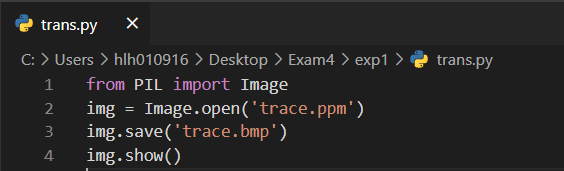
实验报告四

姓名：胡力杭

学号：2019K8009926002

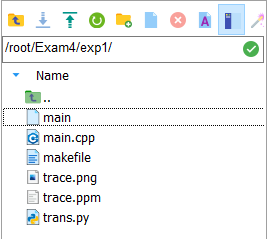
**代码明细**：

基本实验的算法是遍历每个像素点，进行光线追踪算法的实现。这种算法计算复杂度比较高，需要的时间比较长，但是对于我们学习如何实现光线追踪的代码有奠定性的帮助。值得注意的是，老师给的参考代码中输出了ppm格式，我们使用python的PIL库进行格式转换，具体的代码是（参考了csdn上的代码）

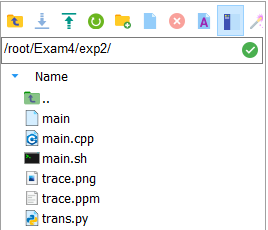


代码和执行脚本放置于/root/Exam4/目录，exp1为光线追踪基础实验，需要使用python进行相应的格式转化，exp2为进阶实验，添加了相机控制和正方体的输入。

exp1目录包含main.cpp、可执行文件main、makefile文件、用于转化格式的trans.py、得到的图片trace.png和trace.ppm。



exp2目录包含main.cpp、可执行文件main、可执行脚本main.sh、用于转化格式的trans.py、得到的图片trace.png和trace.ppm。



**环境配置：**

无

**程序编译及运行命令：**

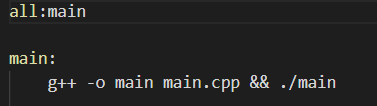
exp1：

①使用make或者make all命令进行编译执行

②在其他linux机器上使用python3 trans.py进行格式转化，导出png。（尝试在给的机器中安装python环境，但是好像失败了）

③使用make clean 删除可执行文件main和.o后缀的文件

在makefile中包含编译命令：



和清除命令：



exp2：

使用main.sh脚本进行编译运行

一共需要输入三个参数，一个是宽度像素数，一个是高度像素数，一个是视角远近

可执行脚本中具体的命令为：

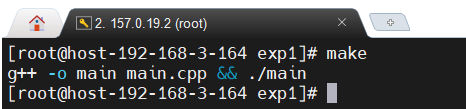
g++ -o main main.cpp && ./main

**实验效果（效果展示+操作说明）：**

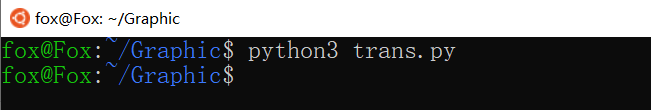
exp1：

光线追踪实验：

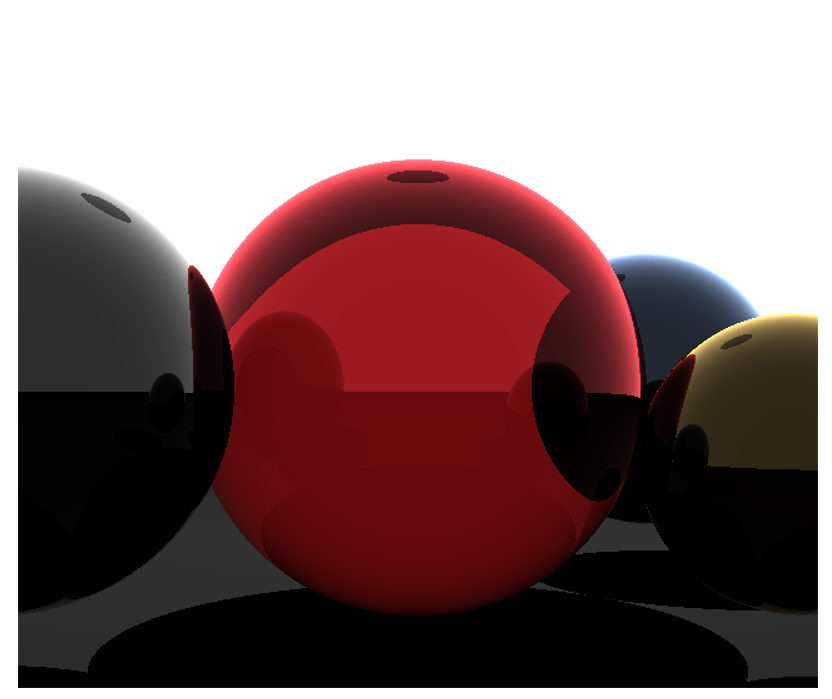
输入下图所示的指令得到一张ppm格式的图片



在WSL中输入如图所示的指令将ppm格式转化为png格式。



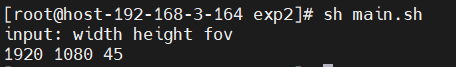
打开png图片查看效果，发现和我们需要的一致。



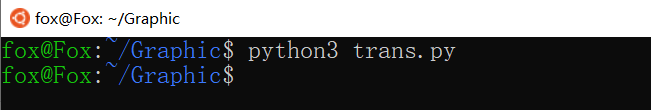
exp2：

进阶实验：

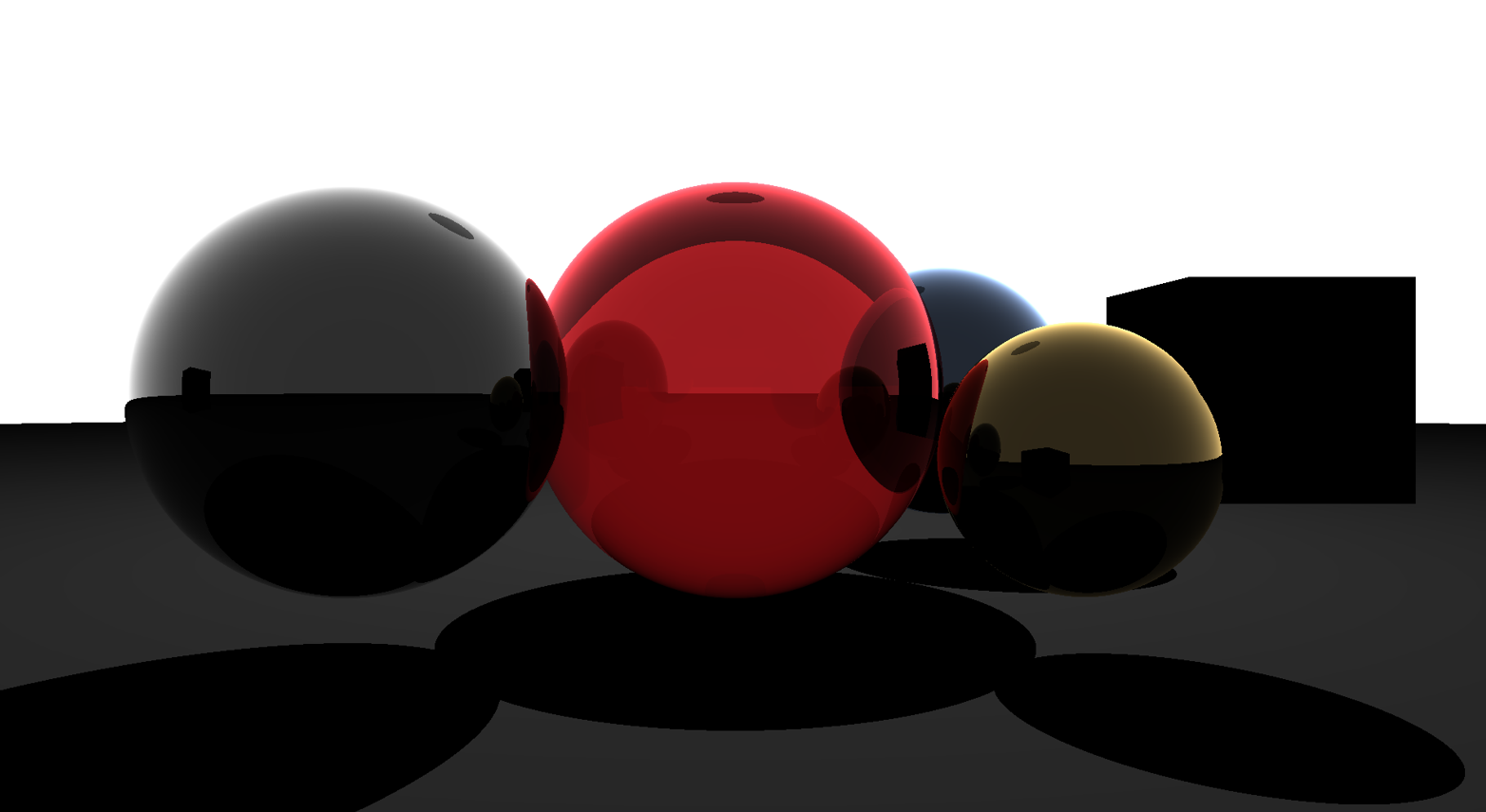
输入下图所示的指令得到一张1920\*1080的图片。



在WSL中输入如图所示的指令将ppm格式转化为png格式。



打开png图片查看效果，发现已经添加上了正方体，并且相机位置也发生了变化。



从具体的属性也可以看出图片的像素点确实增加了。

