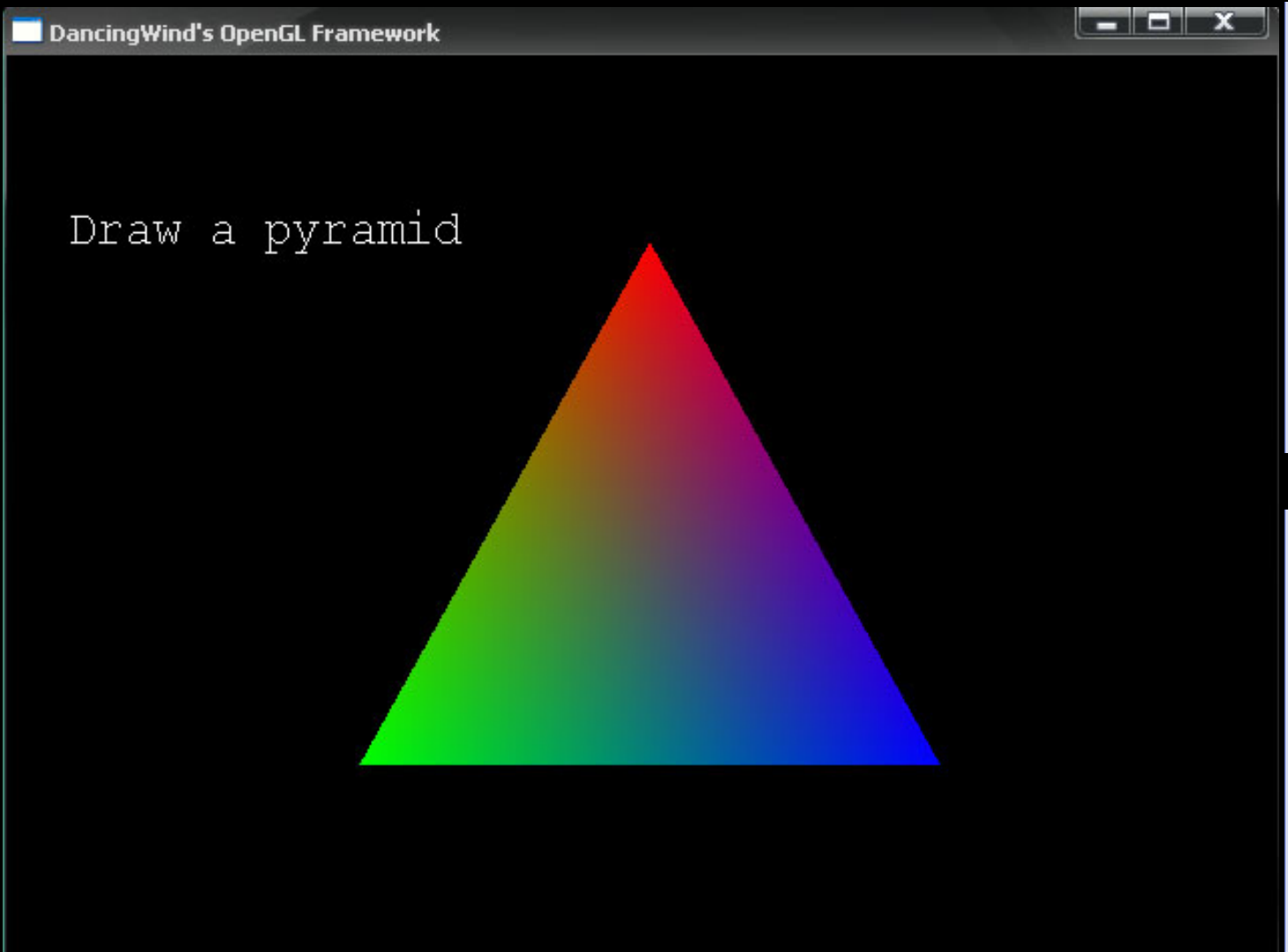


3、绘制2D英文字符



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：

我们在第二课程程序的基础上添加一下功能：

1. 创建一个全局的2D文本类，在场景中绘制2D英文字符
2. 按默认方式初始化2D文本类
3. 使用它在屏幕绘制2D英文字符

为了使用视口类,我们需要包含下面的头文件(text.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2D文本类的声明*****/
#include "text.h"           // 包含2D文字类的声明
/*****新增的代码：包含2D文本类的声明:结束*****/

/*****新增的代码：创建Text2D类*****/
extern Text2D text2D;      // 使用全局变量Text2D类实例
/*****新增的代码：创建Text2D类:结束*****/

```

2、初始化Text2D类

为了使用这个类，我们需要填充TextType数据结构，他用来设置文本的参数，TextType结构的定义如下：

[TextType](#)

我们使用如下的默认设置来初始化我们的字体。

```

/*****新增的代码：初始化字体*****/
TextType ttype;           // 设置字体结构
ttype.name="Courier New"; // 字体名称为"Courier New"
ttype.size=24;            // 字体大小为24
ttype.bold=false;        // 不使用粗体
ttype.italic=false;      // 不使用斜体
ttype.underline=false;   // 不使用下划线
text2D.Setup(&WinOpenGL,ttype); // 设置字体
/*****新增的代码：初始化字体:结束*****/

```

3、在屏幕上绘制2D文字

我重载了C++的流输出符号，你可以向使用std::cout一样输出，但只能输出英文字符，整数。这对于图形应用来说应该是够了。

下面是它的绘制代码，我们在Draw.cpp文件中的DrawScene函数中添加这个功能。

```

/*****新增的代码：绘制文本*****/
*****/
view.Save();           // 保存当前的视口矩阵
view.Reset();          // 重置视口矩阵

```

```
view.Translate(0.0f,0.0f,-5.0f);    // 把绘制的模型坐标向-Z轴移动5个单位
view.Pos2D(-2.5f,1.0f);    // 设置在z=0平面，绘制点的位置
glPushAttrib(GL_DEPTH_BUFFER_BIT);    // 保存深度缓存的属性
glDisable(GL_DEPTH_TEST);    // 禁用深度测试，可以让我们的文本始终显示在最上层
text2D<<"Draw a pyramid";    // 绘制文本
glPopAttrib();    // 弹出保存的深度缓存属性
view.Restore();    // 弹出保存的视口矩阵
/*****新增的代码：绘制文本:结束*****/
```

好了,上面就是简单的使用text2D类绘制英文字符的简单实用,希望你能很轻松的使用它:)