说明文档

姓名：Call 偶围城 学号：XXXXXXXX

本程序实现了光线追踪算法，实现了光泽反射、区域光照、阴影等效果，并在两个场景做了 demo。在文档的最开始，我首先要感谢王锐老师一学期以来的教导，王老师不仅让我对计算机图形学有了新的认识，而且王老师严谨的治学态度、精益求精的科研精神也深深影响着我。同时我也要感谢《光线跟踪算法技术》这本书的作者 Kevin Suffern，我作业中一些细节的处理参考了这本书。最后，我要感谢在我完成此次作业期间所有对我有过帮助的人。

本说明文档分为四个部分：编译环境和使用说明的介绍、整体思想的介绍、数据结构的介绍以及结果的展示。

一、编译环境及使用说明

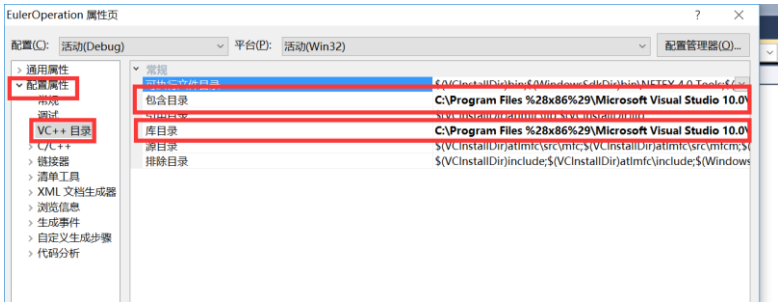
我使用的 IDE 是 VS2010，系统是 win10 64 位 4G 内存的虚拟机。在 VS 中我配置了 OpenGL，需要的配置文件在压缩包内，具体配置方法如下：

1. 将 glut.lib 和 glut32.lib 这两个静态函数库复制到文件目录的 lib 文件夹下 X:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\lib；

2. 将 glut.dll 和 glut32.dll 这两个动态库文件放到操作系统目录下面的 X:\Windows\system32 文件夹内（32 位系统）或X:\Windows\SysWOW64(64 位系统)；

3. 将解压得到的头文件 glut.h 复制到 X:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\include\GL，如果在 incluce 目录下没有 GL 文件夹，则需要手动创建；

4. 打开项目属性，设置包含目录和库目录，分别添加 gl 文件夹和 lib 文件夹所在的目录。



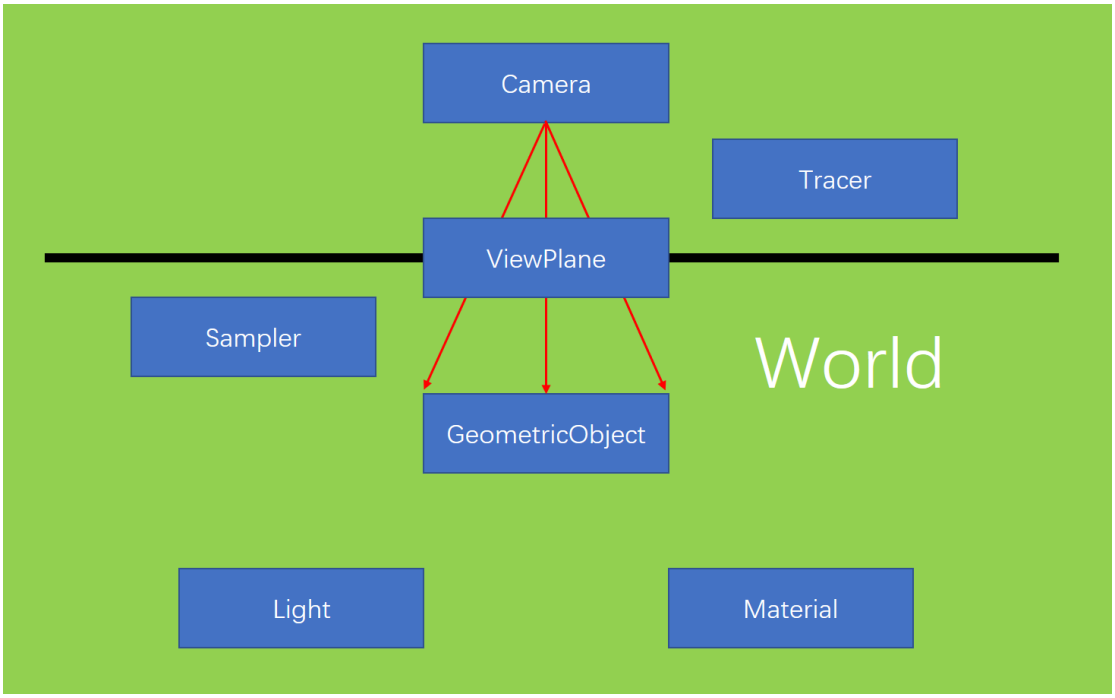
运行程序会有提示"选择场景‘1’或场景‘2’："，这时候输入 1 或者 2，然后回车就可以了。界面会显示"计算需要一些时间，请稍等 … "，如果场景比较复杂可能需要的时间就会久一些。等结果出来的时候，界面会提示总共耗时多长时间。我的程序会把每次运行的结果以 BMP 格式保存在项目文件夹中，界面会有提示结果图片的名称。

二、整体思想

以相机（Camera）为光线起点，向视平面（ViewPlane）的每个像素点发出一条射线（Ray）。如果此射线与场景中的几何物体（GeometricObject）相交，则需要计算出离视平面最近采样交点（Sampler）的颜色，也就得到了相关视平面像素点的颜色。而采样交点的颜色是由采样点所在物体的材质（Material）、场景中的光源（Light），场景中的其它物体及背景等多方面因素相互作用决定的。如果光线没有和任何几何物体相交，则视为光线与背景相交，相关视平面像素点用背景色着色。通过对整个视平面的着色，最后会得到一个逻辑上的二维颜色矩阵，通过 OpenGL 将这个矩阵显示出来，同时把该矩阵以 BMP 格式的文件永久保存。

三、数据结构

先简单的介绍一下，我的程序一共有 39 个类，其中核心的类有 7 组，工具类有 12 个。下图是核心类之间的关系：



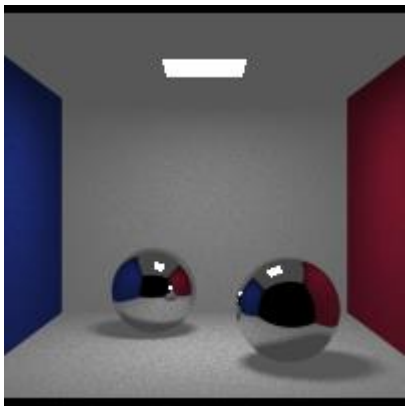
光线从照相机出发，通过视平面每一个像素点，与几何物体相交或者与背景相交。相机和追踪器基于几何物体的材质和场景中光源位置对每个采样交点着色。下图简要的说明了核心类（以及 BRDF）的父子类关系：

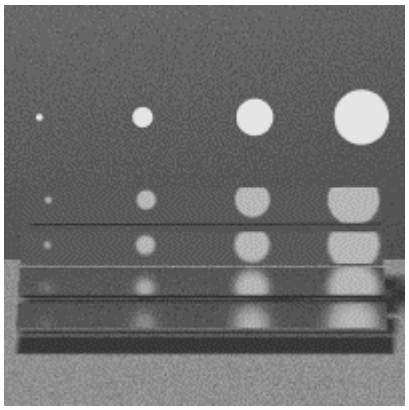
以下是工具类的展示，用于保存一些基本几何元素或者光线信息等，以及定义一些运算，做结果展示等。其中相对比较重要的有容器类 ShadeRec，对齐包围盒 BBox，结果显示类 Result 等。

我的程序类还有个别类没有实际的作用，有几个是被重写了，也有最后没有用到的类如 RayCast，还有 test 用于调用实例的 print 方法以测试信息的正确性。

四、结果展示

我的程序会把结果用 OpenGL 展示出来，同时也会将结果用 BMP 文件的形式永久的保存下来，用场景编号和日期时间进行命名。以下是结果的展示：





在文档的最后我要再次感谢王老师一学期以来的指导，在王老师的计算机图形学课程中我学到了很多知识，同时也开拓了自己的视野。