

OpenGL不同头文件及库的说明

[1] OpenGL Utilities

glu是实用库，包含有43个函数，函数名的前缀为glu。Glu 为了减轻繁重的编程工作，封装了OpenGL函数，Glu函数通过调用核心库的函数，为开发者提供相对简单的用法，实现一些较为复杂的操作。

[2] OpenGL Utility Toolkit

glut是实用工具库，基本上是由于做窗口界面的，并且是跨平台（所以有时你喜欢做简单的demo的话，可以光用glut就ok了）

[3] glew

glew是一个跨平台的C++扩展库，基于OpenGL图形接口。使用OpenGL的朋友都知道，window目前只支持OpenGL1.1的函数，但 OpenGL现在都发展到2.0以上了，要使用这些OpenGL的高级特性，就必须下载最新的扩展，另外，不同的显卡公司，也会发布一些只有自家显卡才支持的扩展函数，你要想用这数函数，不得不去寻找最新的glext.h,有了GLEW扩展库，你就再也不用为找不到函数的接口而烦恼，因为GLEW能自动识别你的平台所支持的全部OpenGL高级扩展函数。也就是说，只要包含一个glew.h头文件，你就能使用gl,glu,glext,wgl,glx的全部函数。GLEW支持目前流行的各种操作系统（including Windows, Linux, Mac OS X, FreeBSD, Irix, and Solaris）。

[4] glx

glx:OpenGL extension for X.

对于X窗口系统，它所使用的的OpenGL扩展（GLX）是作为OpenGL的一个附件提供的，所有的GLX函数都使用前缀glX。

glx是OpenGL Extension to the X Window System的缩写。它作为x的扩展，是x协议和X server的一部分，已经包含在X server的代码中了。GLX提供了x window system使用的OpenGL接口，允许通过x调用OpenGL库。OpenGL 在使用时，需要与一个实际的窗口系统关联起来。在不同平台上有不同的机制以关联窗口系统，在Windows上是WGL，在Linux上是GLX，在Apple OS上是AGL等。

[5] 头文件

<GL/gl.h>：OpenGL所使用的函数和常量声明。

<GL/glu.h>：OpenGL Utility Toolkit所使用的函数和常量声明。GLU库属于OpenGL标准的一部分。（以下各种库则不属于）

<GL/glut.h>：OpenGL Utility Toolkit所使用的函数和常量声明。目前许多OpenGL教程使用这个库来编写演示程序。一些编译系统可能不直接提供这个库（例如VC系列），需要单独下载安装。这个头文件自动包含了**<GL/gl.h>**和**<GL/glu.h>**，编程时不必再次包含它们。

`<GL/glaux.h>`: GLAUX (OpenGL辅助库) 所使用的函数和常量声明。这个库的功能大致与`<GL/glut.h>`类似, 这个库提供了创建窗口, 处理键盘和鼠标事件, 设置调色板等OpenGL本身不提供, 但在编写OpenGL程序时又经常用到的功能。目前这个库已经过时, 只有比较少的编译环境中提供, 例如VC系列。在VC系列编译器中, 使用这个头文件之前必须使用`#include <windows.h>`或者具有类似功能的头文件。

`<GL/glext.h>`: 扩展头文件。因为微软公司对OpenGL的支持不太积极, VC系列编译器虽然有`<GL/gl.h>`这个头文件, 但是里面只有OpenGL 1.1版本中所规定的内容, 而没有OpenGL 1.2及其以后版本。对当前的计算机配置而言, 几乎都支持OpenGL 1.4版本, 更高的则到1.5, 2.0, 2.1, 而VC无法直接使用这些功能。为了解决这一问题, 就有了`<GL/glext.h>`头文件。这个头文件提供了高版本OpenGL所需要的各种常数声明以及函数指针声明。

`<GL/wglext.h>`: 扩展头文件。与`<GL/glext.h>`类似, 但这个头文件中只提供适用于Windows系统的各种OpenGL扩展所使用的函数和常量, 不适用于其它操作系统。

`"glee.h"`: GLEE开源库的头文件。它的出现是因为`<GL/glext.h>`虽然可以使用高版本的OpenGL函数, 但是使用的形式不太方便。GLEE库则让高版本的OpenGL函数与其它OpenGL函数在使用上同样方便。需要注意的是, 这个头文件与`<GL/gl.h>`是冲突的, 在包含`"glee.h"`之前, 不应该包含`<GL/gl.h>`。

```
#include <GL/glut.h>
```

```
#include "glee.h" // 错误, 因为glut.h中含有gl.h, 它与  
glee.h冲突
```

```
// 但是如果把两个include顺序交换, 则正确
```