

实验6

实验名称Ray Tracing 光线跟踪

姓 名：雷昱

学 号：22920202204666

学 院：信息学院

专 业：软件工程

年 级：2020级

二〇二二年 **5** 月 **18** 日

**实验6、Ray Tracing 光线跟踪**

建议阅读资料：

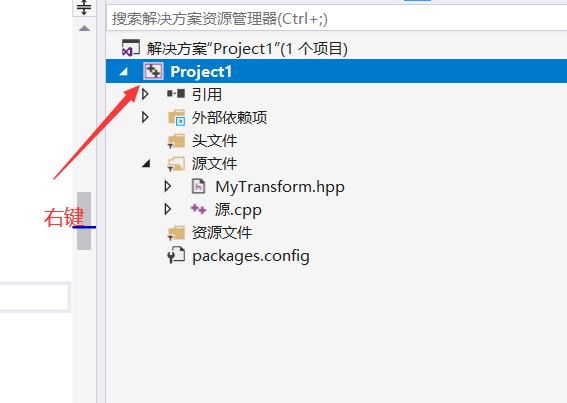
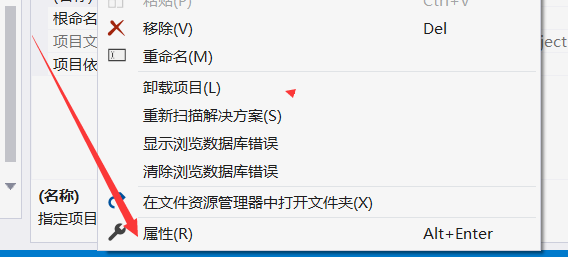
1. 课件

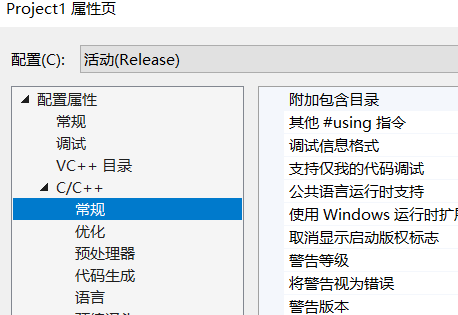
学习要求：

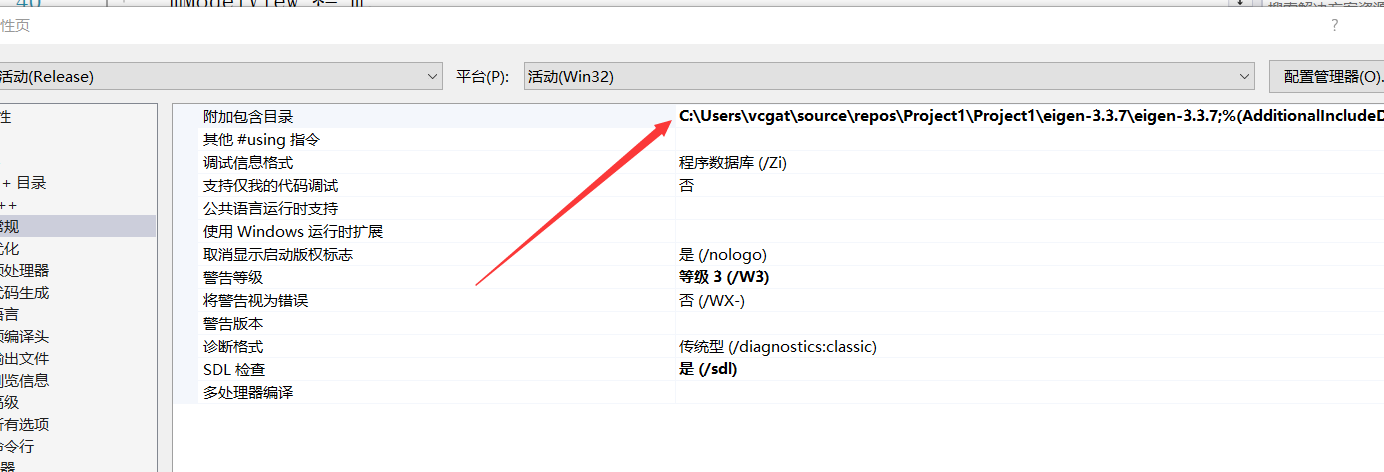
1. 掌握Whitted Style光线跟踪算法

**Task1. 自行实现Whitted光线跟踪程序，要求实现漫反射、镜面反射和折射功能。请严格按照以下要求实现：**

1. **创建空工程，建议设置成release模式（本次实验不需要配置freeglut库）**
2. **配置eigen库（eigen用于矩阵和向量的计算）。做法是：将eigen库解压，同时在刚才创建的工程中设置eigen库路径。右键点击工程->属性->C/C++ 常规 -> 将eigen所在目录填到“附加包含目录”这一栏。**

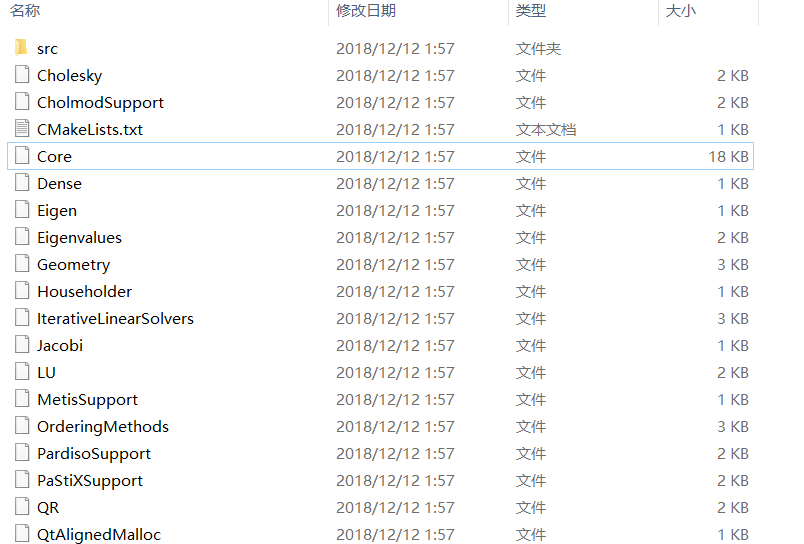
 





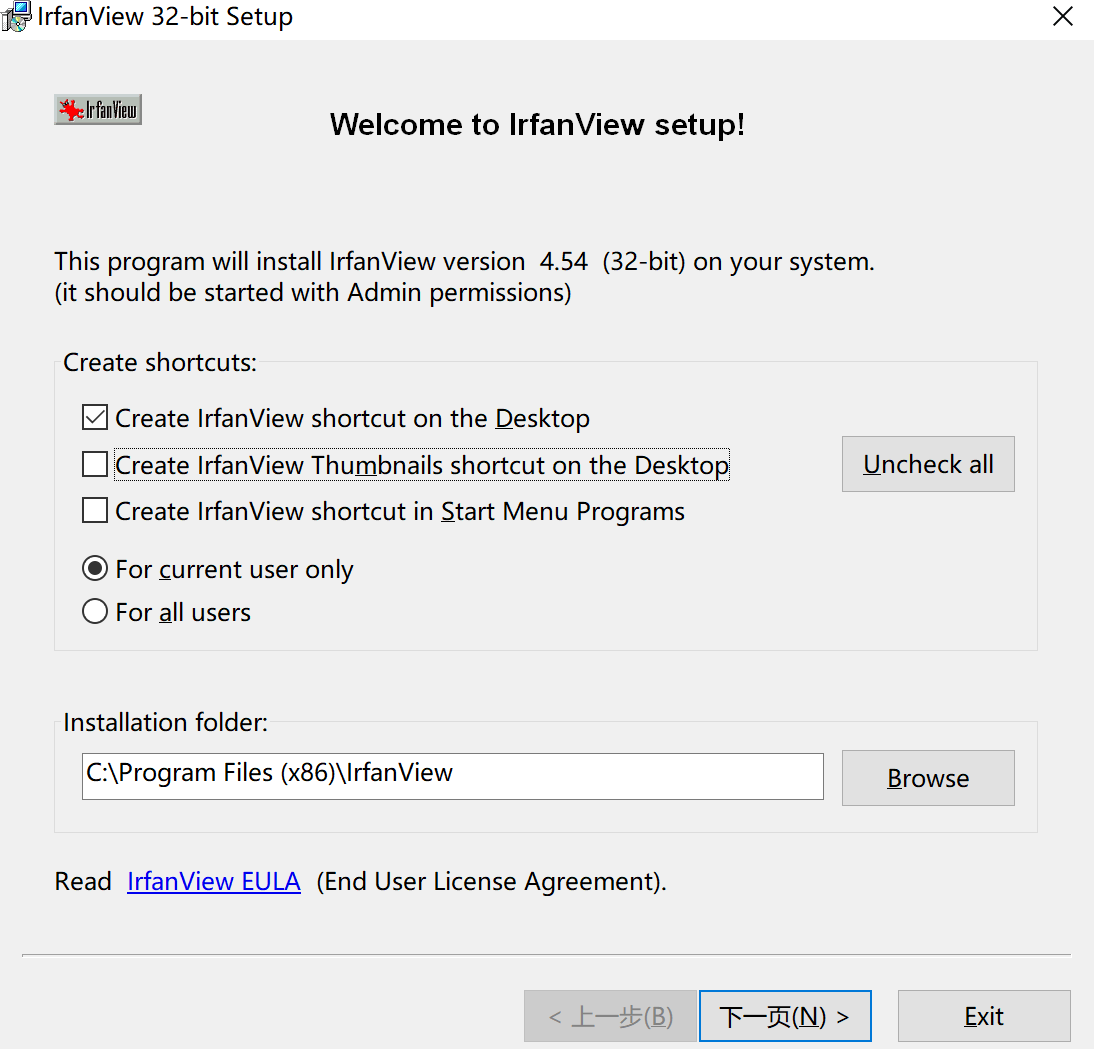
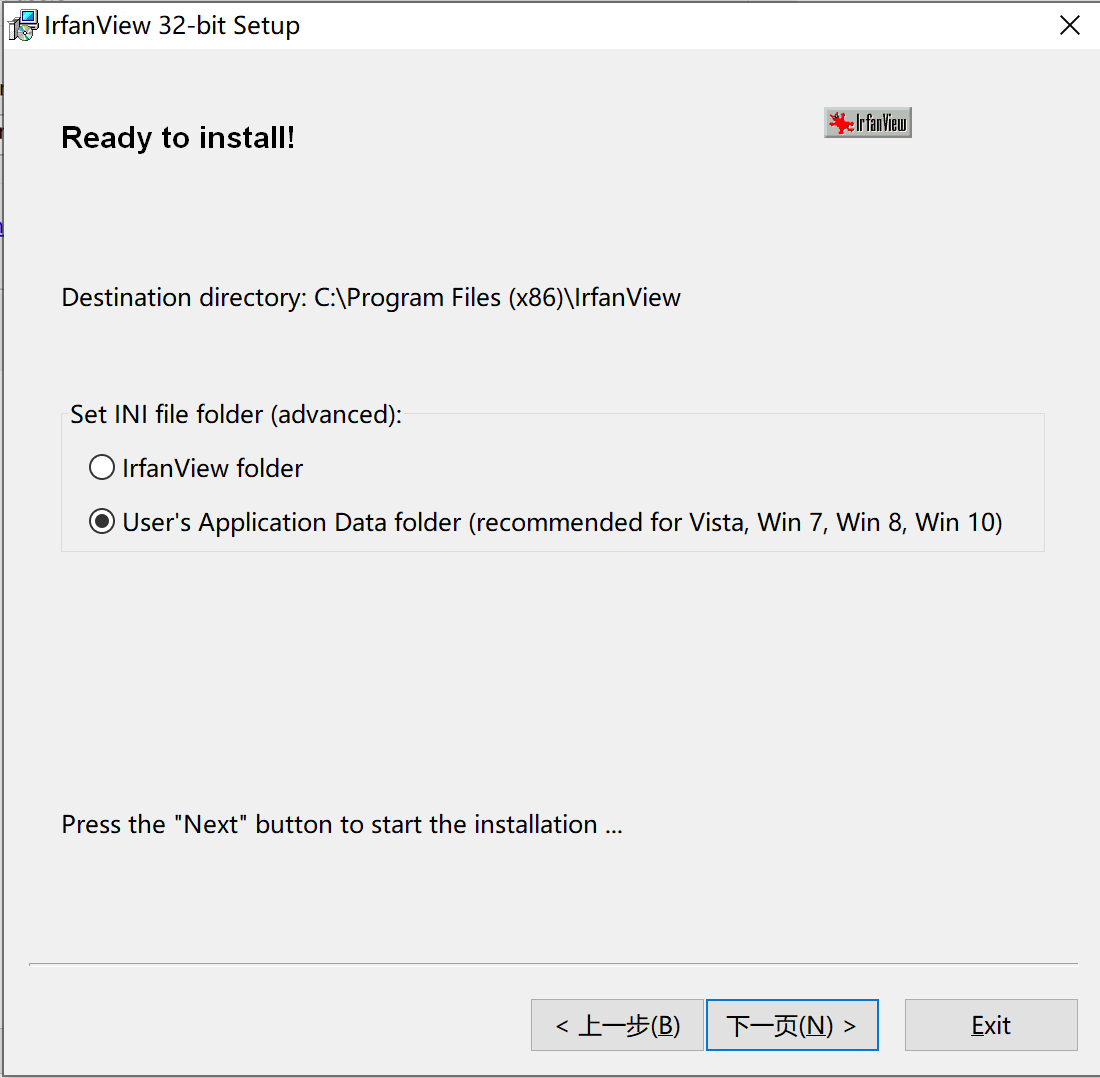
**注意：上述路径是我的路径，请根据自己的解压位置填写路径。**

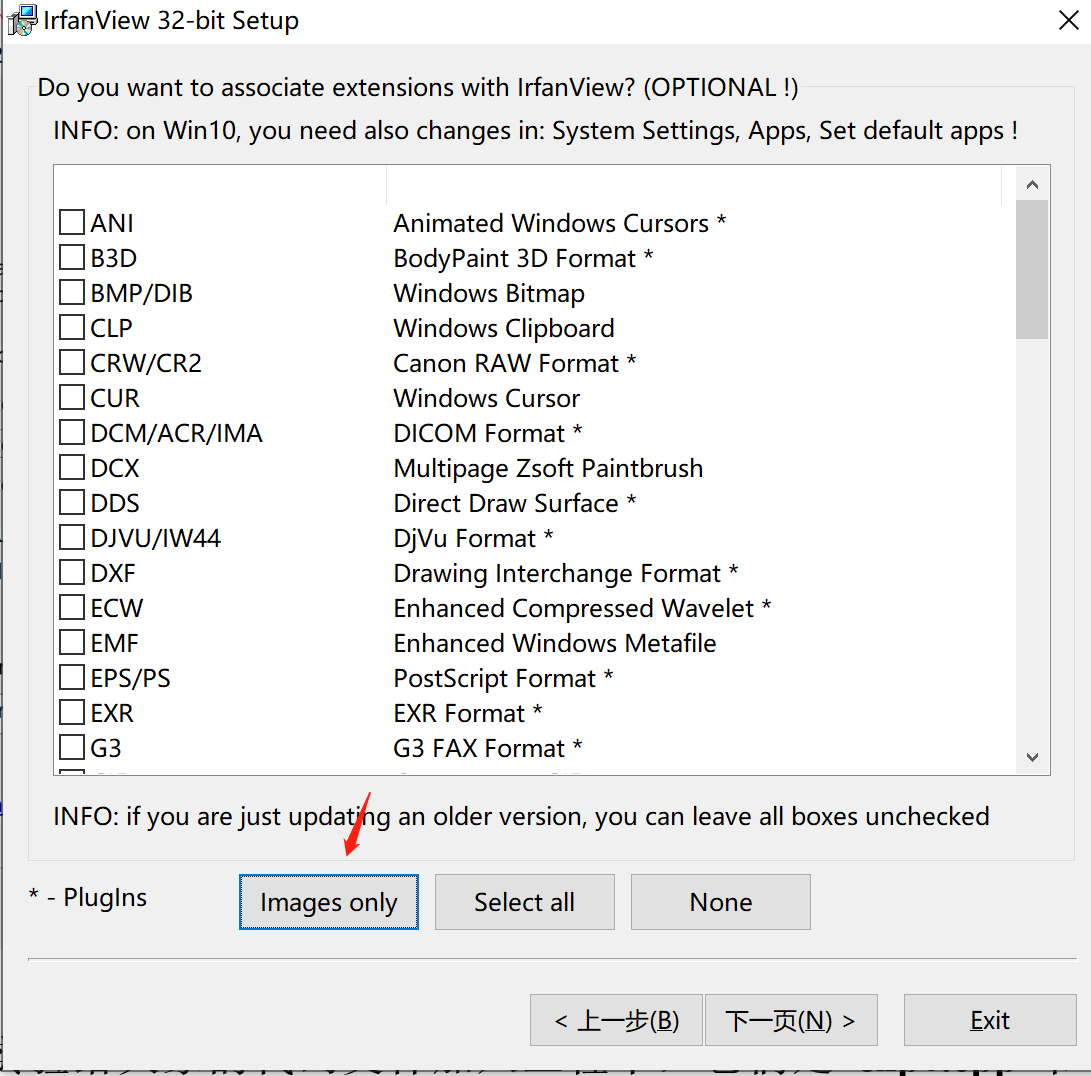
**应该定位到这个目录的上一级：**



1. **安装IrFanViewer，用于查看生成的ppm图像文件。**

安装过程若没有说明，则为默认选项。此处路径是默认安装位置，可根据自己需求选择路径。

不需要勾选其他辅助拓展。

1. **将本次实验给大家的代码文件加入工程中，它们是exp6.cpp和camera.hpp， ppm.hpp， ray.hpp， raytracer.hpp， scene.hpp。在本次实验过程中，除非特别说明，不允许改动exp6.cpp的内容（也不必关心具体代码）。**

**其中exp6.cpp包含主函数用于调用光线跟踪函数；**

**camera.hpp用于生成光线，关于光线生成的函数在此；**

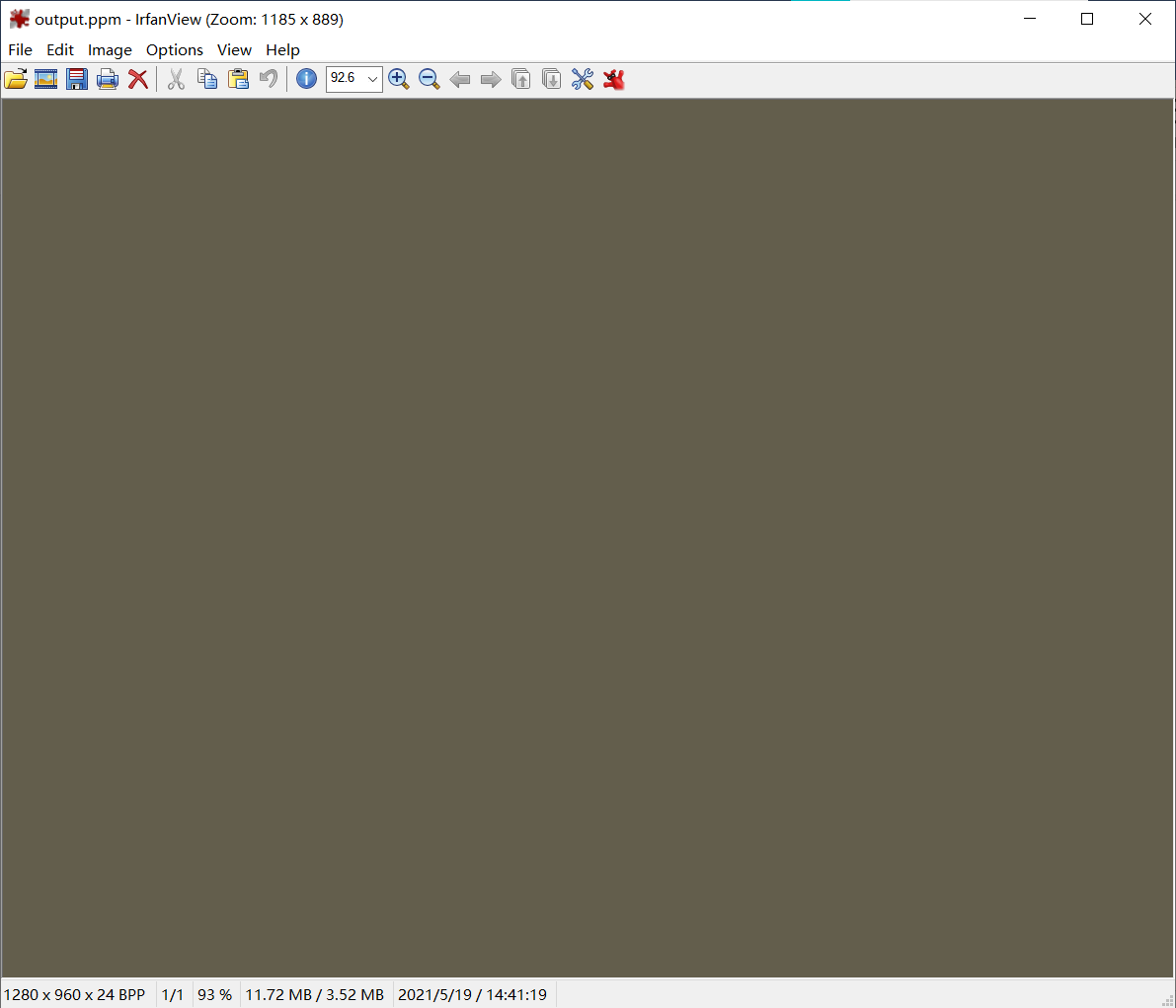
**ray.hpp是光线的类；**

**scene.hpp组织场景，这里主要是球面；**

**raytracer.hpp光线跟踪的主程序；**

**ppm.hpp是用于ppm图像格式的存储。**

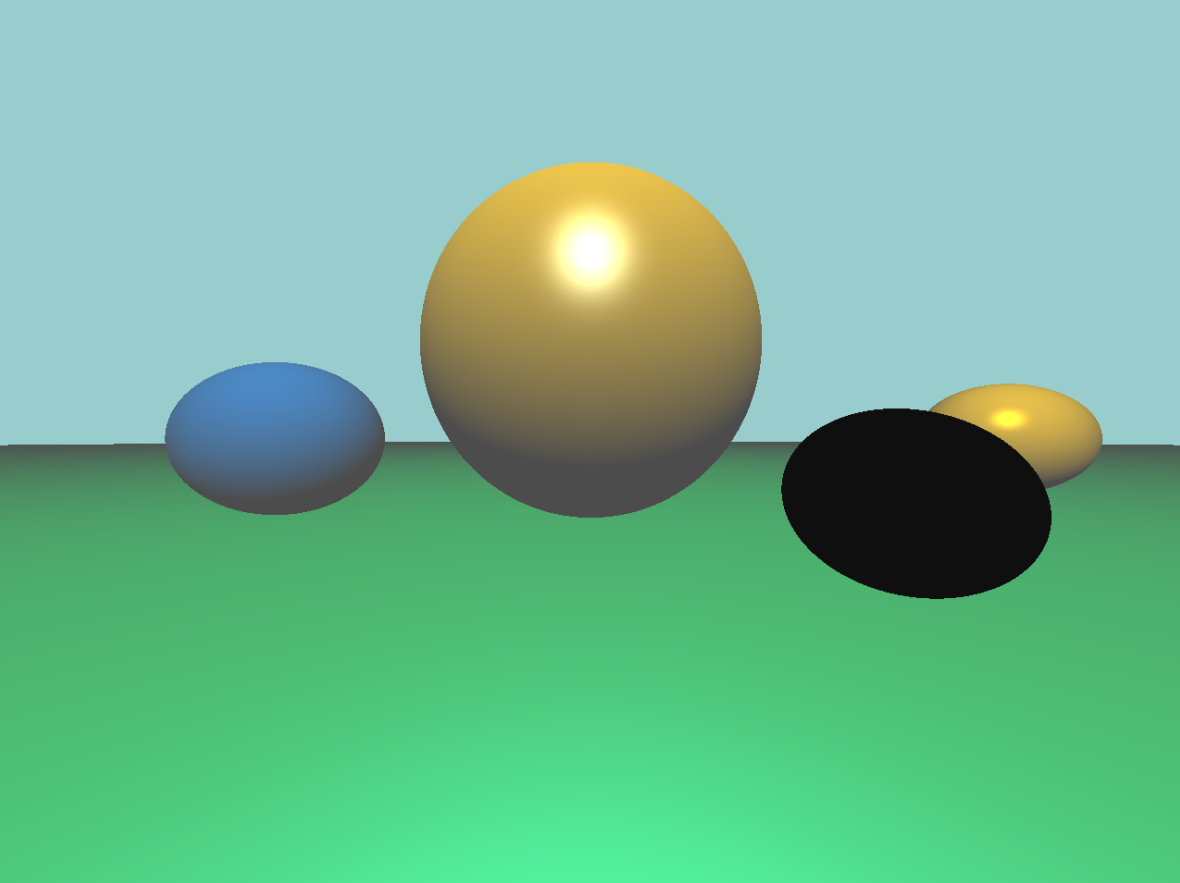
1. **不做任何更改，如果配置正确，运行代码后，你应该能在程序文件夹下找到output.ppm文件，用IrFanViewer打开应该是一张纯色的图像。**



1. **修改camera.hpp中生成光线的函数**

Ray GenerateRay(float u, float v)

**如果改对的话，你应该能看到这样的图像，右侧的球呈现出黑色是因为其漫反射属性被设得很低。**

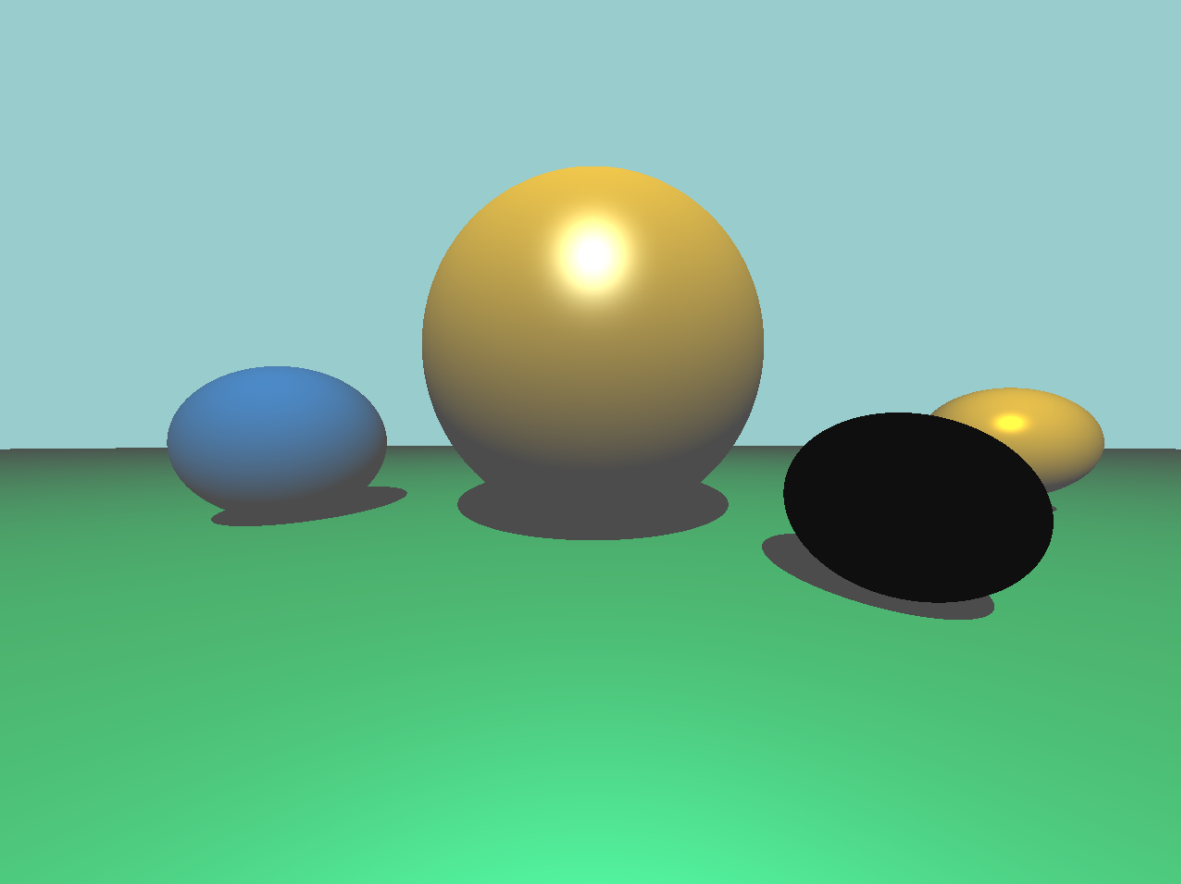


1. **在RayTracer.hpp的**函数

Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false)

**加入shadow ray是否能看到光源的代码，以生成阴影。**

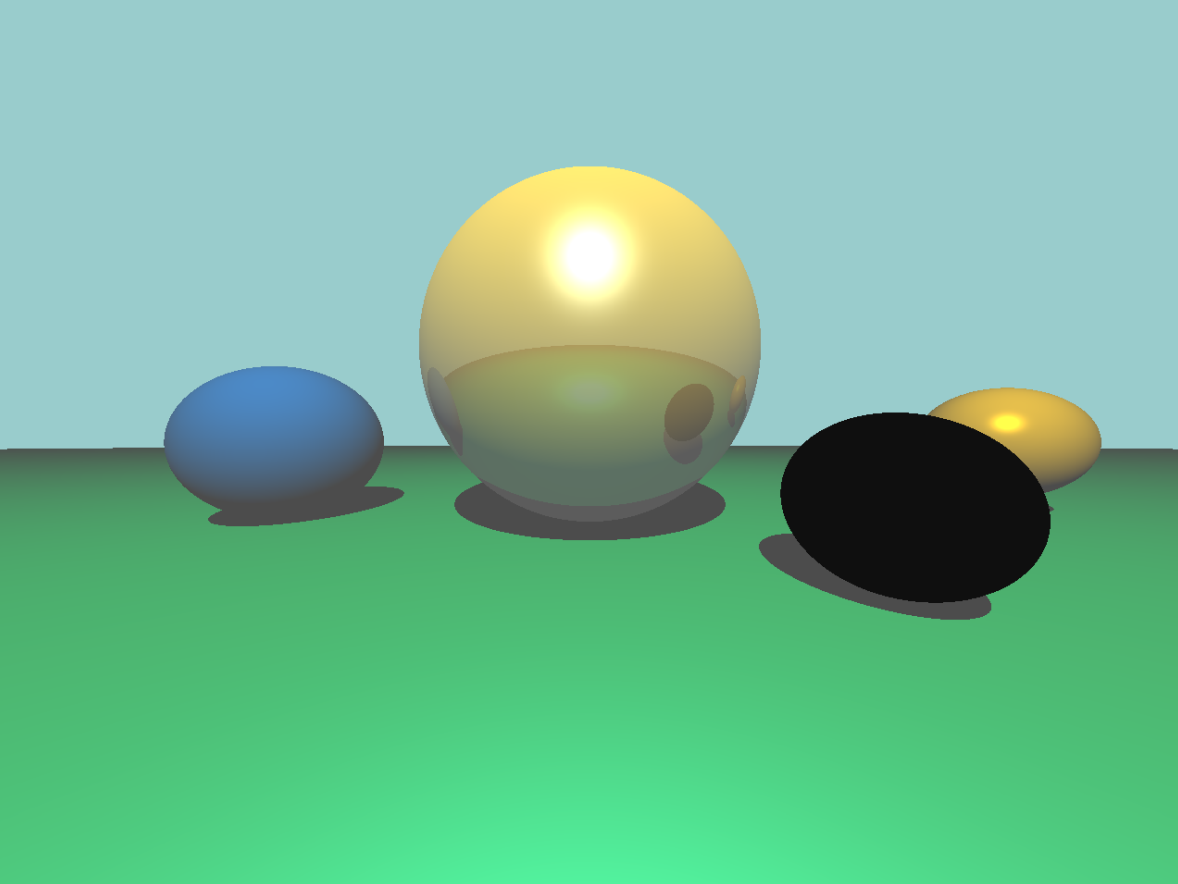
**实现正确的话，你能看到阴影。**



1. **在RayTracer.hpp的**函数

Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false)

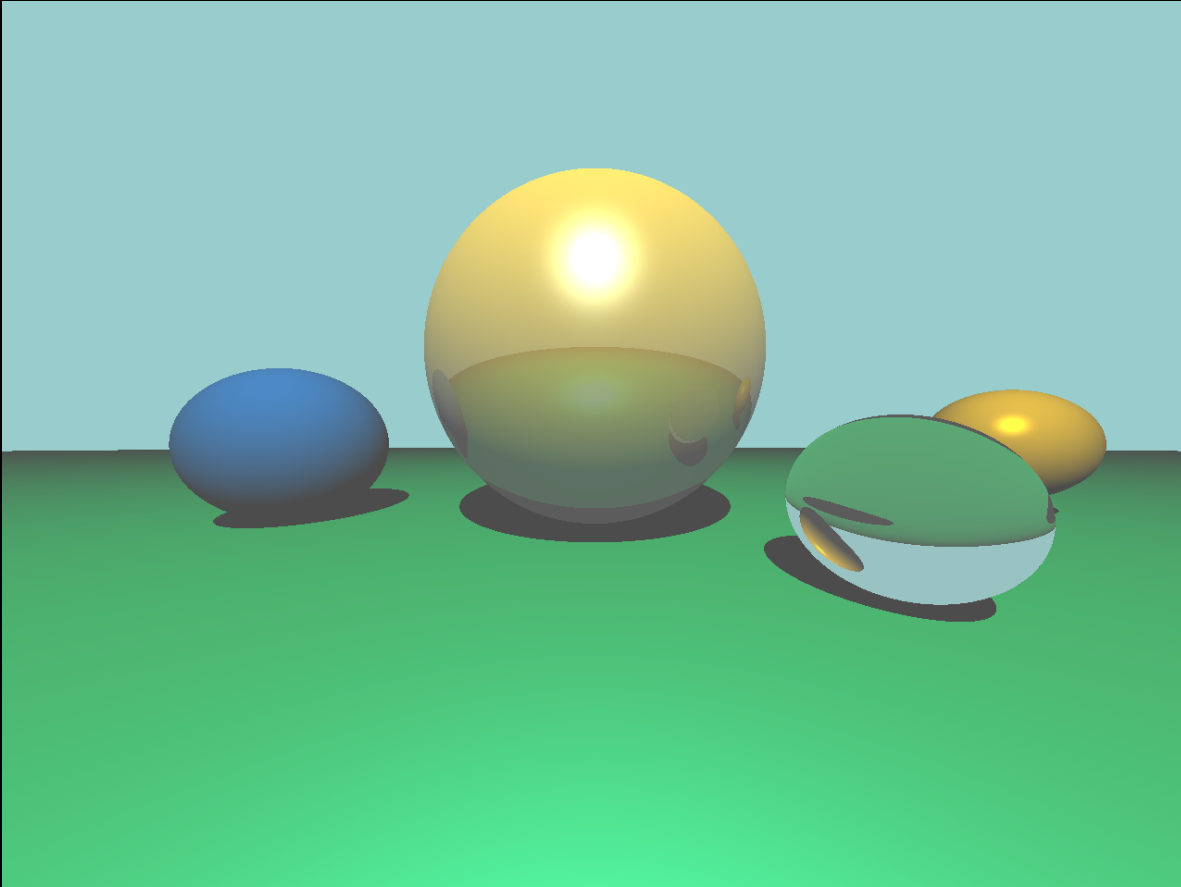
**加入完成镜面反射的递归代码。如果加入正确的话，你能看到中间黄球有镜面反射：**



1. **在RayTracer.hpp的**函数

Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false)

**加入完成折射的递归代码。如果加入正确的话，你能看到右侧小球有折射效果：**



1. **附加选项：（根据实现情况，期末最终的实验成绩可以\*1.05~1.1）**

**在此代码基础上，生成一段小球从天而降的动画，可以考虑加入运动模糊、软阴影效果，小球弹跳符合物理规律。**

**作业提交说明：**

本次实验共有1个任务。

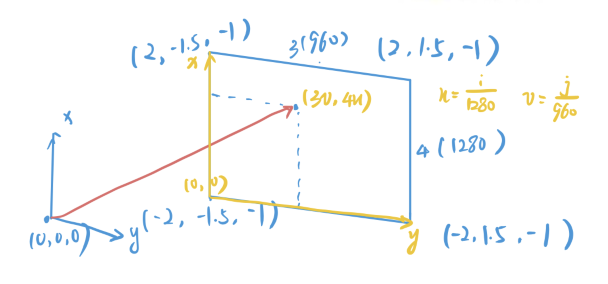
提交方式为：*将代码、实验报告、对应帧的ppm文件、附加题的gif(可选)*放到一个文件夹中，命名为“您的学号\_姓名”，打包上传到ftp服务器中相应目录下。每次实验作业的提交截止日期为下一次实验课前一天晚上。

特别说明：本次实验1个任务都需要提交。您需要提供一个完整的文档（不限格式），逐条说明您完成每一条任务的具体情况。

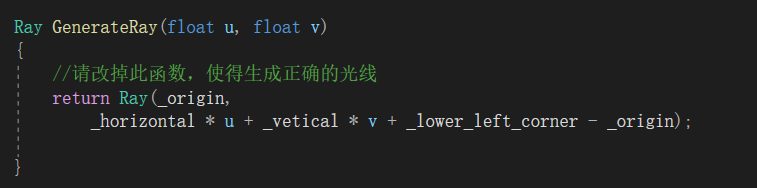
实验步骤如下：

1.修改Ray GenerateRay(float u, float v)

①光线生成计算



②代码展示

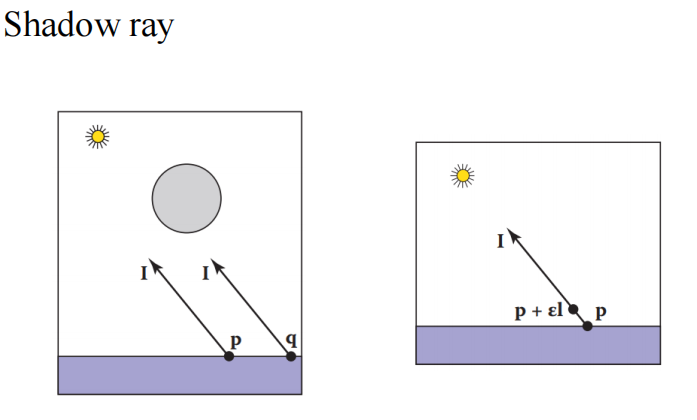


2.在Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false)中加入shadow ray是否能看到光源的代码，以生成阴影。

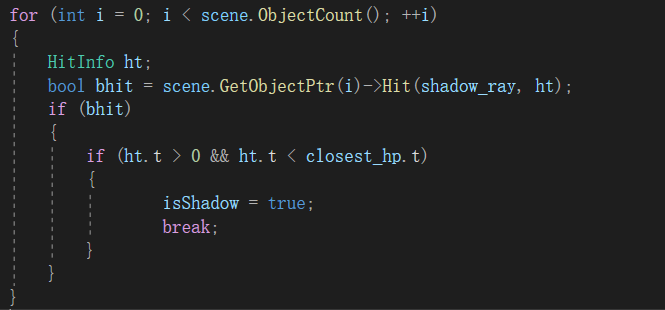
①将shadow\_ray与小球求交，判断是否被遮挡。

当被遮挡时无镜面、漫反射。

②如图所示shadow ray

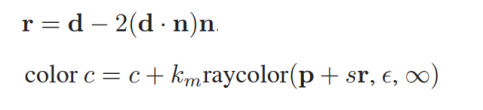


③代码实现



3.在Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false)加入完成镜面反射的递归代码。如果加入正确的话，你能看到中间黄球有镜面反射。

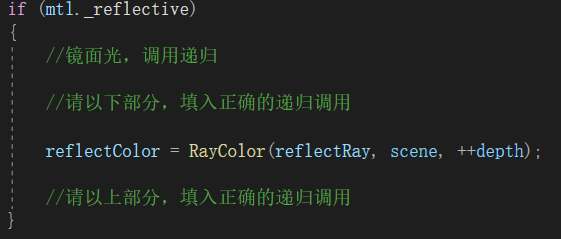
①反射计算公式



②分析

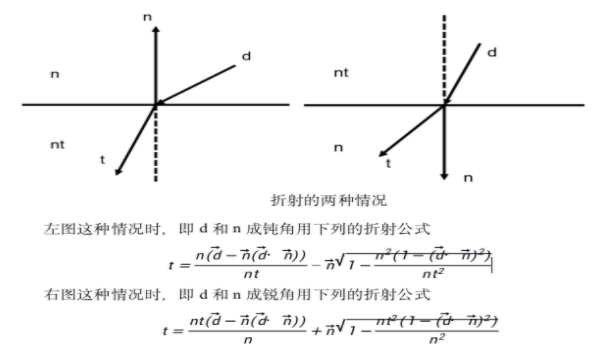
这样的递归定义来决定像素颜色，这个实质上就是不断加上反射光带来的颜色，其中km就是这次反射的表面颜色，p+sr就是入射光线的方向（r是入射光线，p就是交点）

③代码展示



4.在Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false)加入完成折射的递归代码。如果加入正确的话，你能看到右侧小球有折射效果。

①折射



②代码展示

