2016.8.14 创建文档

上次的DxLearningDemo本来打算作为一个长期使用的框架，也是花了不少心思不停的修改来着，但是现在还是觉得一团乱麻。决定开一个新坑，重新写一次。

**上次的缺陷：**

1. 在不同类型的模型之间选择了继承关系，直接导致了结构复杂化；
2. 试图将所有模型、相机等全都看做完全相同物体来管理，导致结构冗余；
3. 模块之间耦合太多，例如相机与模型与其他等；

**本次改进：**

1. 依然以文件夹和过滤器的方式来区分不同的模块，但是要严格区分模块之间的引用权限
   1. 分为一个公共库模块，一个主循环模块，若干功能模块；
   2. 每个模块都分为public部分和private部分，private部分不可被除自己以外的任意模块调用；唯一例外的情况是：公共模块不可调用其他任意模块，主循环模块不可以被其他任意模块调用；
   3. Public部分只能由管理器和纯虚接口定义；
2. 尽可能减少继承关系，以组合的方式来生成新的类；

**本次设计：**

1. 目标
   1. 可以方便的创造、加载、修改各种想要的效果，并方便的展示；
   2. 给予一定的鼠标和相机响应，方便查看；
   3. 锻炼设计能力；
2. 模块
   1. 主循环调用管理器；
   2. 模型管理器；
   3. 相机管理器；
   4. 事件响应管理器；
   5. 着色器管理器；
   6. 物理系统；
   7. 其他待定；
3. 开发顺序

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 内容 | 大致目标 | 预计时间 | 开始 | 结束 | 备注 |
| Main loop | 主循环 | 0 | 8/14 |  |  |
| Model | 显示简单模型 | 0.5D | 8/14 |  |  |
| Model factory | 创建简单模型 | 0.5D |  |  |  |
| Model manager | 显示多个简单模型 | 1D |  |  |  |
| Texture | 为简单模型增加贴图 | 0.5D |  |  |  |
| Model factory/config | 加载、配置模型 | 2D |  |  |  |
| Camera | 相机 | 1D |  |  |  |
| Events | 鼠标响应 | 2D |  |  |  |
| Shader manager | 着色器 | 1D |  |  |  |

1. 命名约定，对匈牙利命名法稍加修改以方便以后程序化的检查代码
   1. 属性部分
      1. 大写字母+下划线开头，代表类型的属性
         1. N\_：namespace
         2. I\_：interface
         3. C\_：instance
         4. T\_：typedef
         5. F\_：function
      2. 小写字母+下环线开头，代表实例的属性
         1. m\_：类属性
         2. s\_：cpp内静态变量
         3. g\_：全局变量
         4. c\_：cpp内静态常量
         5. t\_：所有临时变量
         6. a\_：函数传参
   2. 类型部分
      1. 使用匈牙利命名法的字符+下划线
         1. i\_：int
         2. vec\_：vector
      2. 对于非内建类型，使用其父类描述全名+下划线
         1. A\_：代表继承自父类A
   3. 描述部分
      1. 以大小写区分单词，不含下划线
         1. 对于class、function，使用大写字母开头
         2. 对于instance，使用小写字母开头
2. （不定时更新）

2016.8.14 Main Loop

**开发内容：**

1. 创建全局数据区
2. 创建main loop并生成全局数据
3. 全局数据区增加Direct3DDevice指针

2016.8.14 Model

**开发内容：**

1. 创建简单模型