

# OpenGL 项目文档

李朝文 U201517171 数媒 1501

## 一、IDEA

本人于九月份利用大量课余时间在学习 OpenGL 网站上学习了以三维变换、基础光照、模型加载为核心的前四个章节。本项目基本在国庆长假期间完成。当时正值中秋佳节，明月给了我创作的灵感，于是以中国传统神话故事“嫦娥奔月”为题材，诞生了这个“巨人奔月”的 OpenGL 项目。

## 二、环境配置

软件配置：

VC++2015 运行库(x64)，glfw 运行库，glad 库，Assimp 模型导入库，Windows7 及以上版本操作系统。(尽量在 Win10 上运行)

硬件配置：

兼容 OpenGL3.3 及以上版本的显卡。

注：提交的文件夹中已包含所有项目运行需要的 dll 文件，在没有配置以上运行库的电脑中也能运行，但不同电脑上可能会有不同的运行结果，请以提交的运行截图为准。

经本人多地测试，

- 1.同学和朋友的 Win10 版本电脑上正常运行程序；
- 2.教学楼的 Win7 版本电脑上可以运行程序，但运行结果略有不同；
- 3.图书馆的机房由于是 XP 版系统无法运行程序；
- 4.工程中心的机房由于显卡只支持到 OpenGL3.1，无法运行程序；

## 三、场景介绍

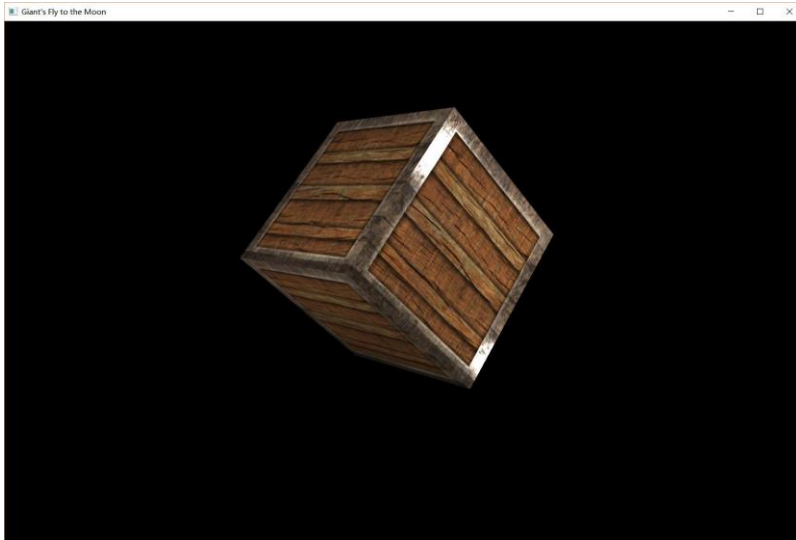
按键操作指南：(所有按键需要在英文模式下键入)

- ①WSAD 控制第一人称位置(摄像机位置)前后左右移动。按 Q 可减速，按 E 可加速，按下时间越长加减速幅度越大。
- ②移动鼠标控制第一人称视角方向(摄像机镜头方向)，鼠标滚轮控制镜头的远近；
- ③按 O 打开箱子；
- ④白天的场景中可以操纵巨人移动：↑↓←→方向键分别控制巨人在 xy 平面内上下左右移动，H 和 N 分别控制在 z 轴方向后退和前进。
- ⑤ESC 键退出程序。

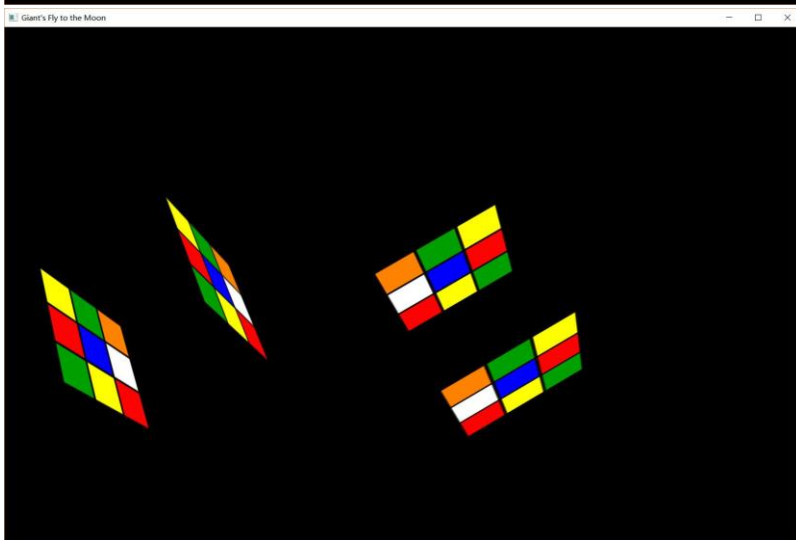
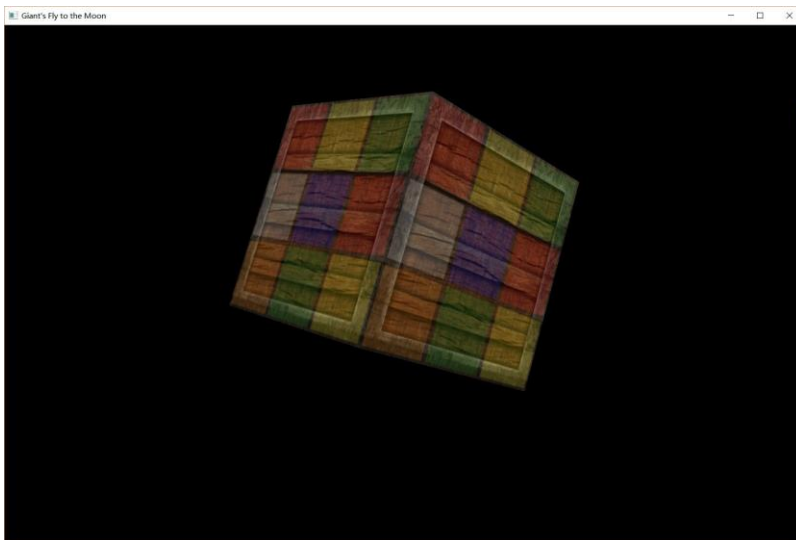
场景描述：

运行 exe(由于加载模型需要等待 10 秒左右出现场景)，进入场景 1，发现自己困在一个正在旋转的箱子里，按 WSAD(将输入法切换为英文)移动摄像机位置走出箱子，配合鼠标移动调整视角可在各

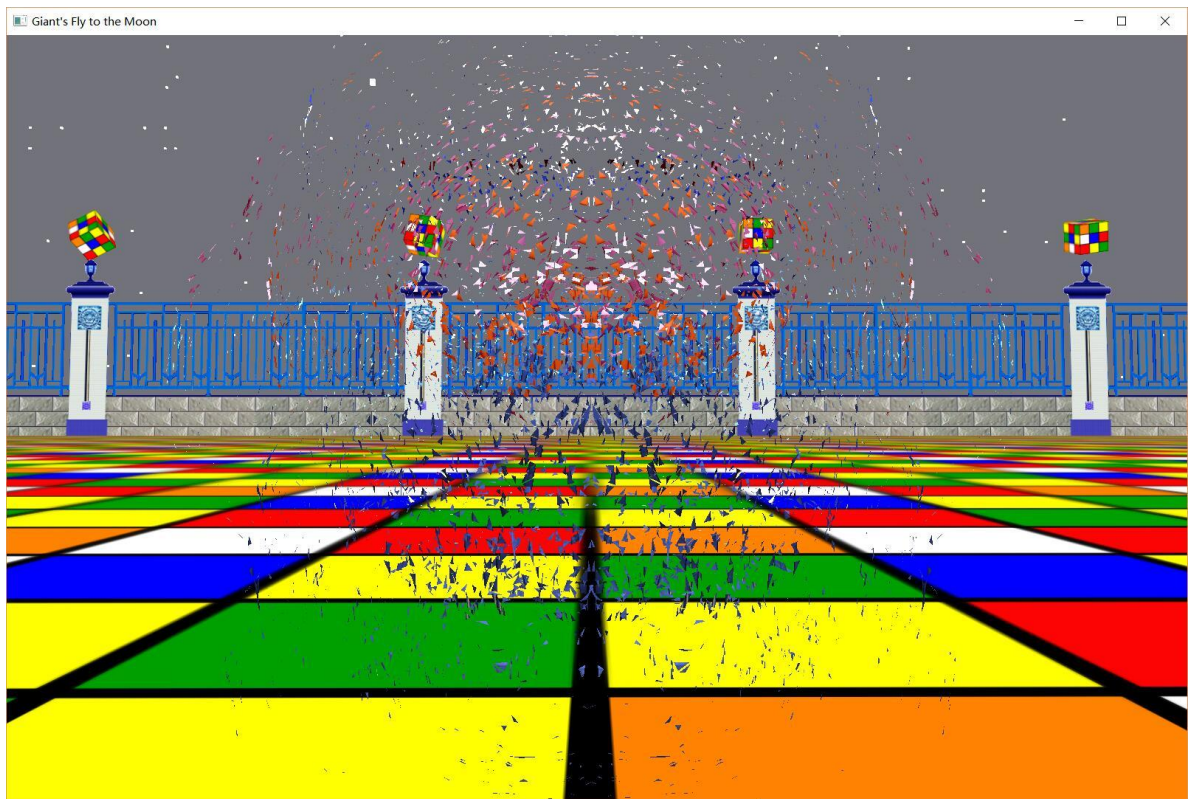
个角度观察箱子，发现箱子受到环境光和点光源的照射，靠近箱子发现表面有手电筒(聚光灯)效果。除此之外，发现箱子的金属边框比木头箱体对光的反射率更高(更加明亮)。



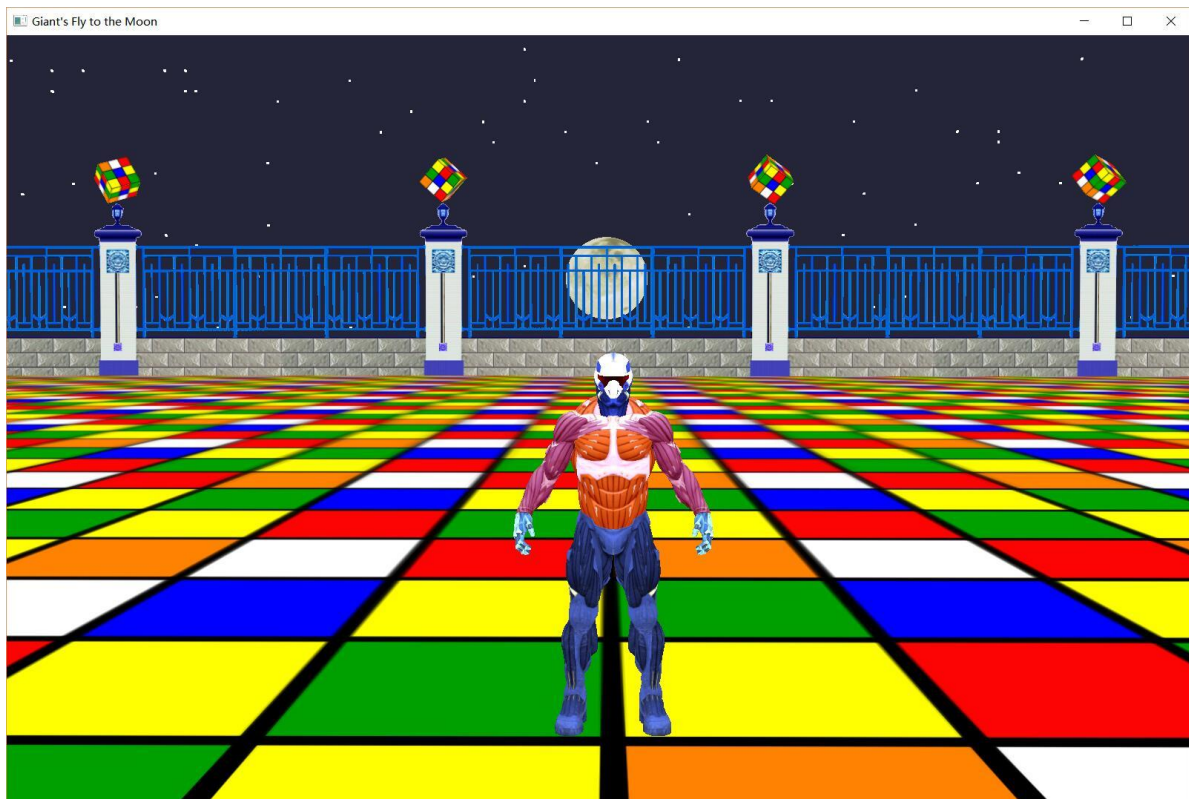
按下 O 键，打开箱子(Open)，可以看到箱子表面的纹理发生渐变，直至变成一个魔方，最后魔方分裂，画面变白，魔方内部的世界即将呈现。

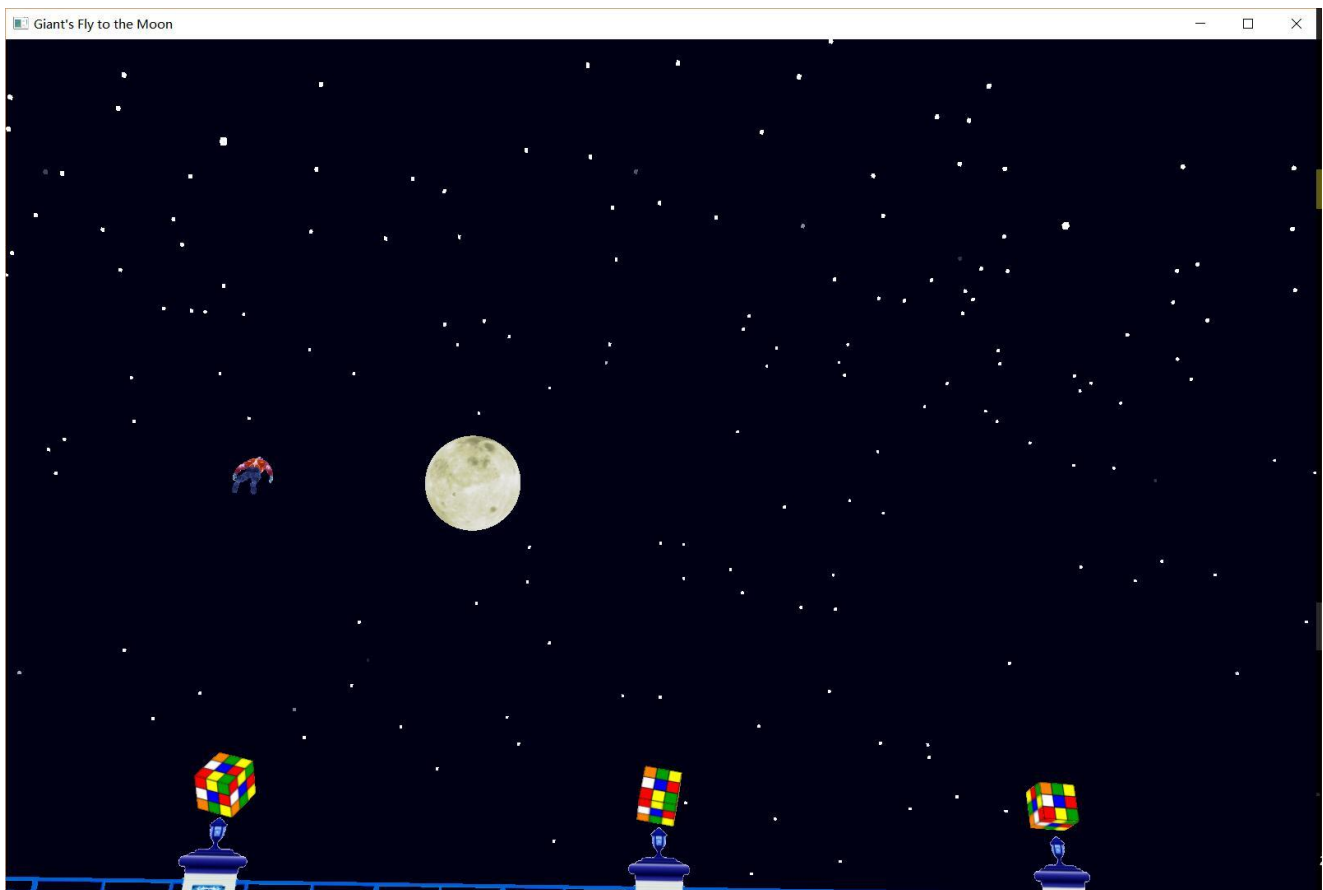
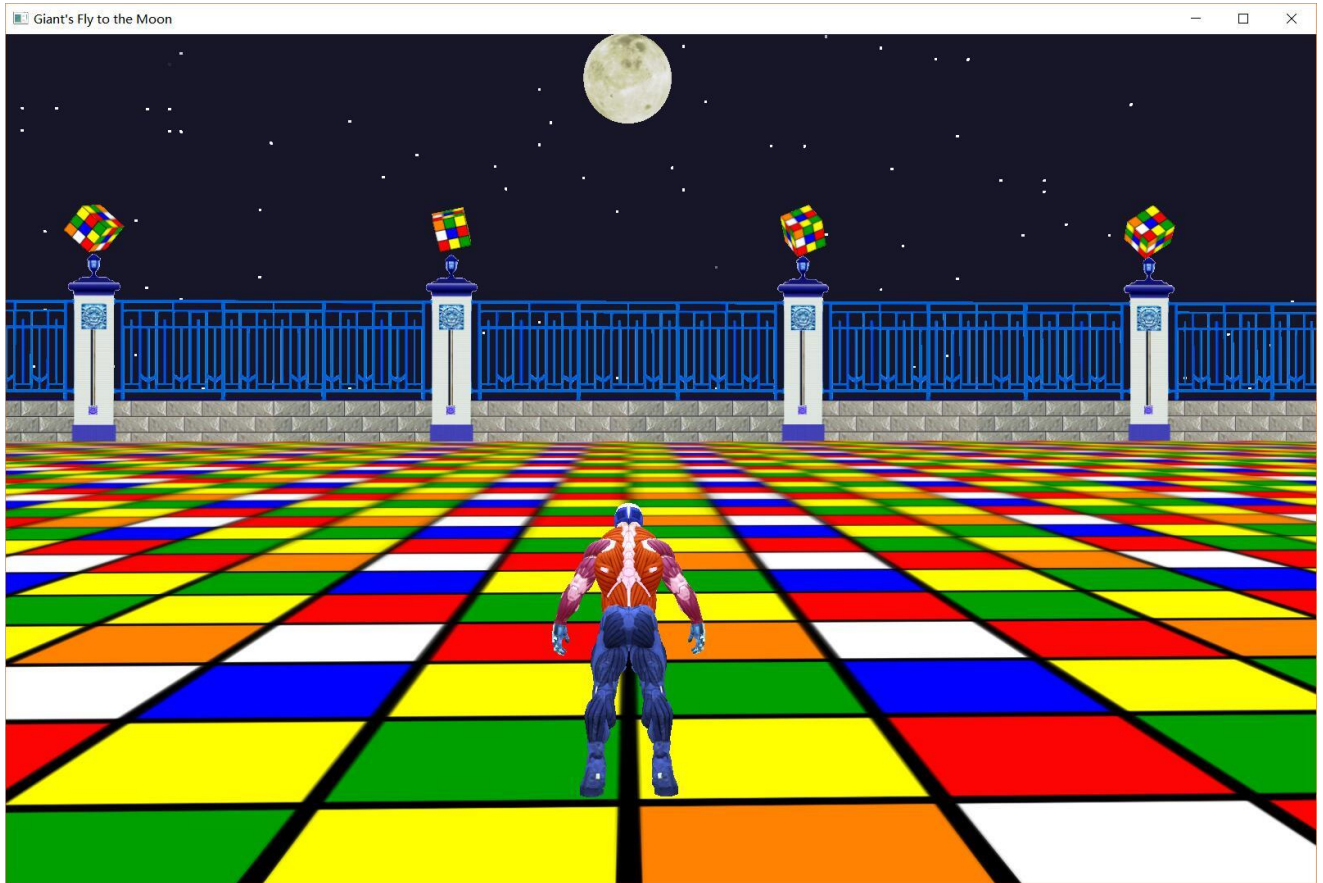


此时场景 1 切换到场景 2，发现自己站在铺满魔方砖块的地面上，四面都是蓝色的围栏，围栏的每个立柱上方都有一个魔方在旋转。夜空中充满了闪烁的星星，并且不断下着流星雨。同时眼前出现一堆彩色的碎片，它们渐渐地聚合，拼凑成一个魔方巨人的形状。

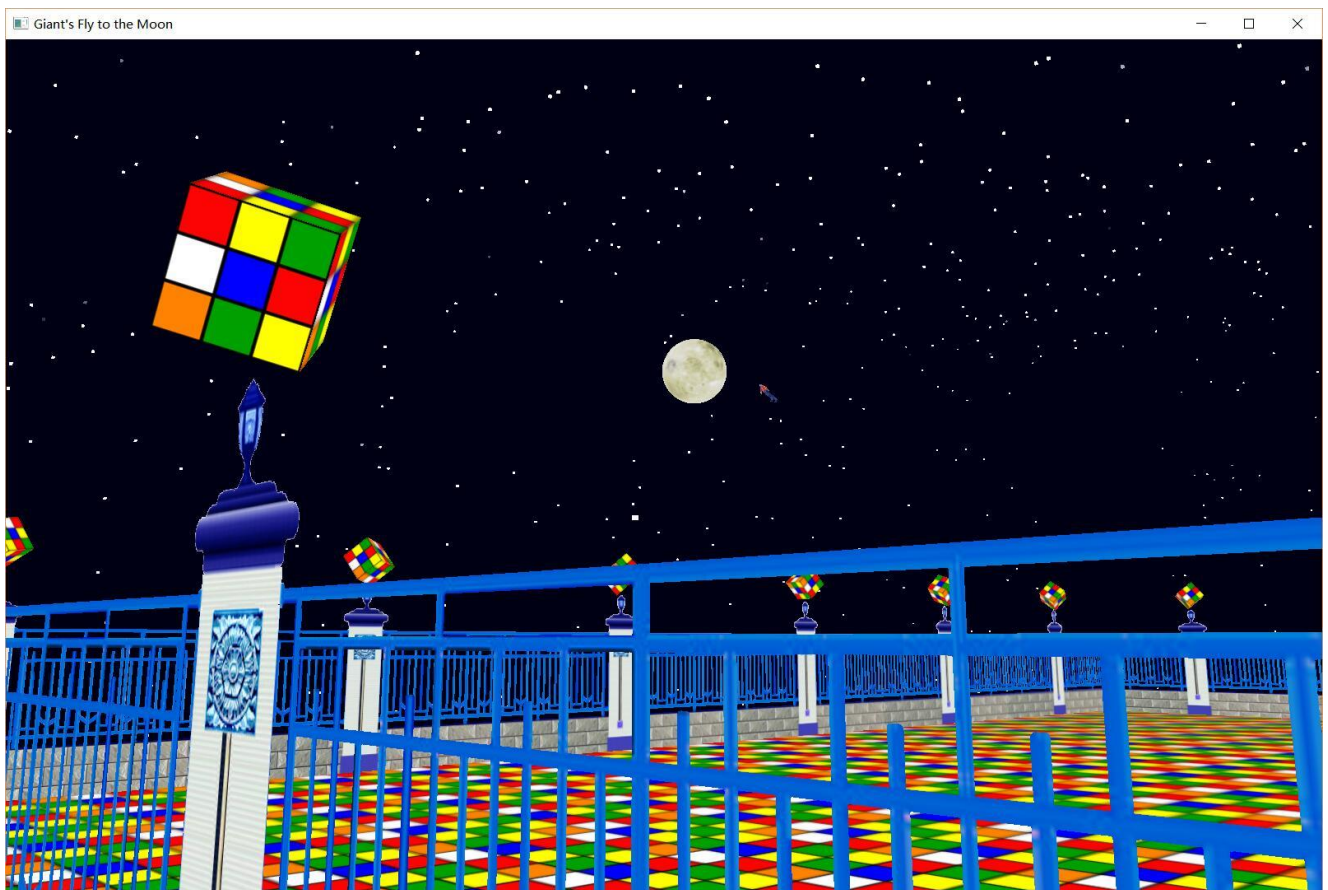
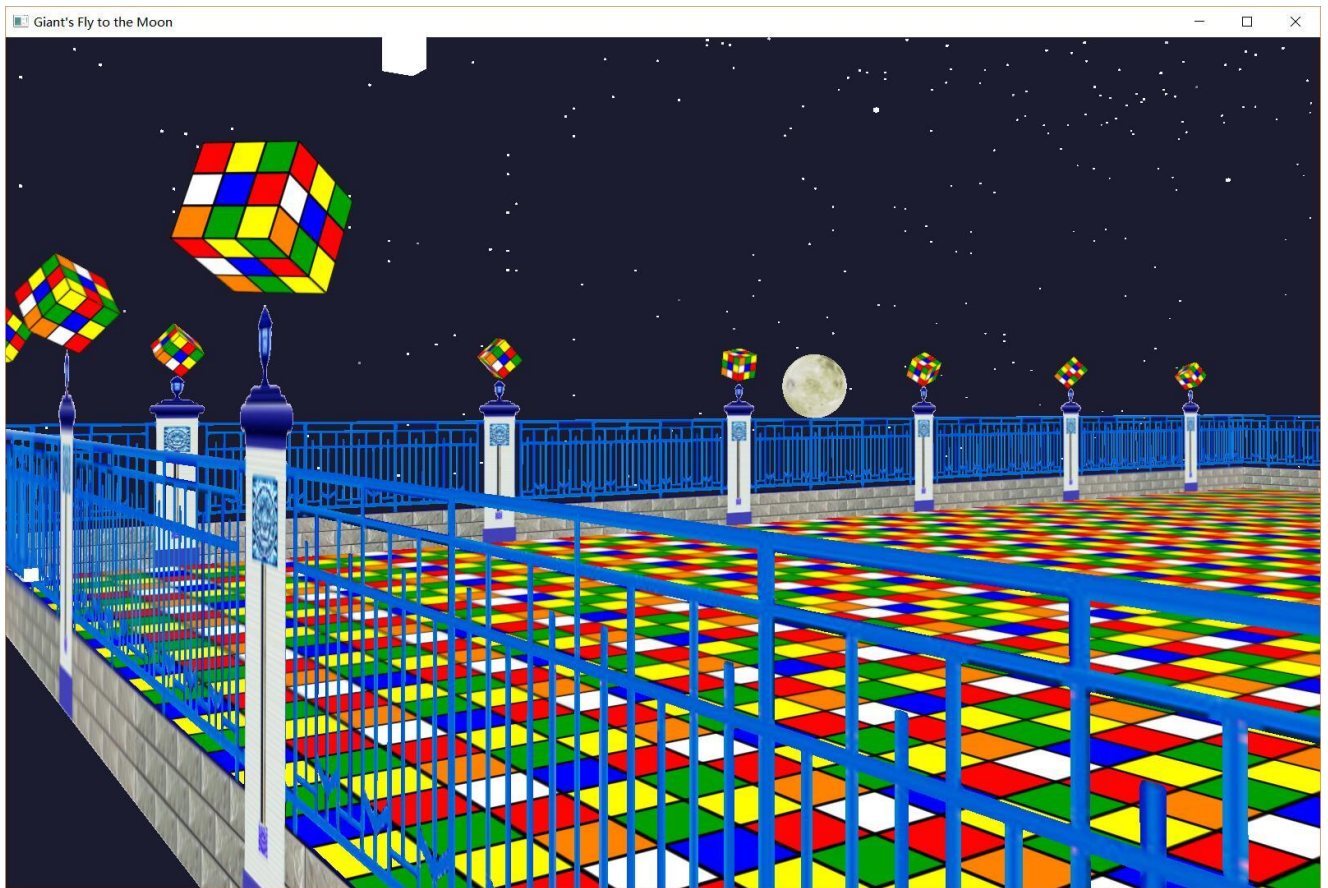


月亮开始升起，巨人沿着 y 轴旋转，转速先逐渐加快再逐渐变慢，最终巨人停止旋转并面向月亮、倾斜身体，起飞，沿着 Z 型路线飞向月亮。

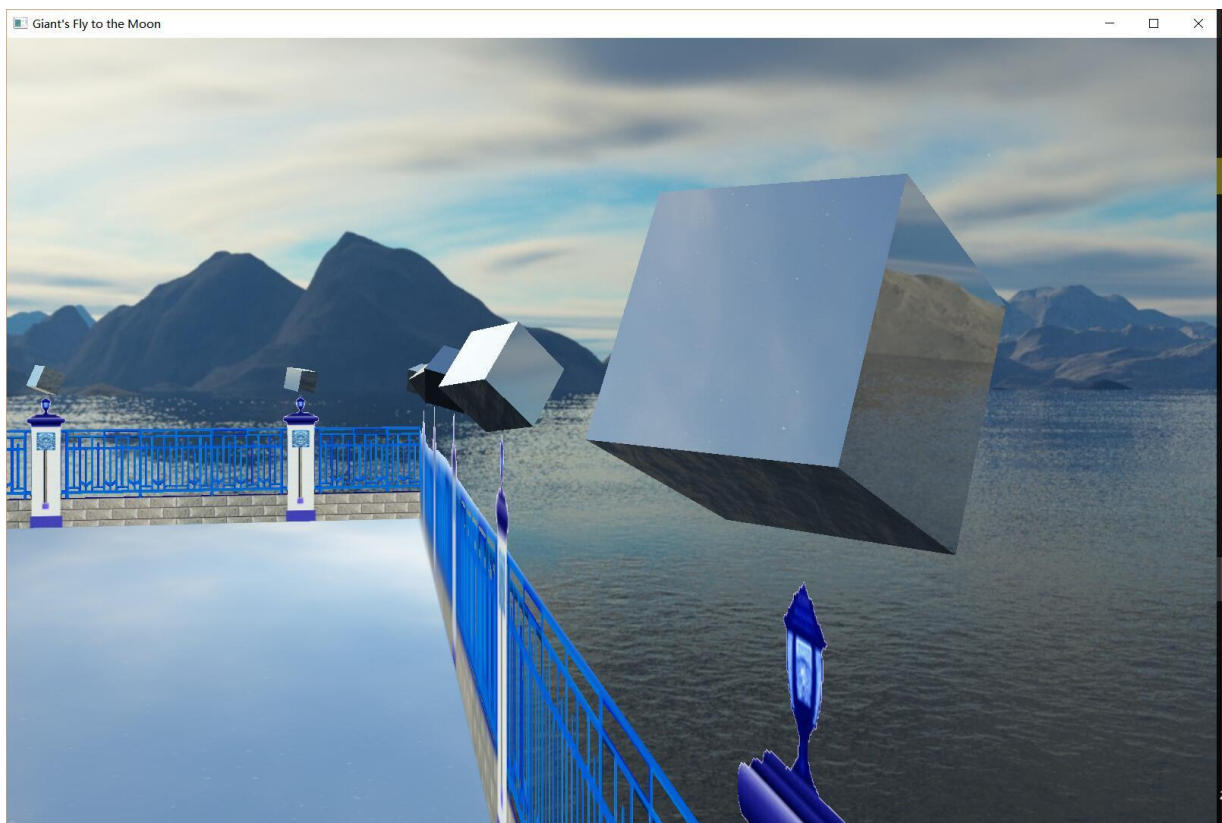








随着月亮升起，夜空的颜色也逐渐加深，过了一会月亮落下，夜空慢慢变亮，最后变成了白天，换上了景色优美的天空盒。此时巨人在原地，身体像是镀了一层金属，映射出天空的景色，围墙上旋转的魔方表面也都变成了立体的镜子，地板也映射出蓝色的天空。这时可用←↑↓→方向键和 H 键、N 键控制巨人在 xyz 空间的全方位移动。



## 四、得意之处

- 1.箱子变成魔方时纹理并非突然变化而是一个**渐变**的过程(箱体纹理逐渐变透明，魔方纹理逐渐显现)，并且停止旋转时应用了**几何着色器**，产生了**爆破**的效果；
- 2.巨人的身体由分裂的碎片逐渐聚合的效果也是通过几何着色器完成的，是**爆破**的逆效果，聚合的速度由**函数控制**，为了产生合适的速度经过了多次调试最终选用了  $\sin^3$  函数。
- 3.巨人的旋转是先变快再变慢，这通过以当前时间为自变量的**二次函数**来实现。为了要求在规定时间内要恰好旋转结束，编程时需要通过计算给二次函数设置合适的参数。
- 4.巨人飞天、月亮升起、天空变暗，三个看似独立的变化在时间上却相互关联，此时又需要用大量 if 条件句来控制变化发生的时序。各种变化所用的函数，它们的参数一方面需要调整参数使各自的动作处于正确的时间，一方面它们相互之间还要**协调参数**来使整体的变化正确呈现。
- 5.夜空的效果没有使用天空盒，可以说是用着色器“画”出来的天空，繁星位置储存在**随机数数组**内，程序运行时随机生成，**闪烁效果**通过改变星星颜色实现，大致方法是在渲染的 while 循环中设置了一个变量，for 循环中每绘制一个星星该变量自增并且达到一定值再清零，在星空颜色的基础上通过该变量继续增加星星颜色的 RGB 值，从而使闪烁效果参差不齐、循环往复。**流星雨**的效果也是通过随机数组随机产生起点并以向下 45°角方向移动星星。
- 6.白天时，魔方表面和巨人身体都采用**环境映射**来产生镜面效果，并加入**对巨人的控制操作**。
- 7.月亮和巨人都是通过**导入外部模型**实现。原来的想法是导入其他的模型，但在网上下载了许多模型均没能导入成功，于是只能充分利用教程中现有的资源(月亮和巨人都是教程里的示例模型)来创建场景，并且为了适应本案例，将原有的黑色巨人模型贴图用 **Photoshop** 改成了各种颜色，如同穿上了五颜六色的衣服。
- 8.代码中最大程度上实现**着色器的共用**。总共九个着色器程序，却只来自 6 个顶点着色器，6 个片源着色器和 1 个几何着色器。