSION\_MINOR, 3);

glfwWindowHint(GLFW\_OPENGL\_PROFILE, GLFW\_OPENGL\_CORE\_PROFILE);//设定使用core profile方式编程

glfwWindowHint(GLFW\_RESIZABLE, GL\_FALSE);//设定不能改变窗口大小

//\*\*\*创建窗口对象，第一二变量设置窗口尺寸，第三窗口名称，第四设置哪个显示器使用全屏模式，或NULL为窗口模式，第五设置哪个窗口共享资源，或NULL不共享

GLFWwindow \*window= glfwCreateWindow(WIDTH,HEIGHT, “NAME”,NULL,NULL);//建立失败的话，window值为NULL

glfwMakeContextCurrent(window);//设置为当前使用context

//\*\*\*设置事件回调函数，第一窗口，第二回调函数名（自定义）

glfwSetKeyCallback(window,key\_callback);//键盘按键

glfwSetCursorPosCallback(window,mouse\_callback);//鼠标移动

glfwSetScrollCallback(window,scroll\_callback);//鼠标滚动

//\*\*\*设置输入模式，第一窗口，第二模式，第三设置值

glfwSetInputMode(window,GLFW\_CURSOR,GLFW\_CURSOR\_DISABLED);//设置隐藏光标和窗口封闭住光标

glewExperimental= GL\_TRUE;//使用glew的高阶功能

glewInit();//初始化glew，成功返回值GLEW\_OK

//\*\*\*设置视口

glfwGetFramebufferSize(window,&width,&height);//获取窗口的帧缓存大小

glViewport(0,0,width,height);//设置视口的原点坐标和尺寸（在创建的窗口屏幕坐标上）

//\*\*\*设置OpenGL的功能

glEnable(GL\_DEPTH\_TEST);//启动深度测试

glPolygonMode(GL\_FRONT\_AND\_BACK,GL\_LINE);//设置用线条方式绘制多边形，默认填充方式绘制多边形

**顶点缓存和顶点信息读取**

//\*\*\*设置顶点缓存，顶点数组，单元缓存对象

GLuint VAO,VBO, EBO;//创建存储缓存ID的变量

glGenVertexArrays(1,&VAO);//生成缓存对象，第一生成对象的数量，第二生成对象的ID数组的首地址传送到哪

glGenBuffers(1,&VBO);

glGenBuffers(1,&EBO);

//\*\*\*绑定顶点数组对象，该对象存储所有的和顶点属性有关的设置和缓存设置（VBO与EBO）

glBindVertexArray(VAO);

glBindBuffer(GL\_ARRAY\_BUFFER,VBO);//绑定缓存对象为当前使用的对象

glBufferData(GL\_ARRAY\_BUFFER,sizeof(vertices),vertices,GL\_STATIC\_DRAW);//将储存顶点信息的数组，传输给缓存对象，第一缓存类型，第二数组大小，第三数组地址，第四绘制方式

glBindBuffer(GL\_ELEMENT\_ARRAY\_BUFFER,EBO);

glBufferData(GL\_ELEMENT\_ARRAY\_BUFFER,sizeof(indices),indices,GL\_STATIC\_DRAW);//将储存顶点索引的数组传输给缓存对象

glVertexAttribPointer(0,3,GL\_FLOAT,GL\_FALSE,5\*sizeof(GLfloat),(GLvoid\*)0);//将VBO中数据按照设定格式读入VAO，第一是属性位置（attribute location），第二是属性所含数值个数，第三数值类型，第四是否归一化数值，第五步长（stride），第六数值起始位置

glEnableVertexAttribArray(0); //激活顶点属性，默认不激活，"0"就是顶点着色器中定义的属性位置

glVertexAttribPointer(1,3,GL\_FLOAT,GL\_FALSE,8\*sizeof(float),(GLvoid\*)(3\*sizeof(float)));

glEnableVertexAttribArray(1);

glVertexAttribPointer(2,2,GL\_FLOAT,GL\_FALSE,5\*sizeof(float),(GLvoid\*)(3\*sizeof(float)));

glEnableVertexAttribArray(2);

glBindVertexArray(0);//反绑定 VAO

glBindBuffer(GL\_ARRAY\_BUFFER,0);//反绑定VBO

glBindBuffer(GL\_ELEMENT\_ARRAY\_BUFFER,0); //反绑定EBO一定要在VAO之后

**纹理读取与设置**

# 二、GLM其他常用语句和函数功能

# 三、GLSL常用语句和函数功能