

# 浙江大学



## 魔方模拟器个人分报告

授课教师：	袁昕
姓名：	钟睿昕
组别：	魔方模拟组
日期：	2023-7-10

# 1 分工任务

在这个项目中，我承担了不同阶段的任务。以下是我在每个迭代阶段的分工说明：

第一轮迭代中，我负责View层的开发工作和文档撰写。我专注于设计和实现魔方模拟器的用户界面，确保用户能够直观地与魔方进行交互，并提供良好的视觉效果。同时，我也负责编写详细的文档，以便团队成员之间的沟通和交流，确保项目的顺利进行。

在第二轮迭代中，我负责ViewModel层和Model层的开发工作。我致力于优化魔方模拟器的性能和用户体验，通过优化数据处理和逻辑处理模块，使得模拟器可以高效地完成解魔方的任务。我也尝试对传统魔方算法进行改进和优化，以提高解魔方的效率和准确性。

在第三轮迭代中，我负责Common层和App层的开发工作。我专注于处理应用程序的常用功能和应用逻辑，确保模拟器的基本功能正常运行，主要是进行了宏的编写，为解魔方提供工具。我与团队成员紧密合作，协调各个模块之间的协作和交互，并确保代码的质量和可靠性。

## 2 设计思路

### 2.1 第一次迭代

第一次迭代的任务是View层，主要承担渲染工作。

#### 2.1.1 输入模块

输入模块负责处理用户输入，并将其加载到Viewmodel层进行处理。它涉及键盘、鼠标和字符回调函数的处理。鼠标回调函数负责捕获鼠标移动距离，并将其传递给应用层进行处理。键盘和字符函数负责处理用户输入的每个指令。

#### 2.1.2 窗口渲染模块

在这部分中，实现了窗口的初始化和魔方的显示功能。根据Common层的定义，可以构建一个ViewMatrix进行视角渲染。通过渲染类的实现，将每个魔方的面分开绘制，以确保程序代码的低耦合性。

#### 2.1.3 旋转控制

在View层中，我实现了RotAni (Rotation Animation)，用于控制魔方旋转操作。通过RotAni，我能够对旋转过程进行控制，并通过通知机制及时发送旋转状态的更新，以便实时渲染旋转效果。这样，用户可以直观地观察到魔方的旋转过程，增强了交互体验。

### 2.2 第二次迭代

第二次迭代的任务是ViewModel层和Model层，主要实现序列化和反序列化、添加Command指令数和实现新功能。

#### 2.2.1 序列化和反序列化

在Model层中，我实现了当前魔方状态的序列化和反序列化，以便于将其状态存储到文件中去。主要用到了Common层中的一些序列化宏进行解构。

## 2.2.2 Command功能添加

第二轮迭代主要是添加功能，包括随机打乱魔方、文件加载和存储以及重置魔方的功能实现。对于Viewmodel层来说，需要在之前的指令集基础上添加四个功能类来实现这些功能。

## 2.2.3 文件读写

在处理文件读写时，使用了C++的fstream库，可以直接对可执行文件同级目录进行添加和读取文件的操作。这样可以方便地将魔方状态存储到文件中或从文件中读取魔方状态。

## 2.3 第三次迭代

第三次迭代的任务是Common层和App层，主要是宏定义实现。

### 2.3.1 宏定义

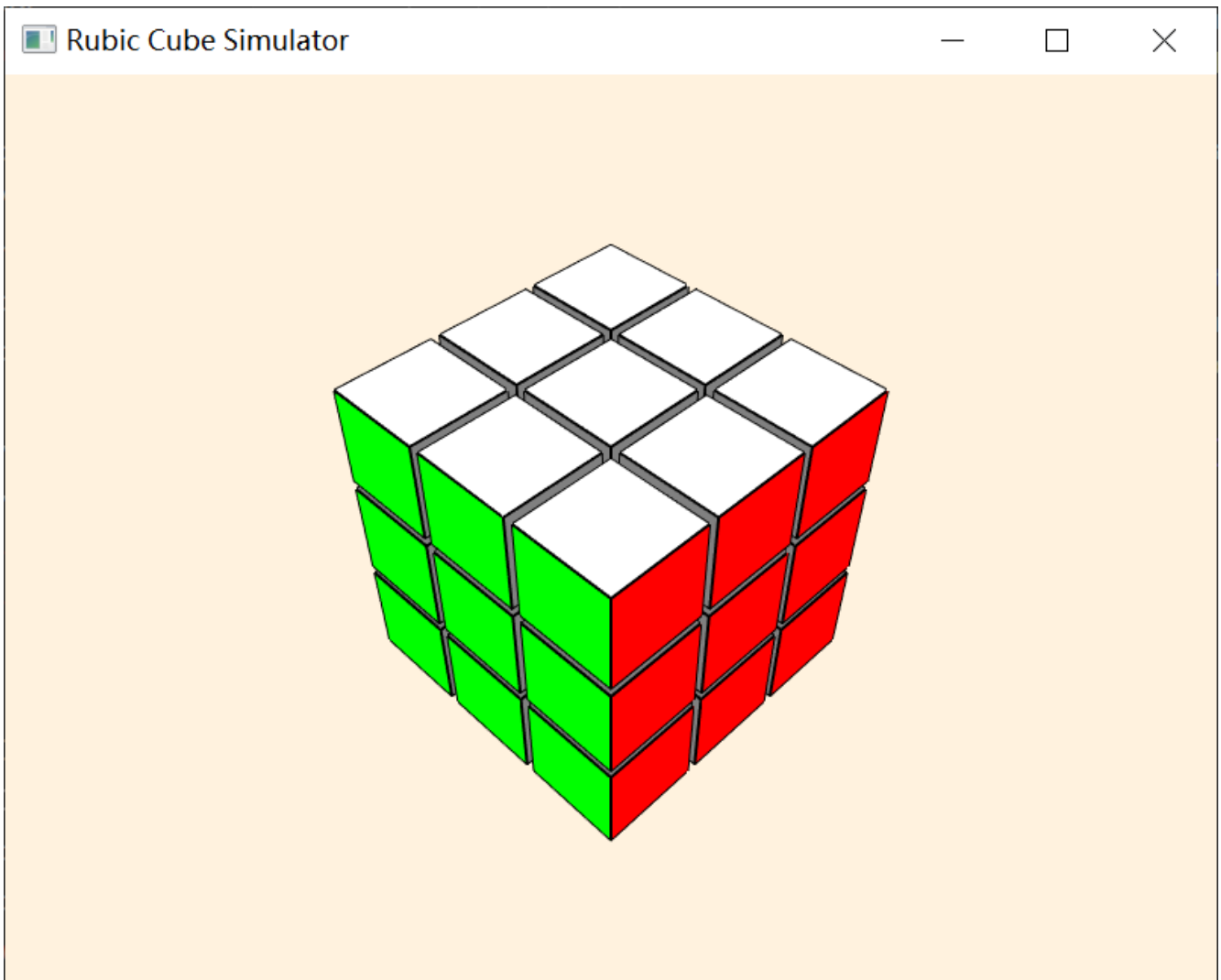
为了在后续开发中更加便捷，在这一次迭代中我添加了魔方构建判断的宏定义，用于判断每个面的颜色情况和找到对应的边或角。

```
1  #define IsColorMatch1(cube, color1) \  
2  (((GET_FRONT(cube) == color1) +\  
3  (GET_BACK(cube) == color1) +\  
4  (GET_LEFT(cube) == color1) +\  
5  (GET_RIGHT(cube) == color1) +\  
6  (GET_UP(cube) == color1) +\  
7  (GET_DOWN(cube) == color1)) == 1)  
8  
9  #define FIND_CORNER(subCubes, color1, color2, color3, stmtFLU, stmtFLD, stmtFRU, stmtFRD,  
10 stmtBLU, stmtBLD, stmtBRU, stmtBRD) \  
11 if (IsColorMatch3(subCubes[FLU_CORNER], color1, color2, color3)) \  
12 { \  
13     stmtFLU; \  
14 }\  
15 else if (IsColorMatch3(subCubes[FLD_CORNER], color1, color2, color3))\  
16 {\  
17     stmtFLD; \  
18 }\  
19 else if (IsColorMatch3(subCubes[FRU_CORNER], color1, color2, color3))\  
20 {\  
21     stmtFRU; \  
22 }\  
23 else if (IsColorMatch3(subCubes[FRD_CORNER], color1, color2, color3))\  
24 {\  
25     stmtFRD; \  
26 }\  
27 else if (IsColorMatch3(subCubes[BLU_CORNER], color1, color2, color3))\  
28 {\  
29     stmtBLU; \  
30 }\  
31 else if (IsColorMatch3(subCubes[BLD_CORNER], color1, color2, color3))\  
32 {\  
33     stmtBLD; \  
34 }
```

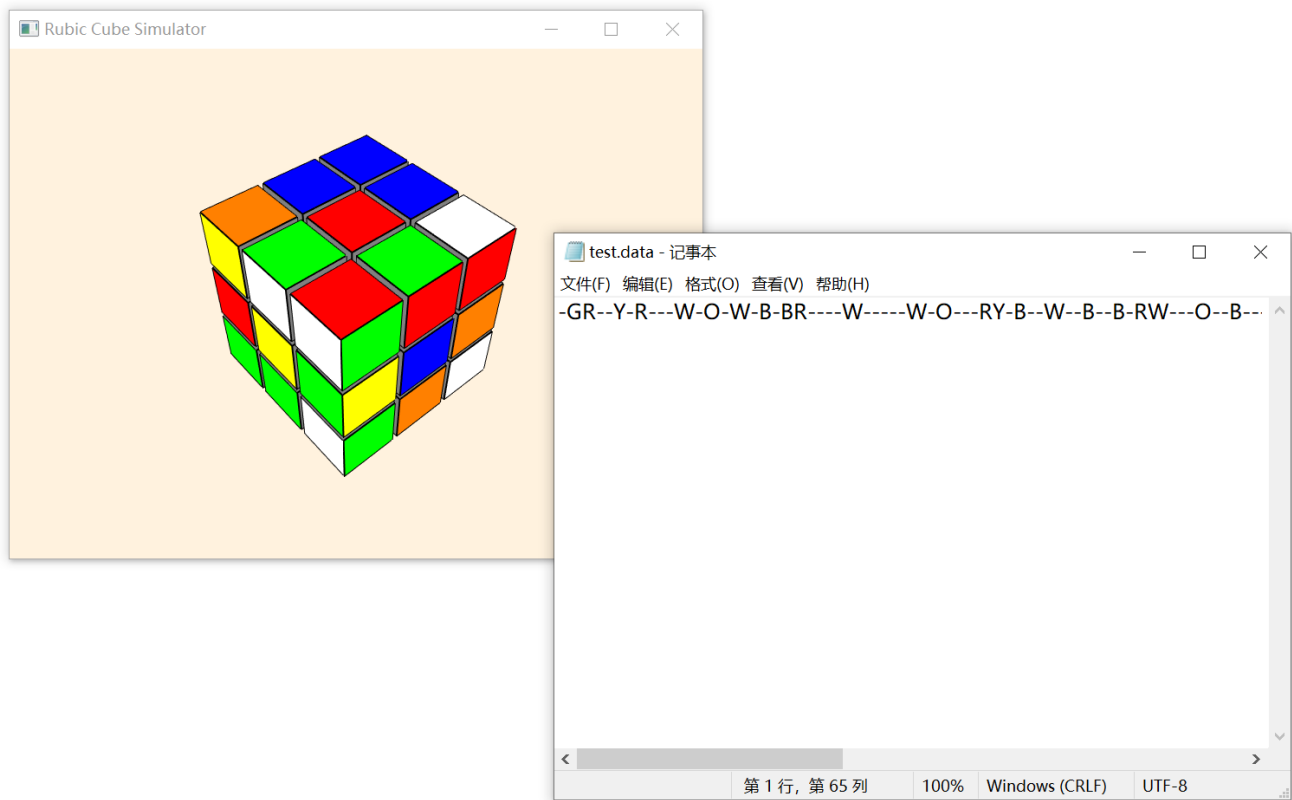
```
33 }\  
34 else if (IsColorMatch3(subCubes[BRU_CORNER], color1, color2, color3))\  
35 {\  
36     stmtBRU; \  
37 }\  
38 else if (IsColorMatch3(subCubes[BRD_CORNER], color1, color2, color3))\  
39 {\  
40     stmtBRD; \  
41 }
```

### 3 运行效果图

窗口渲染：



文件存储：



## 4 心得体会

项目中我担任了View层和文档撰写的任务，并在第二轮迭代中负责了ViewModel层和Model层的开发。通过参与这个项目，我收获了许多宝贵的经验和知识。

在第一轮迭代中，我致力于实现View层的功能，并负责了文档的撰写。通过开发View层，我深入理解了用户界面设计和实现的方法。我学会了如何与用户进行交互，并将魔方的状态以清晰的方式展示给用户。同时，通过撰写文档，我提高了自己的表达能力，并将项目的思路和功能清晰地记录下来，便于团队成员之间的沟通和协作。在第二轮迭代中，我转变了角色，负责了ViewModel层和Model层的开发工作。这个阶段使我更加熟练地运用了C++语言，并加深了对其语法和语言特性的理解。例如，我学会了如何定义纯虚类，以及如何进行指针的转化等技巧。这些知识的掌握使我能够更加灵活地编写代码，并提高了我的编程能力。