

计算机图形学实验

实验6 光线跟踪

姓 名：马鑫\_\_\_\_\_\_\_\_\_\_\_

学 号：\_37220232203780\_

学 院：\_信息学院\_\_\_\_\_\_

专 业：\_数字媒体技术\_\_

年 级：\_2023级\_\_\_\_\_\_\_

2025年 5月 25日

目录

[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。 3](#_Toc19356)

[1.完成。配置环境 3](#_Toc26935)

[2.完成。修改camera.hpp中生成光线的函数 3](#_Toc18672)

[3.完成。在RayTracer.hpp的函数 4](#_Toc919)

[Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false) 4](#_Toc31710)

[加入shadow ray是否能看到光源的代码，以生成阴影。 4](#_Toc14663)

[4. 完成。在RayTracer.hpp的函数 6](#_Toc29056)

[Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false) 6](#_Toc3563)

[加入完成镜面反射的递归代码。 6](#_Toc26133)

[5.完成。在RayTracer.hpp的函数 7](#_Toc8009)

[Vector3f RayColor(const Ray& ray, Scene& scene, int depth=0, bool test=false) 7](#_Toc4293)

[加入完成折射的递归代码。 7](#_Toc32377)

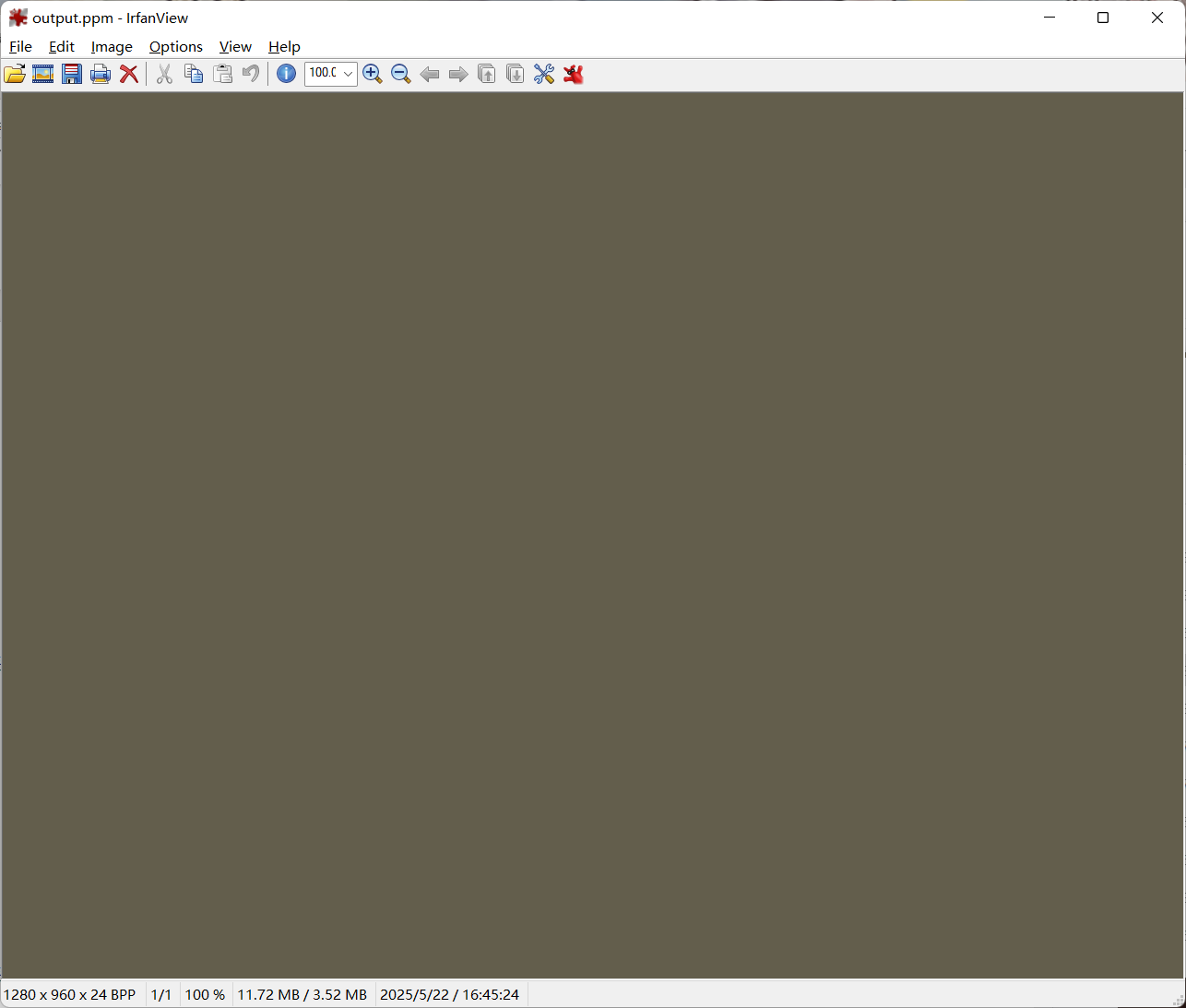
[6.完成。 在此代码基础上，生成一段小球从天而降的动画，可以考虑加入运动模糊、软阴影效果，小球弹跳符合物理规律。 8](#_Toc25233)

# 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。**

## 1.完成。配置环境

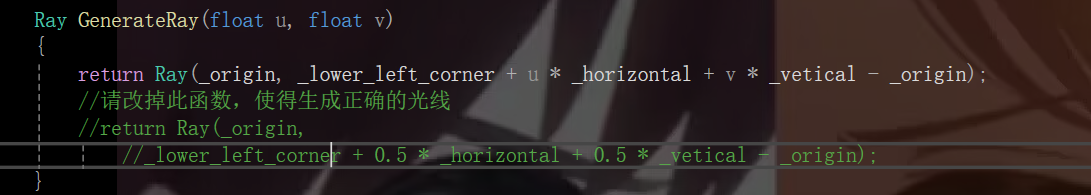
操作流程：按照1~5的要求完成

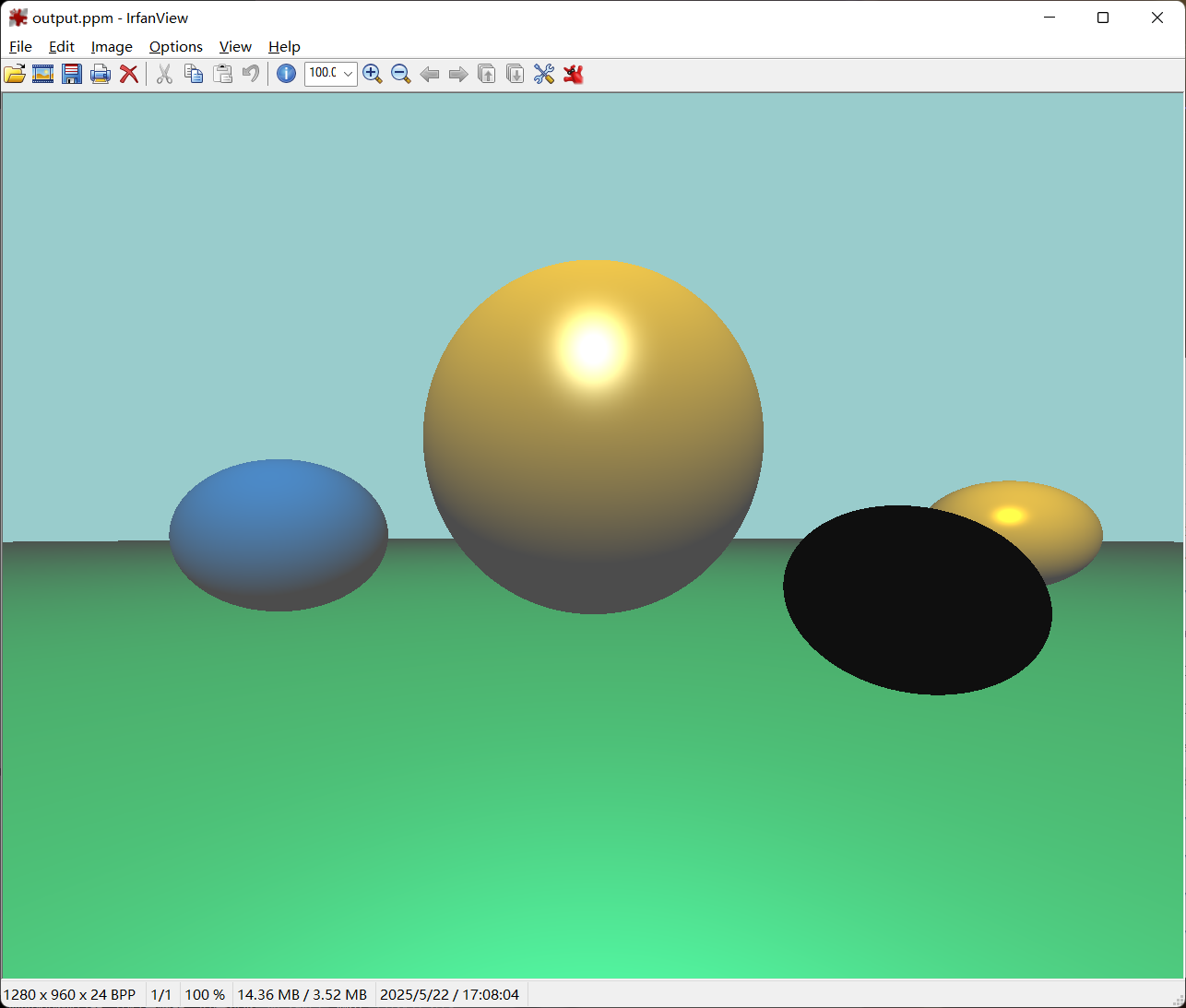
关键代码截图：略

运行结果截图： 

## 2.完成。修改camera.hpp中生成光线的函数

操作流程：修改camera.hpp中的GenerateRay函数，使之返回的光线的方向是从光源指向从左下角开始计算的第u的百分比行，第v的百分比列。

关键代码截图：

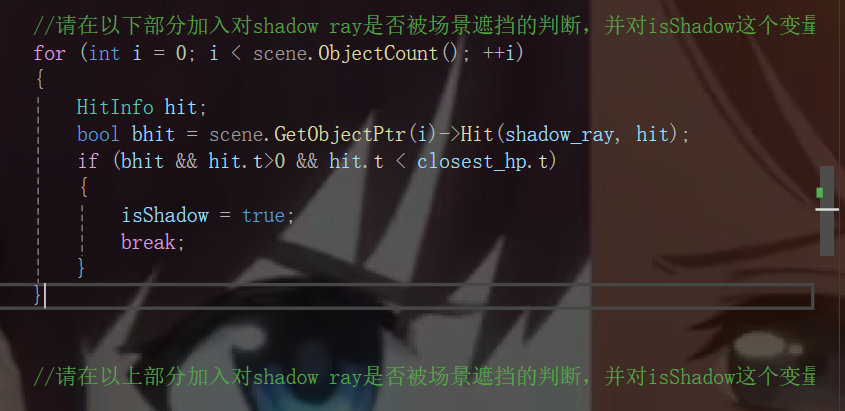
运行结果截图： 

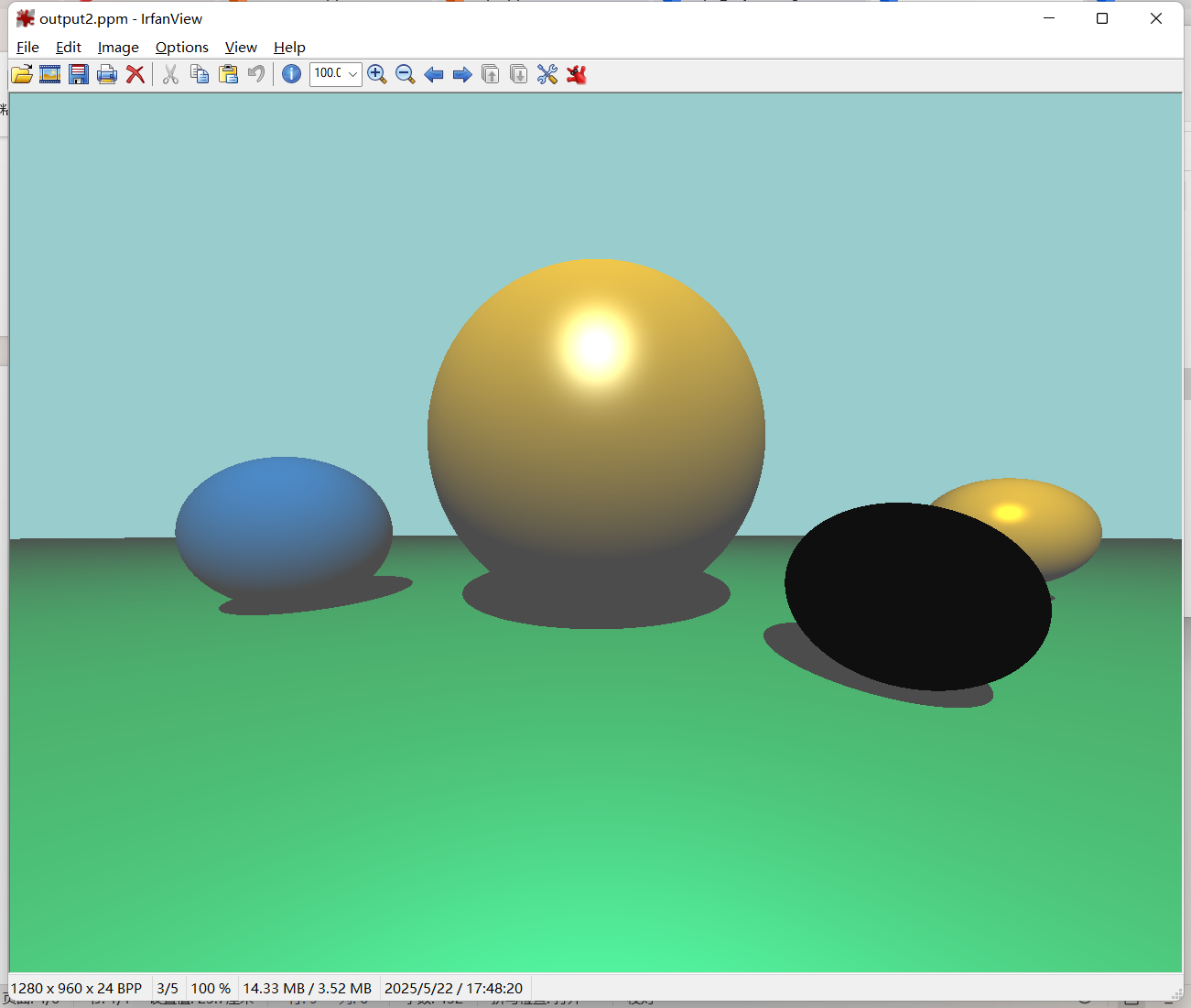
## 3.完成。在RayTracer.hpp的函数

## 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。

操作流程：补充RayColor函数，遍历场景中的每个物体，检测是否与shadowRay相交，如果是，则修改isShadow为true。

关键代码截图：

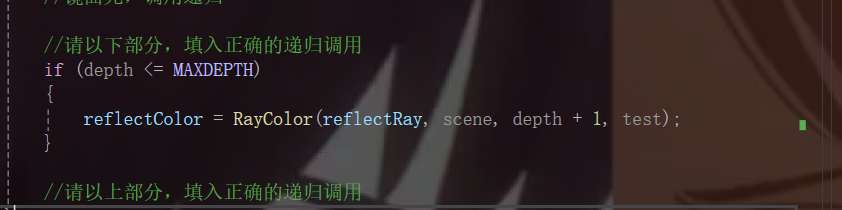
运行结果截图： 

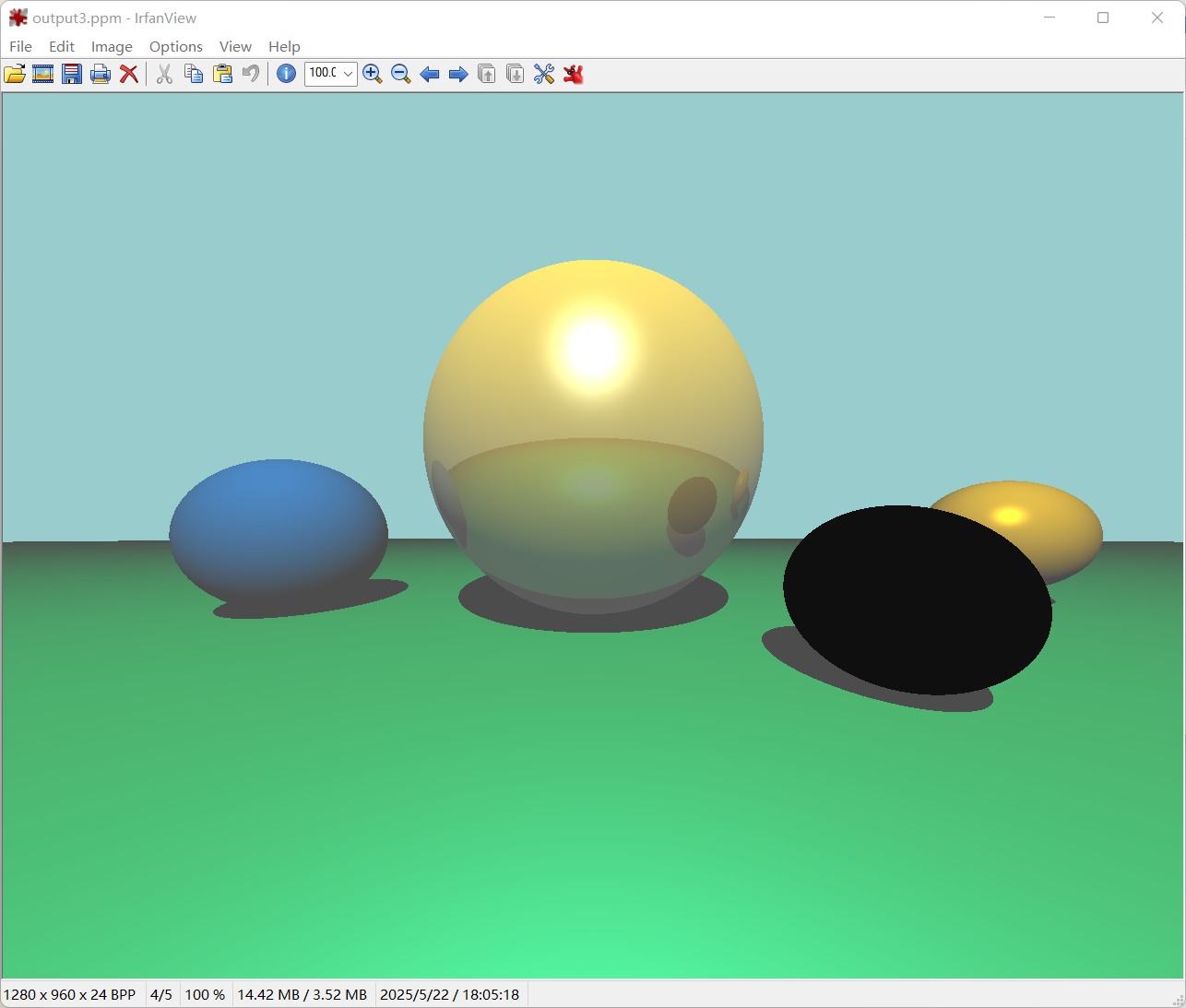
## 4. 完成。在RayTracer.hpp的函数

## 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。

操作流程：补充RayColor函数，使用reflectRay进行递归，将递归的值返回给reflectColor，递归层数+1。

关键代码截图：

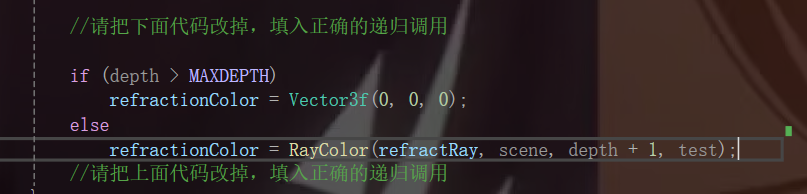
运行结果截图： 

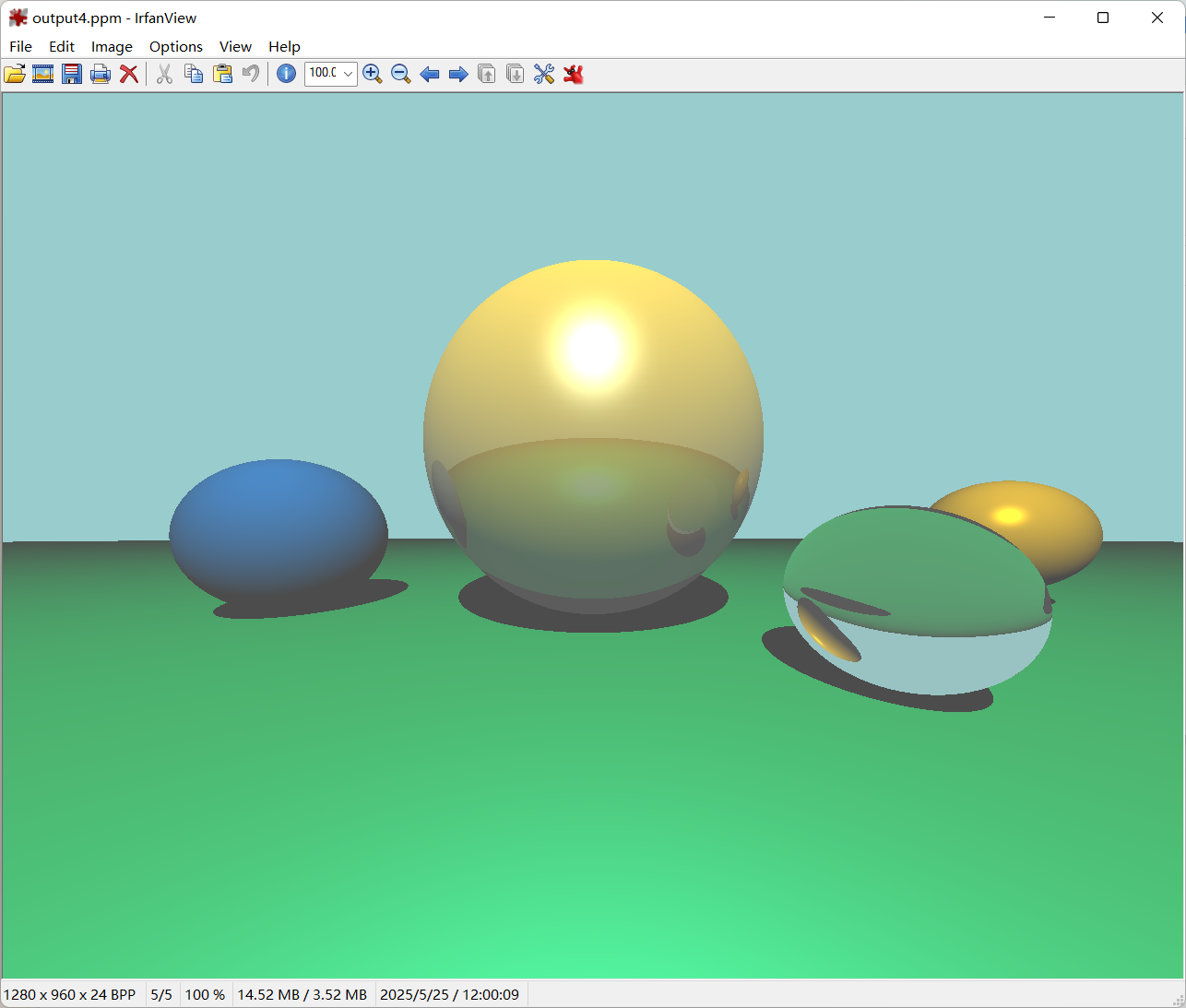
## 5.完成。在RayTracer.hpp的函数

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

## 加入完成折射的递归代码。

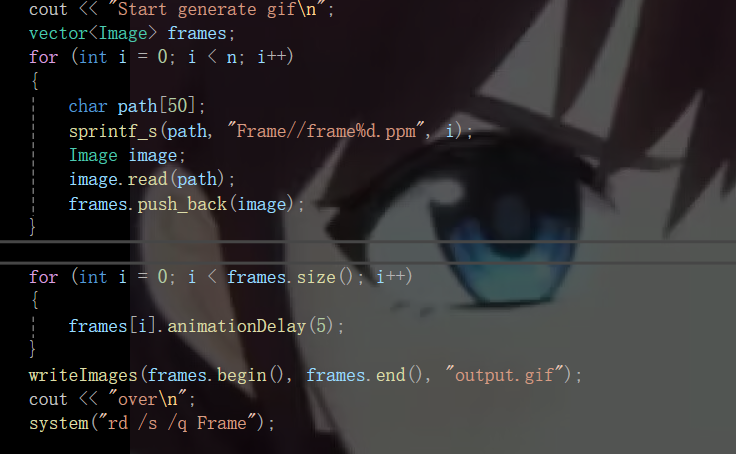
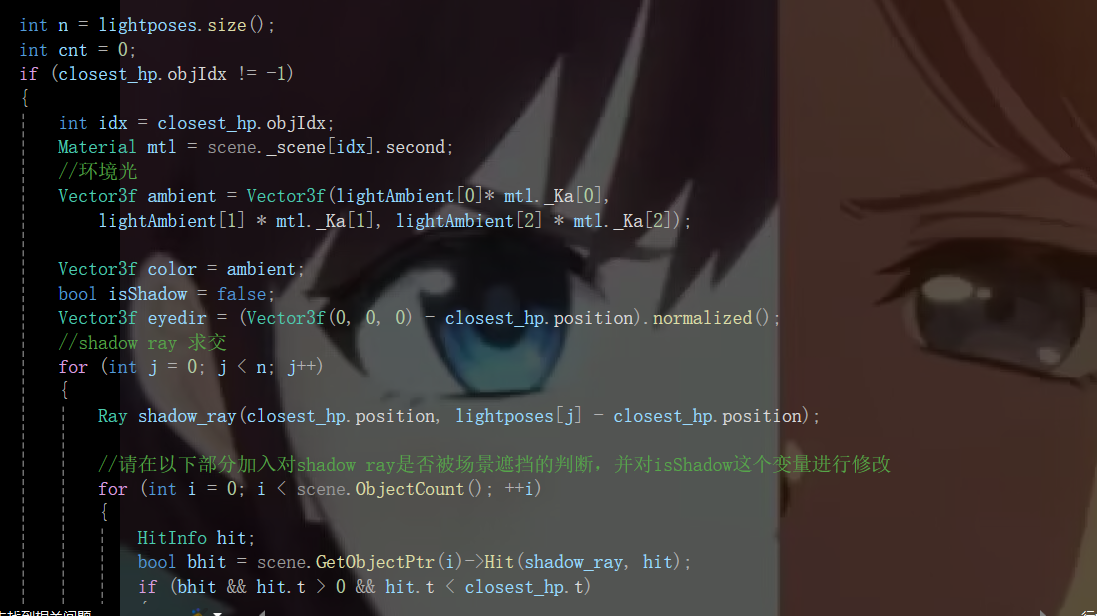
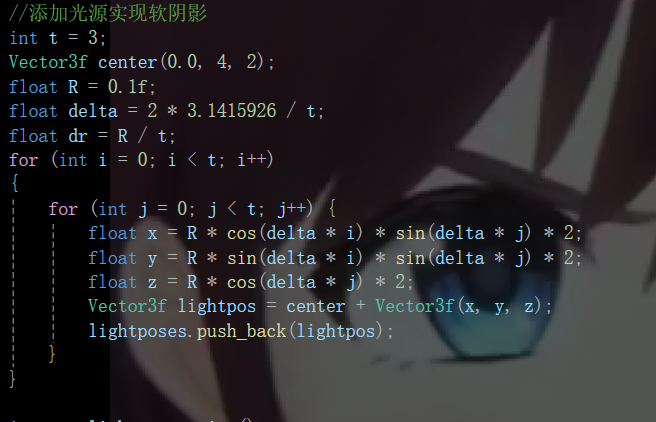
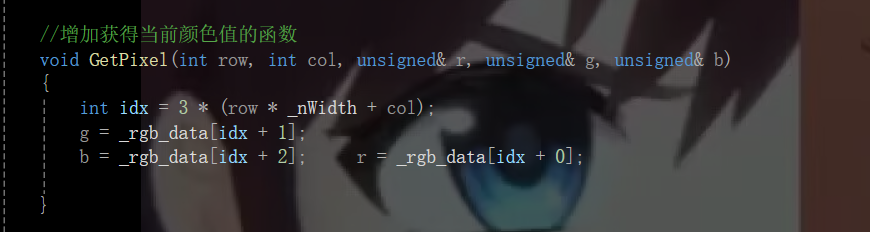
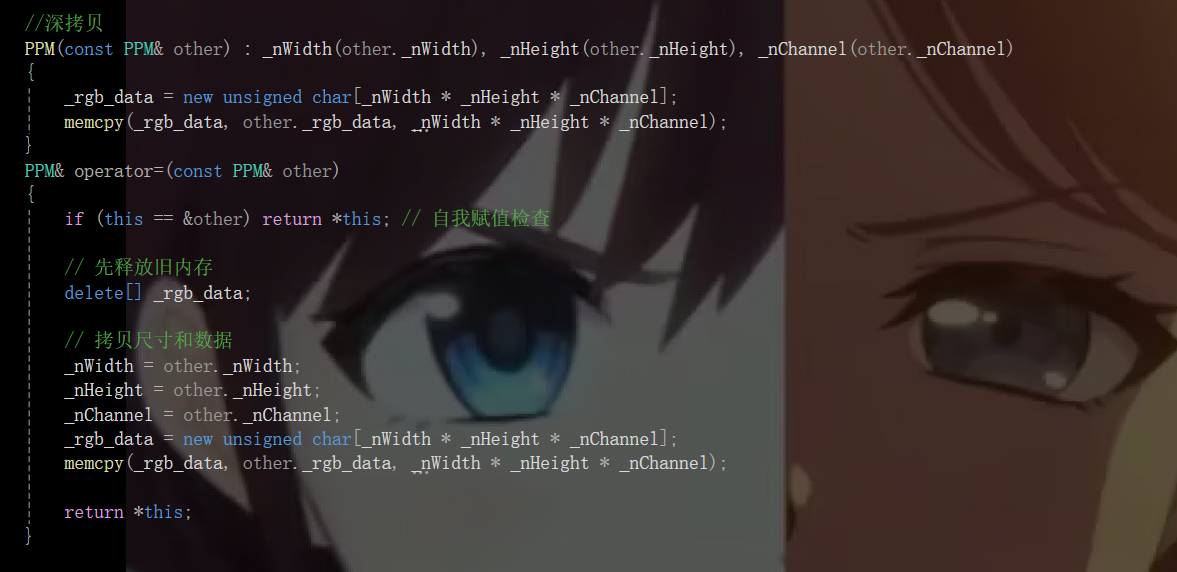
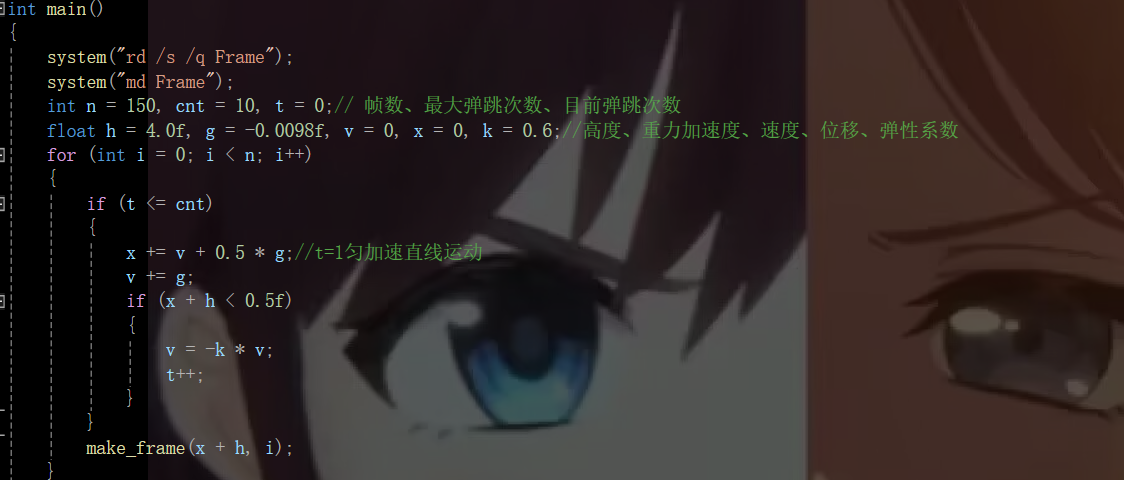
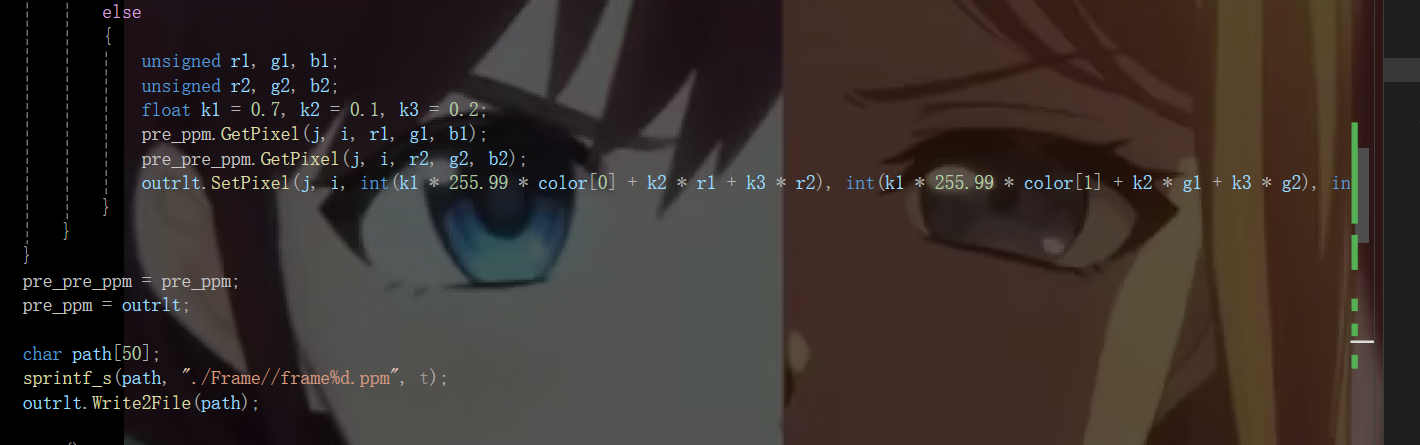
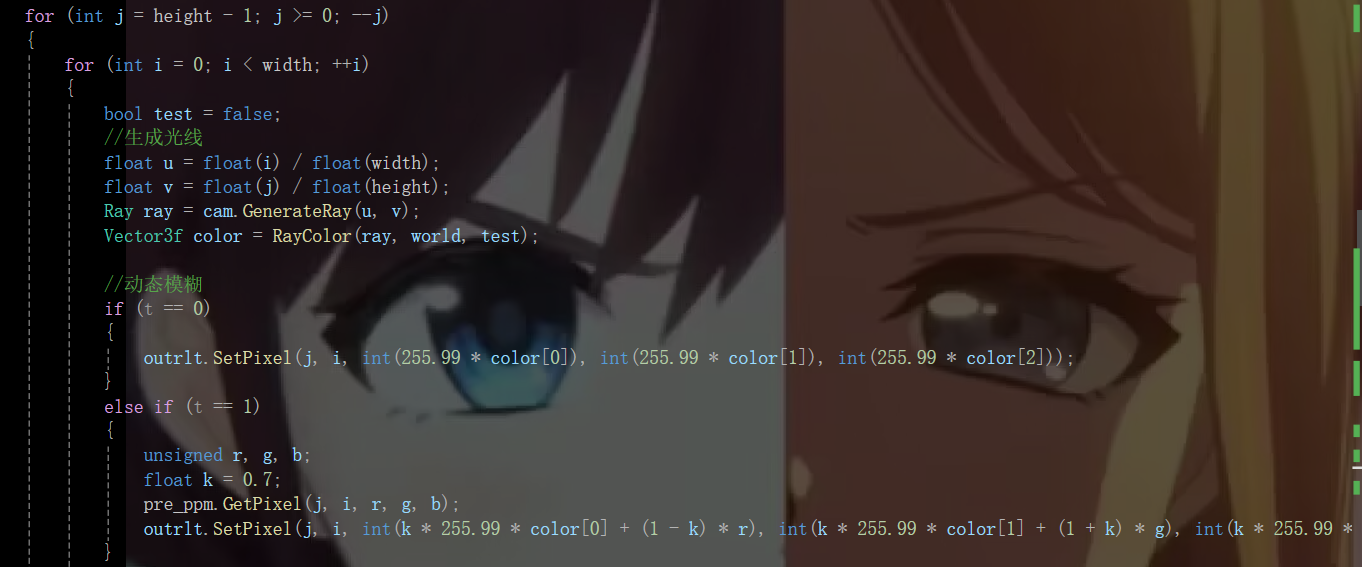
操作流程：补充RayColor函数，使用refractRay进行递归，将递归的值返回给refractionColor，递归层数+1。

关键代码截图：

运行结果截图： 

## 6.完成。 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。

操作流程：修改exp6.cpp的main函数，将定义球和生成光线的部分单独拿出作为一个新的函数make\_frame，通过参数h控制第一个球的y轴，参数t表示这是第几帧。在main函数中，创建一个临时文件夹Frame存储帧序列，使用物理公式进行递推，每帧相当于1的时间间距。在make\_frame函数中，为了实现动态模糊，新定义了两帧用于存储前两帧画面的PPM变量，当前帧画面的像素将是由三个画面进行线性插值得来的，在ppm.hpp文件中，新增了GetPixel函数用于获取前两帧的像素颜色，实现深拷贝并将画面比例改小，避免爆栈。在raytracer.hpp中，围绕原光源增加光源数量，最终阴影是每个光线产生阴影的加权和，从而实现软光源。配置Magick++.h，读取临时文件夹中的帧序列，制作成gif动画输出。

关键代码截图：

运行结果截图：