**模板测试**

模板测试把像素存储在模板缓冲区的点与一个参考值进行比较（glStencilFunc），根据测试结果对模板缓冲区的值进行响应修改（glStencilOp）。为了使用模板测试，我们必须首先使用平台特定的OpenGL设置过程请求一个模板缓冲区，当我们使用GLUT时，我们在初始化显示模式时请求一个模板缓冲区，例如下面的代码：

glutInitDisplayMode(GLUT\_RGB | GLUT\_DOUBLE | GLUT\_STENCIL);

我们可以采用glEnable(GL\_STENCIL\_TEST)启动模板测试

打开了模板测试之后，绘图就只在那些通过了模板测试的位置进行，我们可以使用下面的函数进行模板测试：

Void glStencilFunc(GLenum func, Glint ref, GLuint mask)

模板函数的func可以取下面的任意一个值：GL\_NEVER、GL\_ALWAYS、GL\_LESS、GL\_LEUUAL、GL\_EQUAL、GL\_GEQUAL和GL\_NOTEQUAL，这些值告诉OpenGL如何把已经存储在模板缓冲分区中的值与ref参数所指定的值进行比较，另外我们可以指定一个掩码，在比较之前，将这个掩码与参考值和摹本缓冲区中的值进行位AND操作

glStencilMask允许我们给末班之设置一个位遮罩（bitmask），它与模板值进行与（and）运算决定缓冲是否可写。默认设置的位遮罩都是1，这样就不会影响输出，但是如果我们设置为0x00，所有写入模板缓冲最后都是0，这和深度缓冲glDepthMash(GL\_FALSE)一样。

模板方程：glStencilFunc和glStencilOp

glStencilFunc(GLenum func，GLint ref，GLuint mask)

func：设置模板测试方程，这个测试方程应用到已经存储的末班之和glStencilFunc的ref值上，可以选项是：GL\_NEVER、GL\_LEQUAL、GL\_GREATER等

ref：指定模板测试的引用值。模板缓冲的内容会与这个值对比

mask：指定一个遮罩，在模板测试对比引用值和储存的模板值前，对它们进行与（and）操作。初始值设置为1

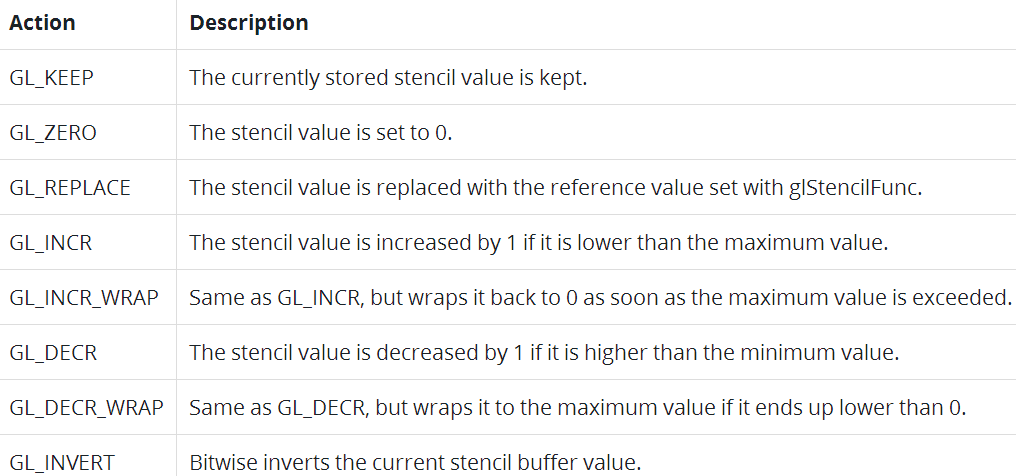
glStencilFunc(GL\_LESS, 3, mask);  
这段代码设置模板测试的条件为：“小于3则通过”。glStencilFunc的前两个参数意义与glAlphaFunc的两个参数类似，第三个参数的意义为：如果进行比较，则只比较mask中二进制为1的位。例如，某个像素模板值为5（二进制101），而mask的二进制值为00000011，因为只比较最后两位，5的最后两位为01，其实是小于3的，因此会通过测试。

glStencilOp(GLenum sfail, GLenum dpfail, GLenum dppass)

sfail：如果模板测试失败将采取的动作

dpfail：如果模板测试通过，但是深度测试失败时采取的动作

dppass：如果深度测试和模板测试都通过，将采取的动作。



glSencilOp函数默认设置为（GL\_KEEP，GL\_KEEP，GL\_KEEP），所以任何测试的任何结果，模板缓冲都会保留它的值。默认行为不会更新模板缓冲，所以如果想写入模板缓冲的话，必须像任意选项指定至少一个不同的动作。