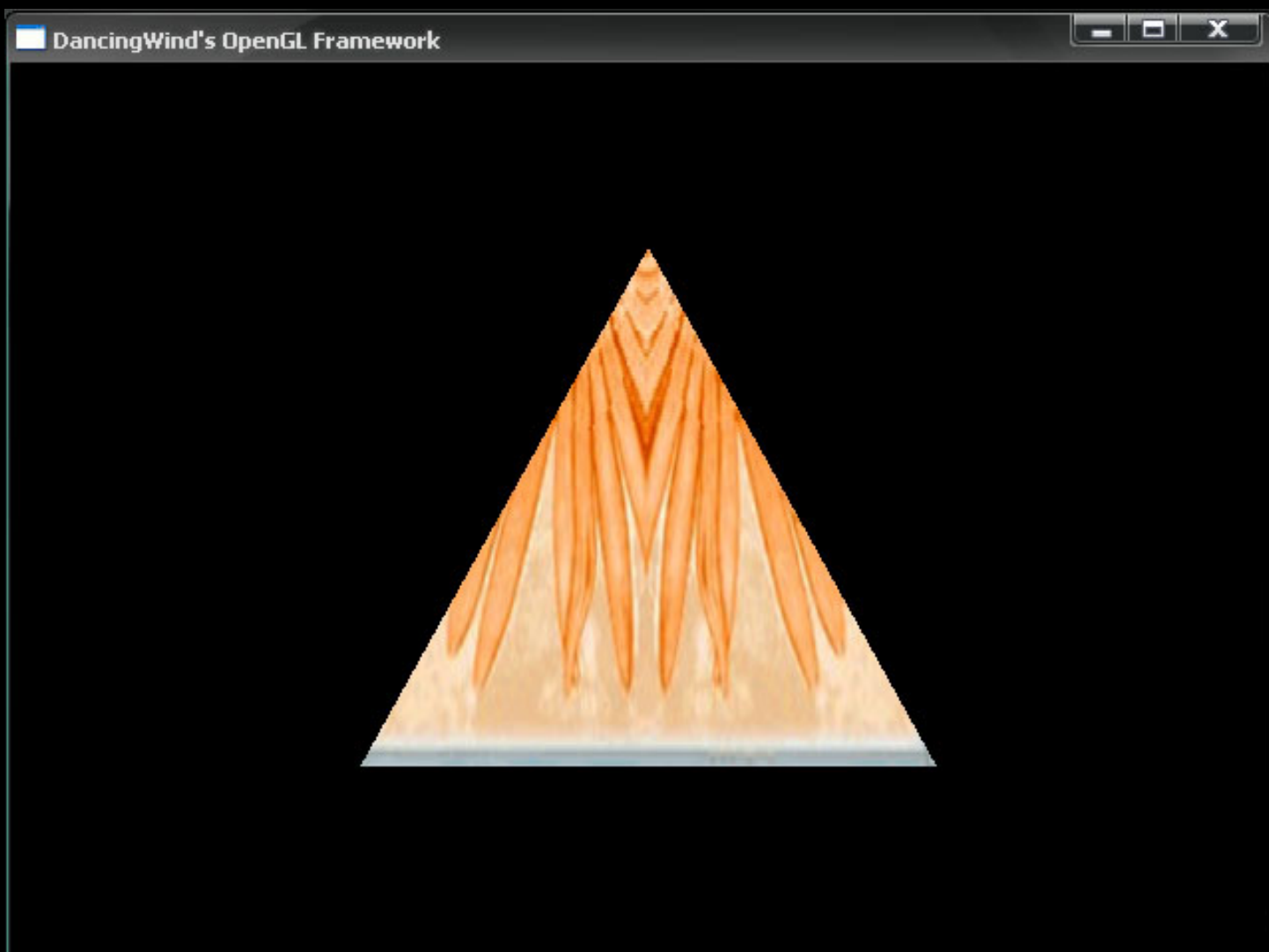


## 4、绘制带纹理三棱锥



NeHe SDK是把Nehe的教程中所介绍的所有功能，以面向对象的形式，提供给编程人员快速开发的一套编程接口。在下面的教程中，我将按NeHe SDK源码的功能分类，一步一步把这套api介绍给大家。如果你觉得有更好的学习方法，或者有其他有益的建议，请联系我。zhouwei02@mails.tsinghua.edu.cn，zhouwei506@mails.gucas.ac.cn

程序结构：

我们在第三课程程序的基础上添加一下功能：

1. 创建一个全局的纹理类
2. 利用纹理类载入纹理文件
3. 创建绘制带纹理坐标的三棱锥的函数
4. 设置当前的纹理并启用它，绘制带纹理坐标的三棱锥

为了使用纹理类,我们需要包含下面的头文件(texture.h),并声明一个全局的视口变量text2D。我们在主程序文件和绘制文件中添加如下的代码:

```

/*****新增的代码:包含纹理类的声明*****/
#include "texture.h"           // 包含纹理类的声明
/*****新增的代码:包含纹理类的声明:结束*****/

/*****新增的代码:创建Texture类*****/
Texture tex;                  // 使用全局变量Texture类的实例
int texID;                    // 使用全局变量texID,保存加载的纹理ID
/*****新增的代码:创建Text2D类:结束*****/

```

## 2、利用纹理类载入纹理文件

我们在IniScene函数中完成这个工作,首先调用纹理类的Load函数载入base.bmp图像文件,接着检查是否成功,如果不成功则弹出对话框,提示载入失败。

实现代码如下:

```

/*****初始化场景*****/
void IniScene(OpenGL* gl, ControlData* cont)
{
/*****新增的代码:载入纹理*****/
    texID = tex.Load("base.bmp");
    if(texID==0)
    {
        MessageBox(NULL,"不能加载base.bmp图像","Error",MB_OK|MB_ICONEXCLAMATION);
        cont->quit=true;
        return;
    }
/*****新增的代码:载入纹理:结束*****/
}
/*****初始化场景:结束*****/

```

## 3、创建绘制带纹理坐标的三棱锥的函数

我们使用标准的OpenGL函数完成这个功能

```

/*****新增的代码：绘制带纹理坐标的三棱锥*****/
// 绘制带纹理坐标的三棱锥
void DrawTexTri(void)
{
    glPushAttrib(GL_CURRENT_BIT);           // 保存当前的绘制属性
    glBegin(GL_TRIANGLES);
        // 前面
        glTexCoord2f(0.5f,0.5f);
        glVertex3f( 0.0f, 1.0f, 0.0f);
        glTexCoord2f(0.0f,0.0f);
        glVertex3f(-1.0f,-1.0f, 1.0f);
        glTexCoord2f(1.0f,0.0f);
        glVertex3f( 1.0f,-1.0f, 1.0f);

        // 右面
        glTexCoord2f(0.5f,0.5f);
        glVertex3f( 0.0f, 1.0f, 0.0f);
        glTexCoord2f(1.0f,0.0f);
        glVertex3f( 1.0f,-1.0f, 1.0f);
        glTexCoord2f(1.0f,1.0f);
        glVertex3f( 1.0f,-1.0f, -1.0f);

        // 后面
        glTexCoord2f(0.5f,0.5f);
        glVertex3f( 0.0f, 1.0f, 0.0f);
        glTexCoord2f(1.0f,1.0f);
        glVertex3f( 1.0f,-1.0f, -1.0f);
        glTexCoord2f(0.0f,1.0f);
        glVertex3f(-1.0f,-1.0f, -1.0f);

        // 左面
        glTexCoord2f(0.5f,0.5f);
        glVertex3f( 0.0f, 1.0f, 0.0f);
        glTexCoord2f(1.0f,1.0f);
        glVertex3f(-1.0f,-1.0f,-1.0f);
        glTexCoord2f(0.0f,0.0f);
        glVertex3f(-1.0f,-1.0f, 1.0f);
    glEnd();
    glPopAttrib();                          // 弹出保存的绘制属性
}
/*****新增的代码：绘制带纹理坐标的三棱锥:结束*****/

```

最后我们启用纹理，并绘制一个带纹理的三棱锥，代码如下：

```
/******新增的代码：绘制带纹理三棱锥  
*****/  
tex.Set(texID);  
cont->state->SetTexturing(true);  
DrawTexTri();  
/******新增的代码：绘制带纹理三棱锥:结束******/
```

好了，这一课的内容到这就结束了，相信你已经能熟练的使用纹理了。下一课，我将根据我们所学的知识，创建一个具有基本功能的框架结构，并使用这个框架结构逐步解释NeHe SDK中的各个类的详细使用方法。