

浙江大学



魔方模拟器中期文档

授课教师：	袁 昕
组 长：	王伟杰
组 员：	钟睿昕
	戴 卿
组 别：	魔方模拟组
日 期：	2023-7-16

1 项目介绍

1.1 项目说明

- 项目名称：魔方模拟器
- 开发小组名称：魔方模拟组
- 开发小组成员：王伟杰（组长）、戴卿、钟睿昕

1.2 项目背景

魔方模拟器是一个基于C++编程语言开发的应用程序，旨在提供一个实时、交互式的魔方解谜工具。在现实世界中，解开魔方可能需要花费大量的时间和精力，而通过模拟器，用户可以在计算机上模拟魔方的各种操作和解法，更加便捷的操作魔方，并高效地学习和掌握解谜技巧。该模拟器允许用户在图形界面中操作虚拟魔方，模拟真实的魔方操作过程和解法，通过使用计算机算法和交互界面，帮助用户更好地理解和掌握魔方的解法过程。

1.3 项目成果

这是一个功能完善、易于使用的魔方模拟器应用程序。用户可以通过运行该程序，在计算机上模拟魔方的各种操作和解法过程，提高自己的解谜技巧，并学习魔方的解法。

该魔方模拟器为用户提供了一个功能齐全、交互性强的魔方模拟平台，能够实现以下功能：

1. 魔方模拟：提供一个魔方模拟器应用程序，实现在计算机上模拟魔方的各种操作和解谜过程。
2. 实时交互：确保模拟器能够实时响应用户的操作，提供流畅的魔方操作体验，使用户感觉像是在操控真实的魔方一样。
3. 支持多种操作命令：实现支持各种操作命令，如旋转、视角移动、打乱、复原、存储文件等，以满足用户对魔方的灵活控制需求。
4. 提供解谜演示：能够根据用户的输入解谜步骤，展示魔方复原的过程，并支持动画效果，以帮助用户学习和理解解谜的技巧。
5. 支持求解算法：模拟器支持模仿复原通用算法，以满足用户的需求和挑战。

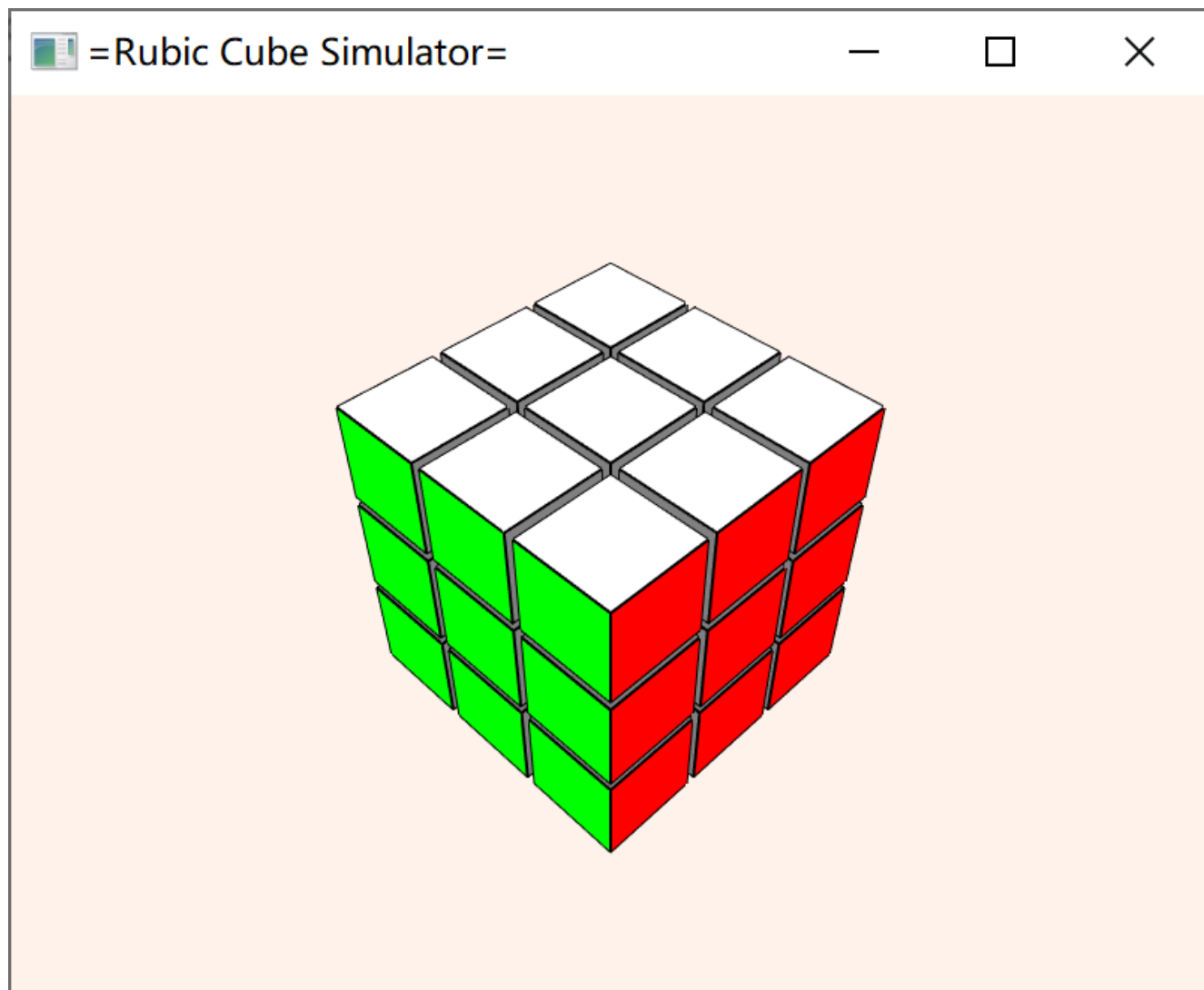
通过实现这些目标，该魔方模拟器将为魔方爱好者和初学者提供一个综合的学习和娱乐平台。用户可以通过模拟器的操作和演示功能，自由地探索和体验魔方解谜的乐趣，并逐步提高自己的解谜技巧。

1.4 项目用户

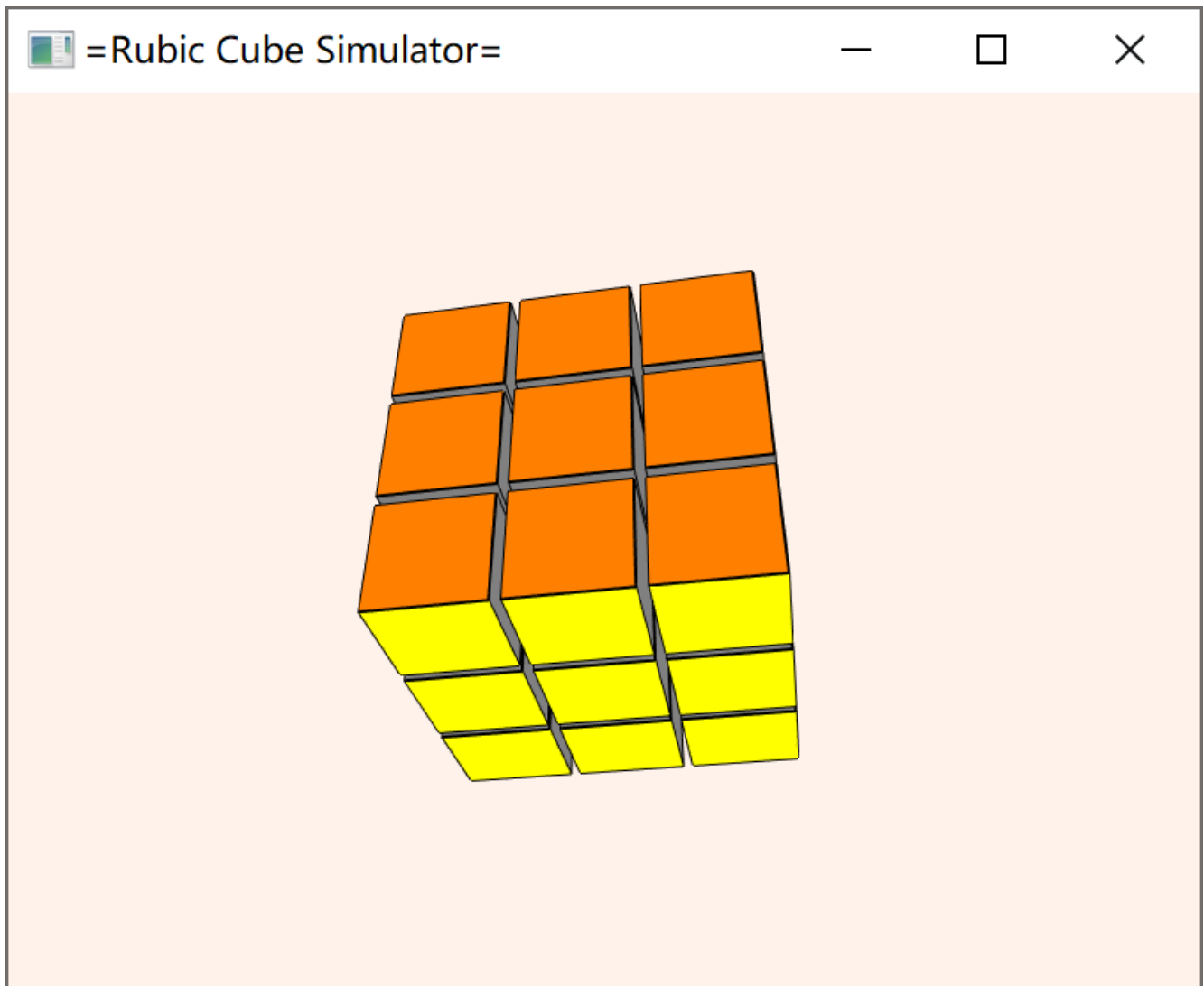
本项目面向对象为各层次水平魔方爱好者，包括魔方初学者、解谜爱好者、专业魔方选手等。

2 项目效果

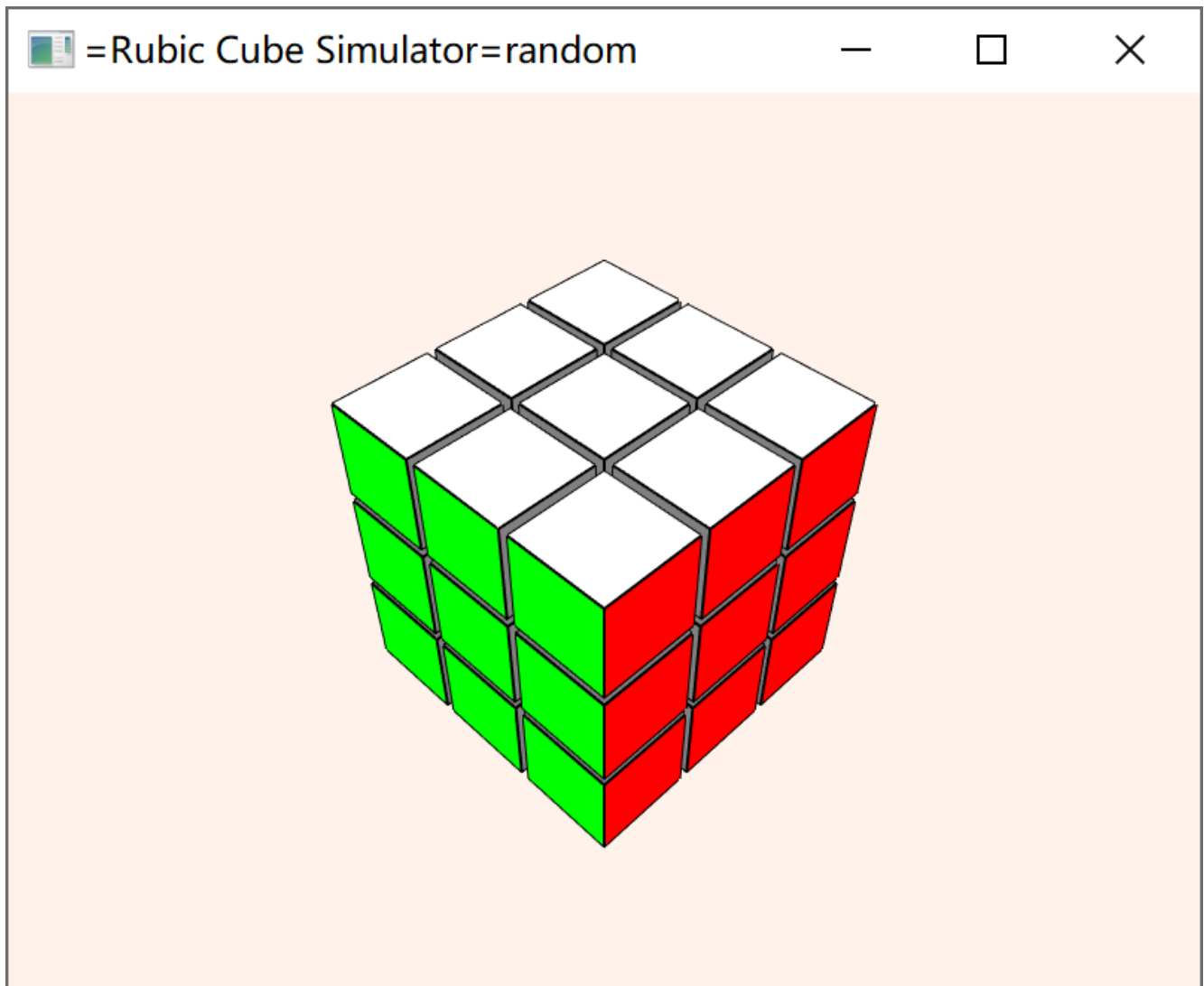
2.1 初始界面



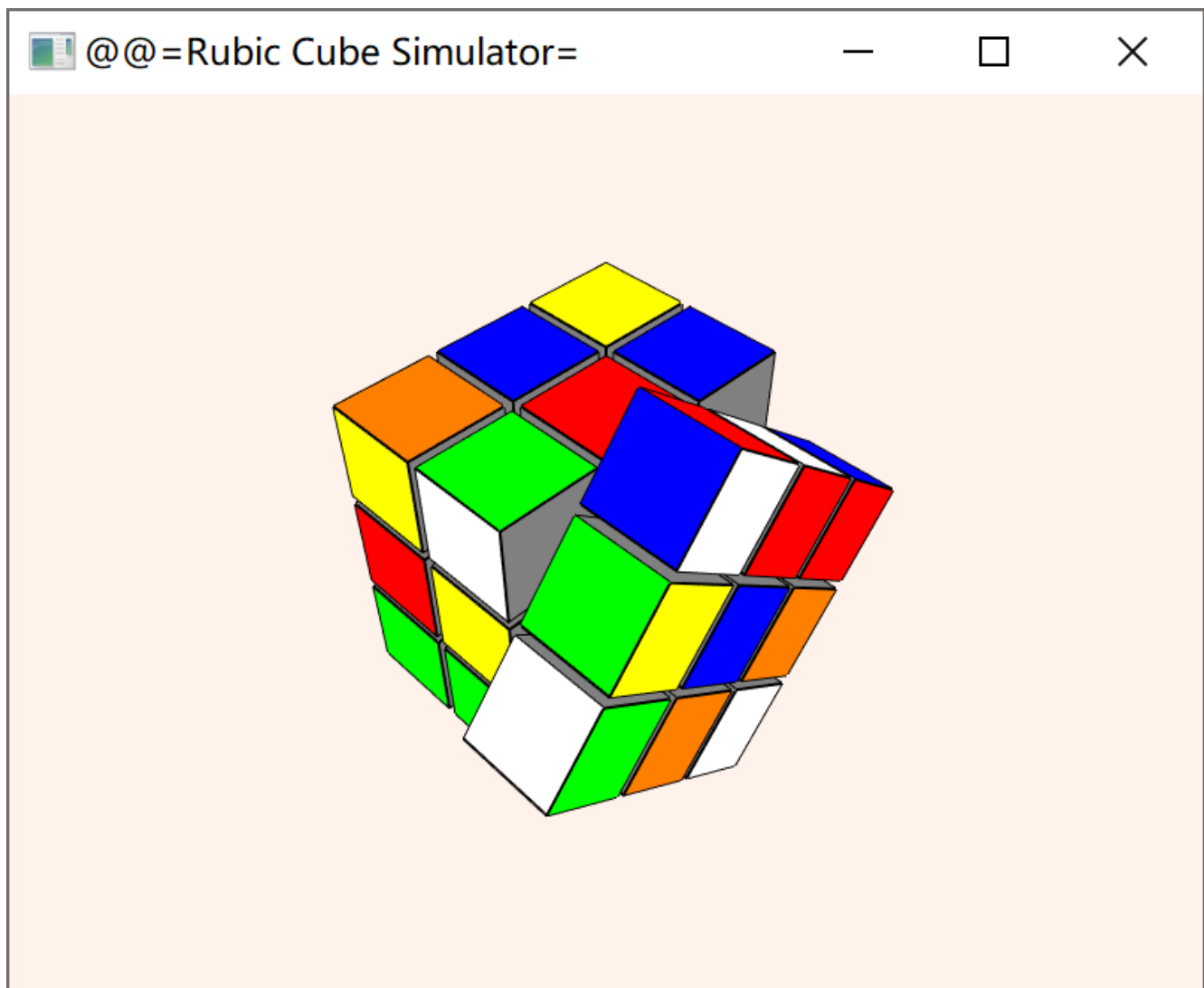
2.2 移动视角



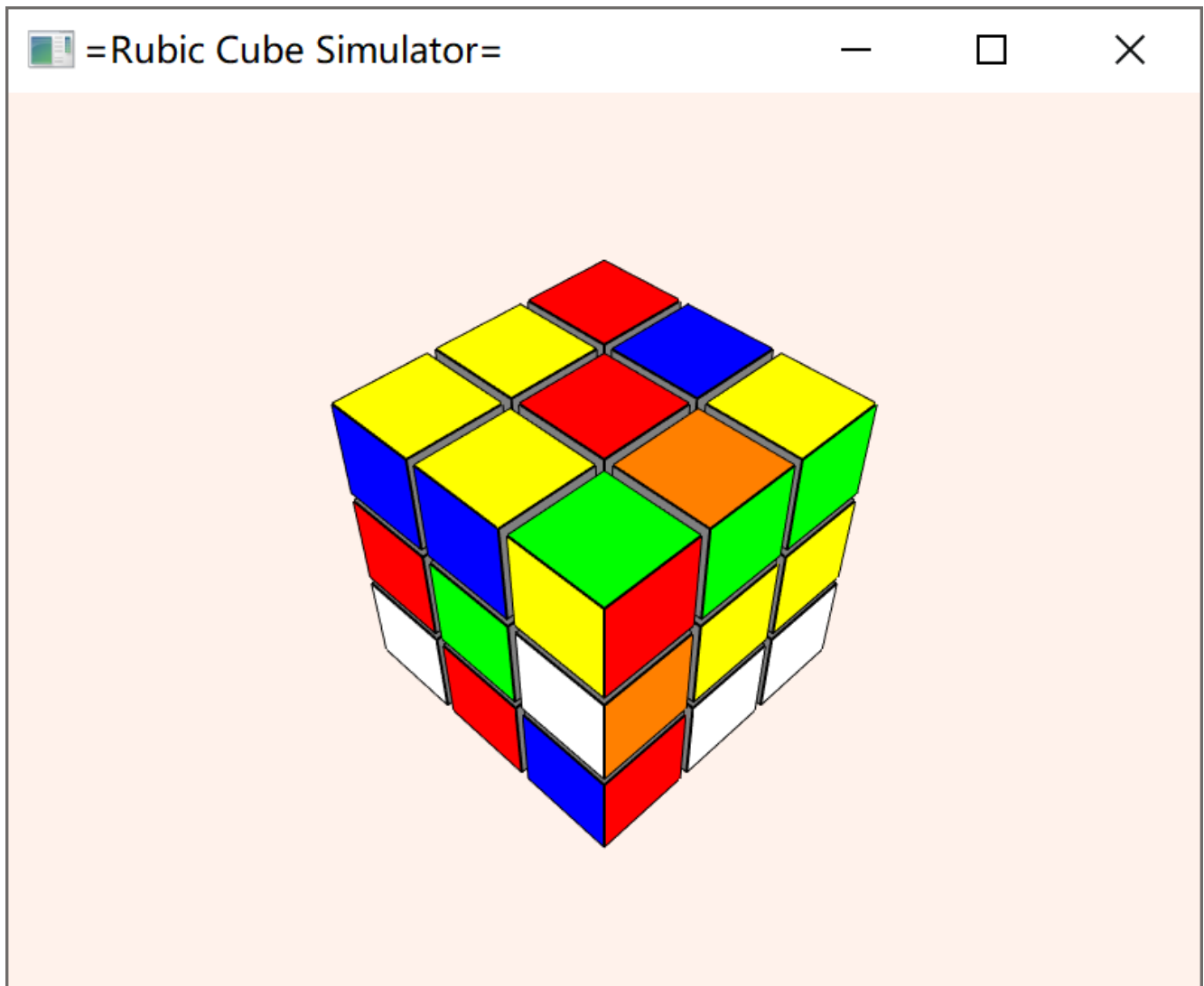
2.3 输入指令



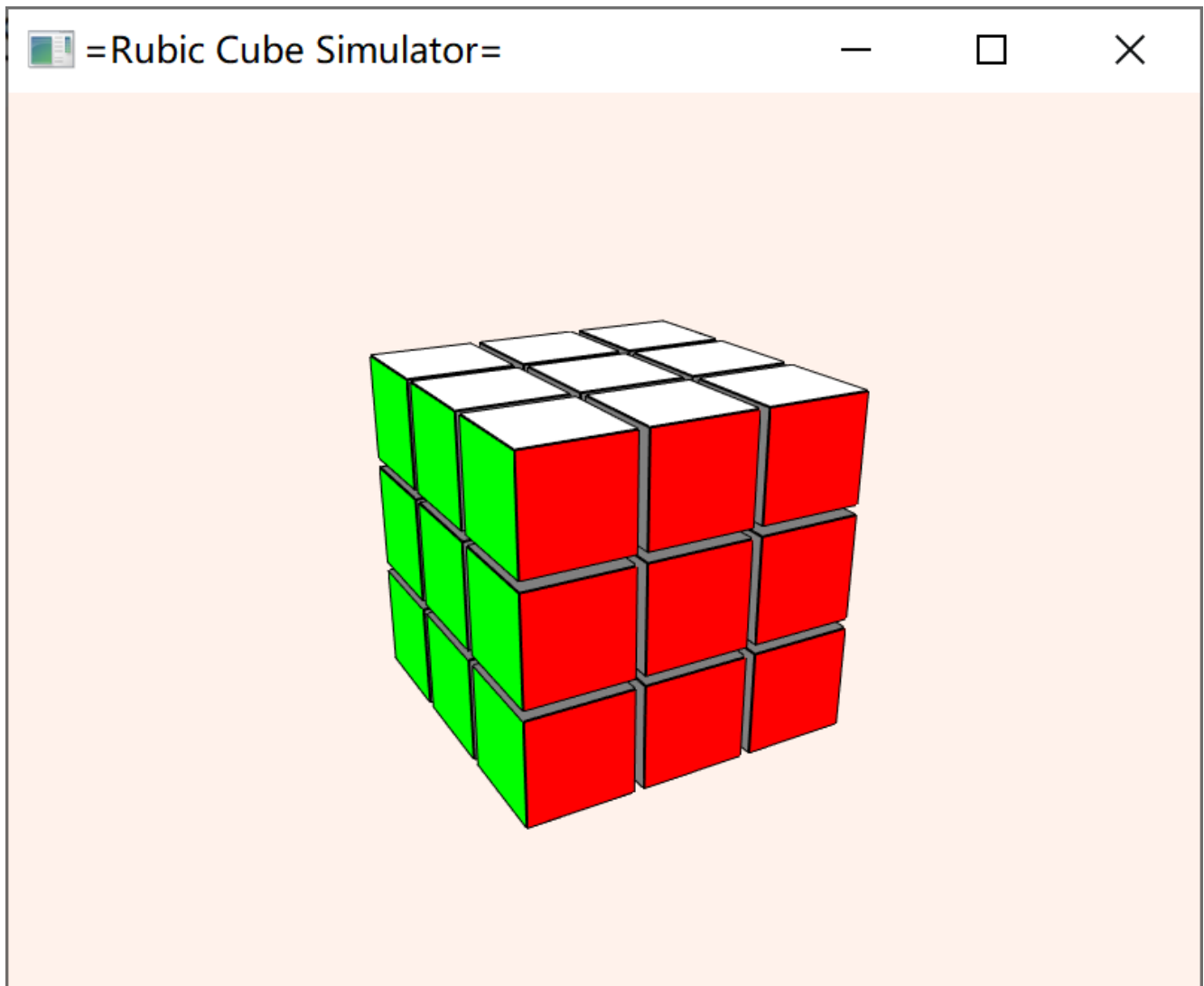
2.4 旋转



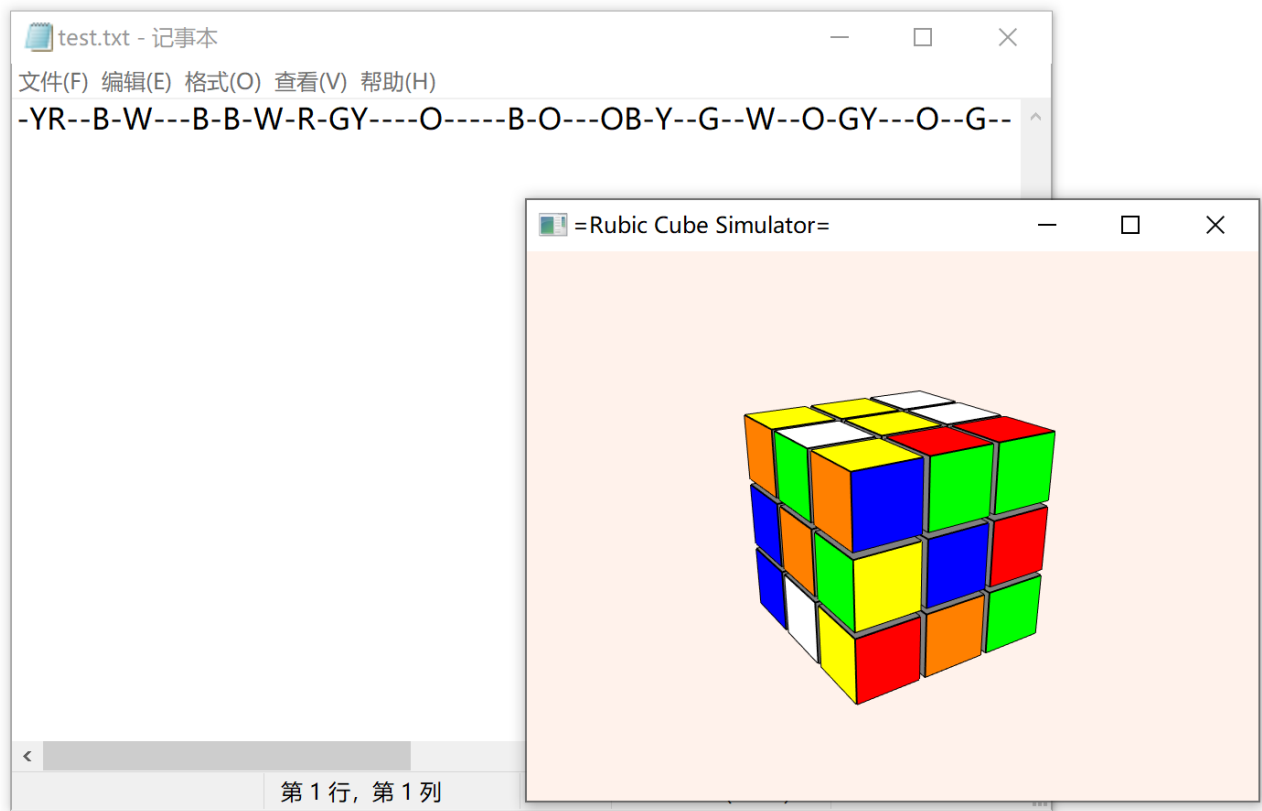
2.5 随机打乱



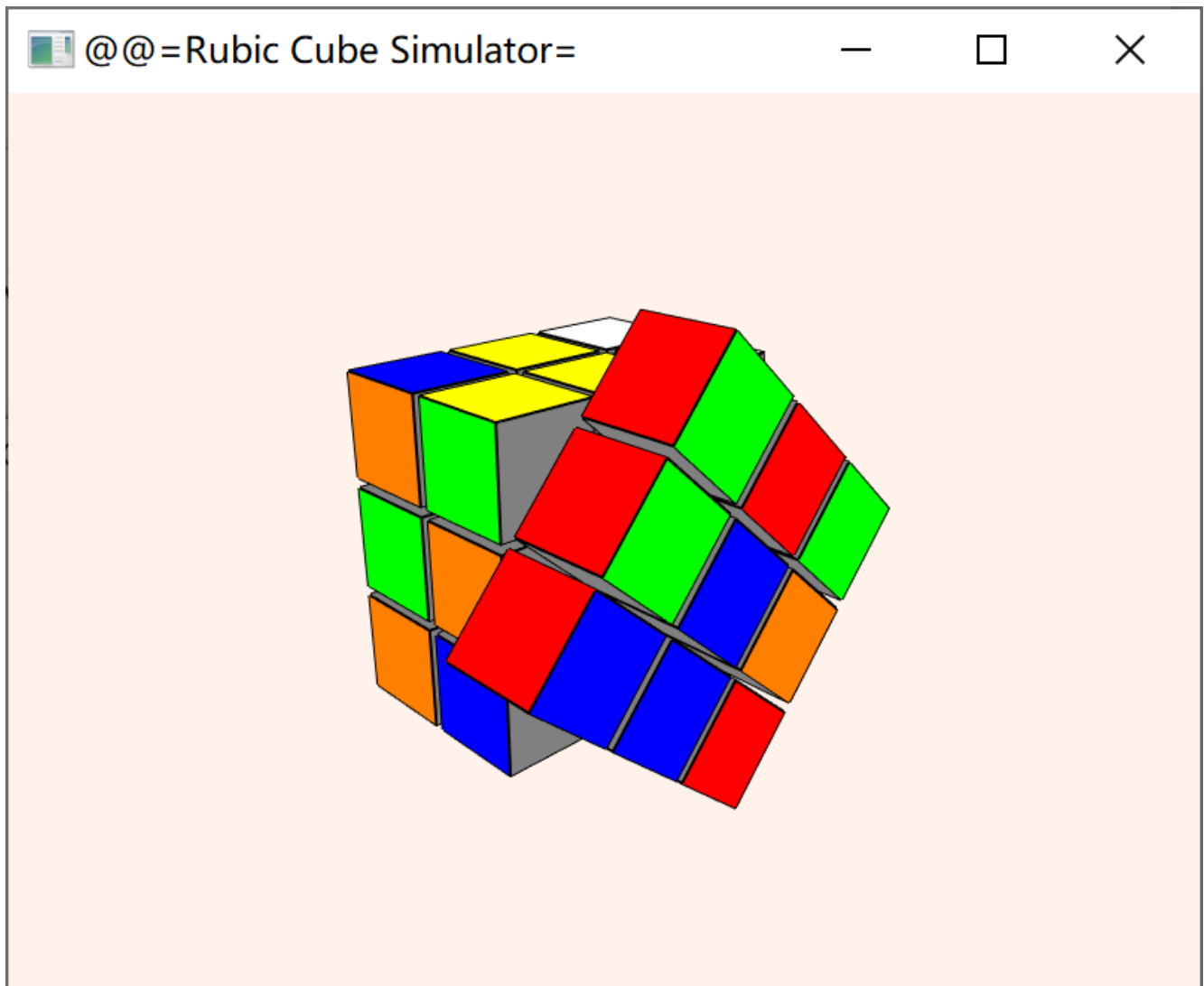
2.6 复原



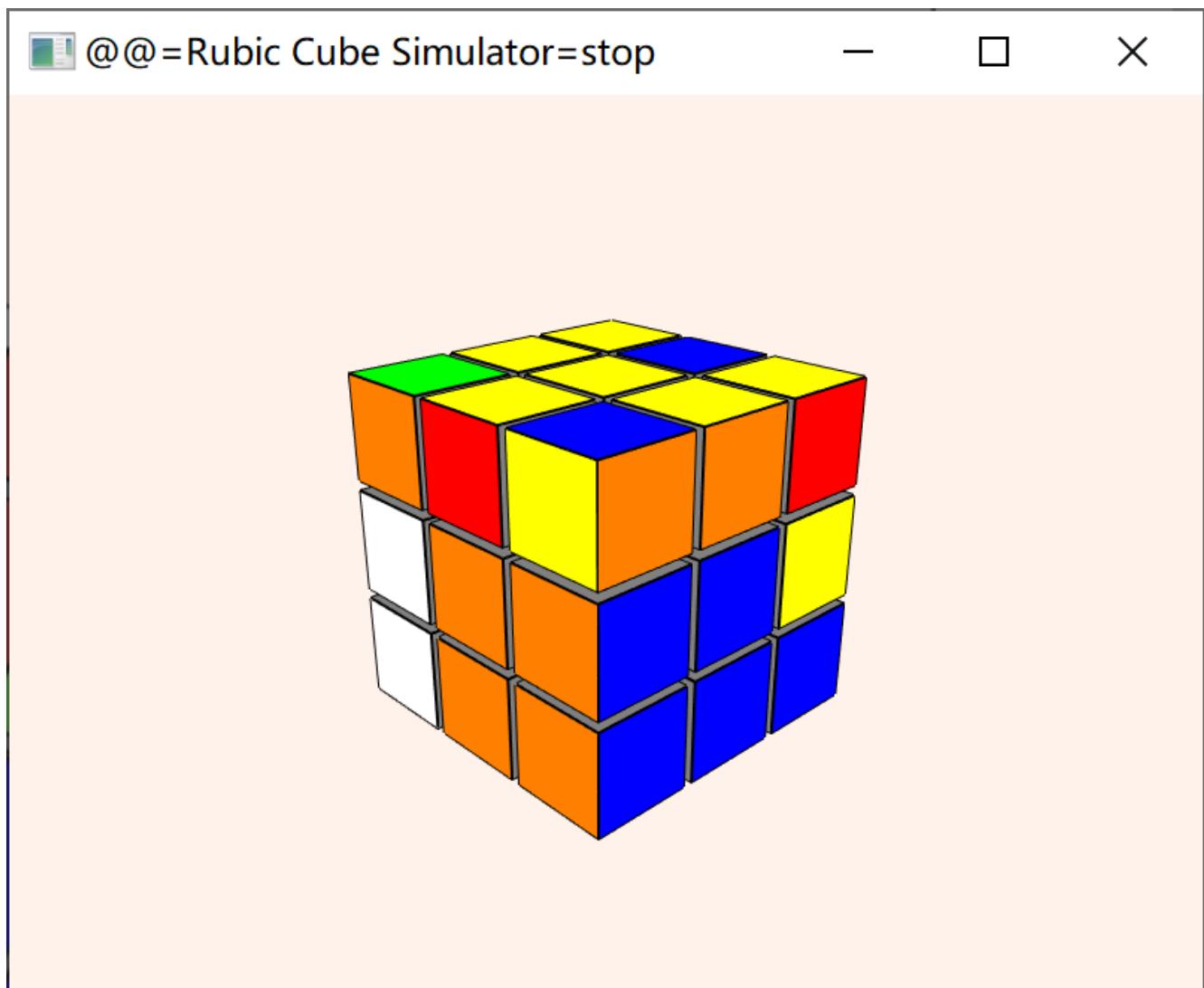
2.7 文件存储与读取



2.8 求解魔方并演示



2.9 停止旋转



3 操作说明

指令	参数	效果
鼠标拖动	/	移动视角
reset	/	重置魔方至初始状态
random	/	随机打乱魔方
r	/	顺时针旋转魔方右面
ri	/	逆时针旋转魔方右面
l	/	顺时针旋转魔方左面
li	/	逆时针旋转魔方左面
u	/	顺时针旋转魔方上面
ui	/	逆时针旋转魔方上面
b	/	顺时针旋转魔方后面
bi	/	逆时针旋转魔方后面
f	/	顺时针旋转魔方前面
fi	/	逆时针旋转魔方前面
x	/	顺时针沿x轴旋转整个魔方
xi	/	逆时针沿x轴旋转整个魔方
y	/	顺时针沿y轴旋转整个魔方
yi	/	逆时针沿y轴旋转整个魔方
z	/	顺时针沿z轴旋转整个魔方
zi	/	逆时针沿z轴旋转整个魔方
save	<FILE NAME>	保存魔方当前状态到指定文件
load	<FILE NAME>	加载对应文件的魔方状态
solve	/	求解魔方并开始自动复原
stop	/	停止旋转

3.1 协作情况

使用github进行项目协作

提交记录：



Commits

main

Commits on Jul 12, 2023

中期文档提交

lhmd committed 4 days ago

753c3f0 <>

个人分报告提交

JaqEst committed 4 days ago

a0c88d0 <>

第三次迭代-wwj

lhmd committed 4 days ago

Verified

1a1aaca <>

第三轮迭代-zrx

lhmd committed 4 days ago

Verified

f702467 <>

VIEWMODEL层: 完成第三轮迭代

lhmd committed 4 days ago

0c2f1ef <>

COMMON层: 添加魔方判断宏

zhongpang2023 committed 4 days ago

b1b65ae <>

COMMON层: 完成复原魔方错误处理

zhongpang2023 committed 4 days ago

6a8d396 <>

VIEWMODEL层: 添加复原命令

lhmd committed 4 days ago

09bf87e <>

VIEWMODEL层: 完成复原算法过滤器

lhmd committed 4 days ago

9b5221d <>

完成魔方自动复原算法

lhmd committed 4 days ago

1252c31 <>

更新为第二轮版本迭代

lhmd committed 4 days ago

fd19670 <>

VIEW层: 添加命令, 以及命令调用关系

JaqEst committed 4 days ago

4ab40dc <>

VIEWMODEL层: 实现旋转结束指令

JaqEst committed 4 days ago

394e686 <>

COMMON层: 简化实用工具

JaqEst committed 4 days ago

7223cdd <>

APP层: 添加旋转完成指令绑定

JaqEst committed 4 days ago

1dc790d <>

Commits on Jul 11, 2023

调整项目结构

JaqEst committed 5 days ago

7361672 <>

Merge pull request #6 from zju-lhmd/zrx_iter1

lhmd committed 5 days ago

Verified

c86b948 <>

merge mainwindow

lhmd committed 5 days ago

aba0372 <>

merge branch

lhmd committed 5 days ago

1b65549 <>

merge main branch

lhmd committed 5 days ago

b546ea0 <>

VIEWMODEL: 增加槽函数和命令实现

zhongpang2023 committed 5 days ago

65d2fe7 <>

VIEWMODEL: 定义ViewModel类, 发送各指令

zhongpang2023 committed 5 days ago

df93269 <>

补丁: 添加命令可视化

lhmd committed 5 days ago

Verified

1bd4894 <>

MODEL层: 实现Cube类 (定义魔方结构、实现魔方旋转函数等)

zhongpang2023 committed 5 days ago

2b35fc0 <>

添加命令可视化

lhmd committed 5 days ago

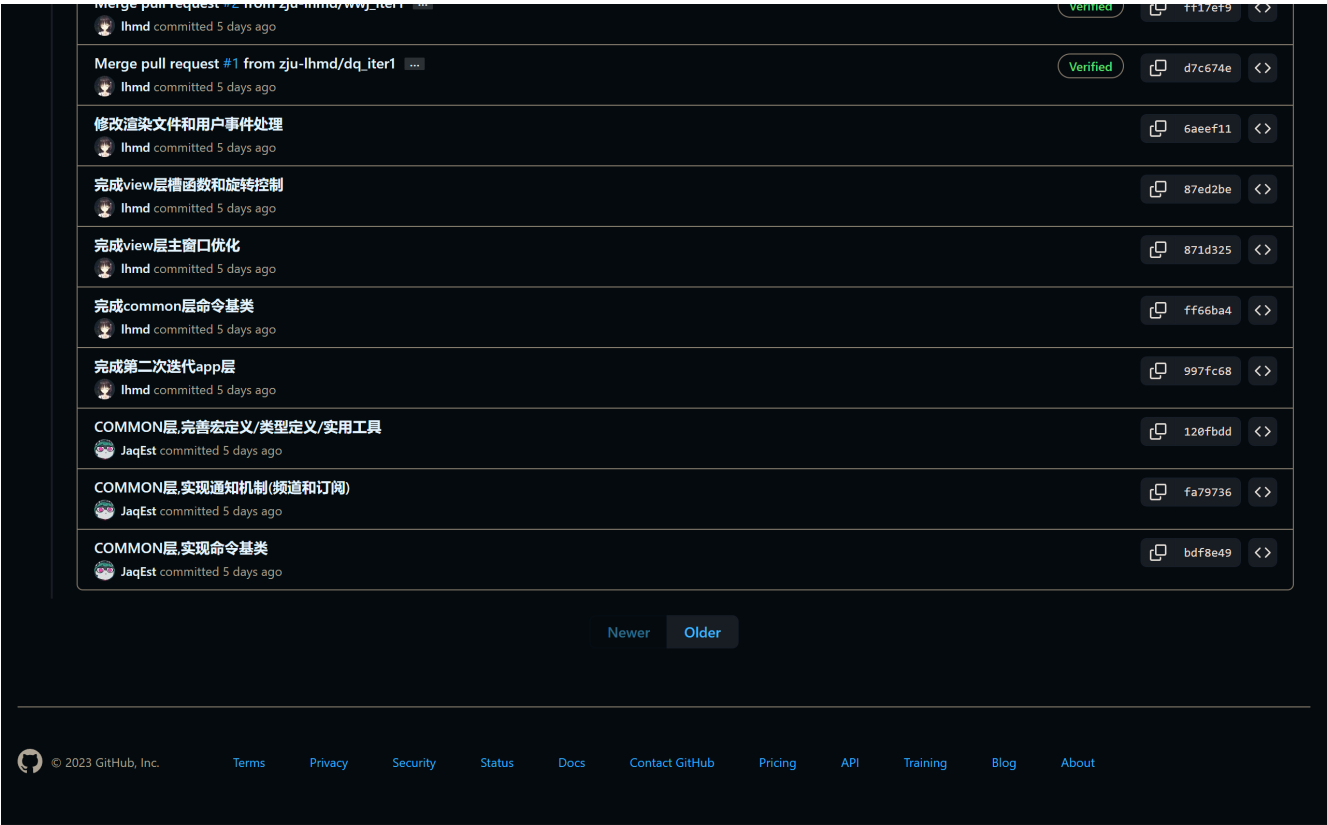
2bf1fdc <>

Merge pull request #2 from zju-lhmd/wwj_iter1

lhmd committed 5 days ago

Verified







1a1aaca <>



分支管理：



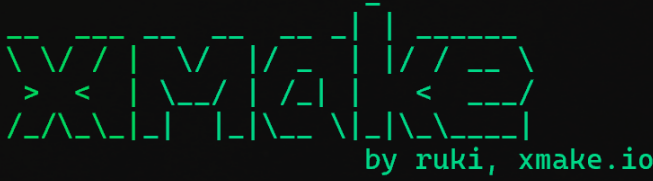
采用Pull request的方式进行集成：

<input type="checkbox"/>	 第三次迭代-wwj #8 by lhmd was merged 4 days ago
<input type="checkbox"/>	 第三轮迭代-zrx #7 by zhongpang2023 was merged 4 days ago
<input type="checkbox"/>	 第二次迭代-zrx #6 by zhongpang2023 was merged 5 days ago
<input type="checkbox"/>	 第二次迭代-zrx #5 by zhongpang2023 was closed 5 days ago
<input type="checkbox"/>	 补丁：添加命令可视化 #3 by lhmd was merged 5 days ago
<input type="checkbox"/>	 第二次迭代-wwj #2 by lhmd was merged 5 days ago
<input type="checkbox"/>	 第二次迭代-dq #1 by JaqEst was merged 5 days ago

3.2 开发工具

Xmake:

```
PS C:\Windows\system32> xmake --version
xmake v2.7.9+HEAD.c87922676, A cross-platform build utility based on Lua
Copyright (C) 2015-present Ruki Wang, tboox.org, xmake.io
```

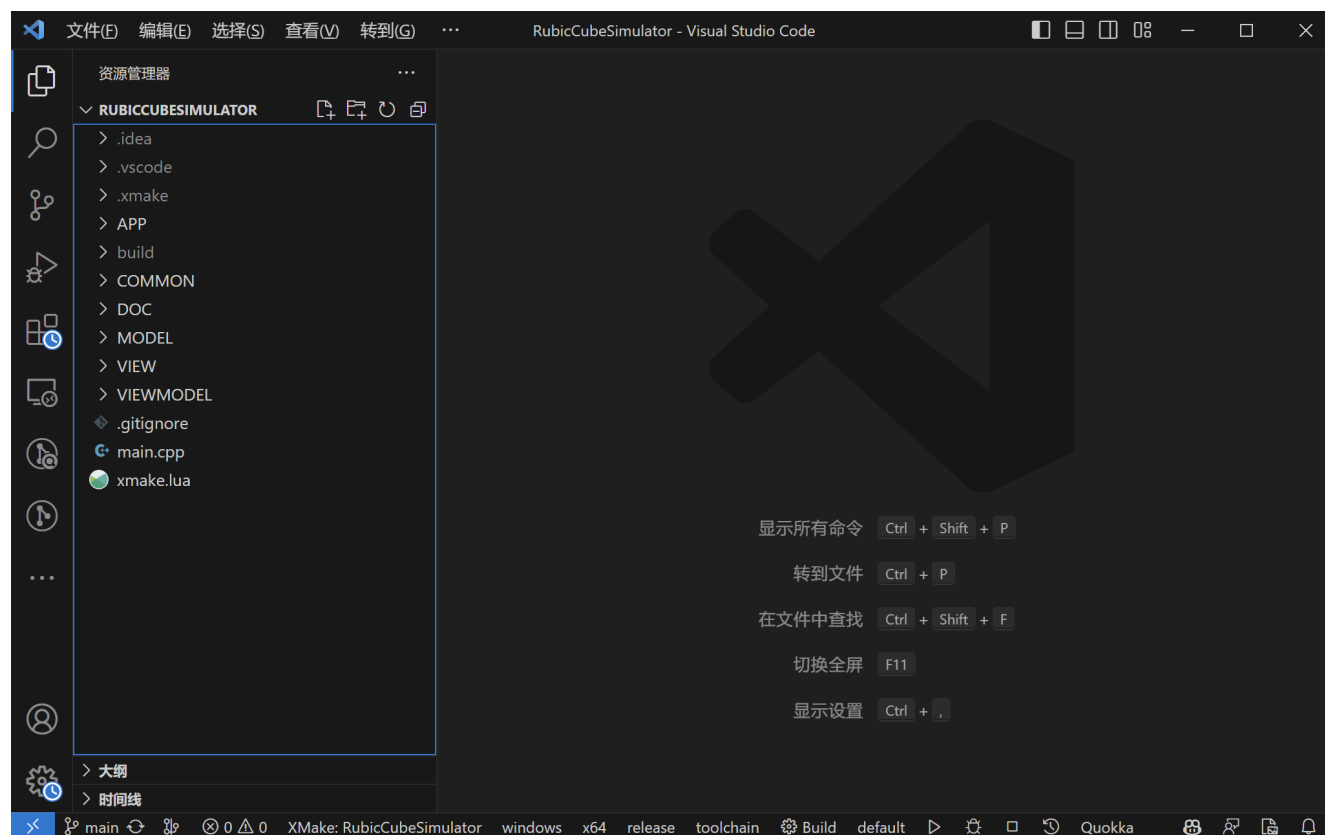


```
Manual: https://xmake.io/#/getting\_started
Donate: https://xmake.io/#/sponsor
```

编译过程:

```
PS E:\JUNIOR_AW\C++_PMEP\RubicCubeSimulator> xmake
checking for Microsoft Visual Studio (x64) version ... 2022
[ 4%]: compiling.release APP\App.cpp
[ 4%]: compiling.release VIEWMODEL\commands\ResetCommand.cpp
[ 4%]: compiling.release VIEWMODEL\commands\StopCommand.cpp
[ 4%]: compiling.release VIEW\MainWindow.cpp
[ 4%]: compiling.release VIEW\events\EventCallback.cpp
[ 4%]: compiling.release VIEWMODEL\commands\SolveCommand.cpp
[ 4%]: compiling.release VIEWMODEL\commands\RotEndCommand.cpp
[ 4%]: compiling.release main.cpp
[ 4%]: compiling.release VIEWMODEL\commands\LoadCommand.cpp
[ 4%]: compiling.release VIEWMODEL\solver\CubeSolver.cpp
[ 4%]: compiling.release VIEW\rotani\RotAni.cpp
[ 4%]: compiling.release VIEW\sinks\MainWindowSink.cpp
[ 4%]: compiling.release VIEWMODEL\commands\SaveCommand.cpp
[ 4%]: compiling.release VIEWMODEL\commands\RandomCommand.cpp
[ 4%]: compiling.release MODEL\Cube.cpp
[ 4%]: compiling.release VIEW\render\Renderer.cpp
[ 4%]: compiling.release COMMON\utilities.cpp
[ 4%]: compiling.release VIEWMODEL\commands\RotateCommand.cpp
[ 20%]: compiling.release VIEWMODEL\solver\NoXYZFilter.cpp
[ 20%]: compiling.release VIEWMODEL\solver\ReduceFilter.cpp
[ 20%]: compiling.release VIEWMODEL\ViewModel.cpp
[ 91%]: linking.release RubicCubeSimulator.exe
[100%]: build ok, spent 3.453s
```

使用VS Code进行开发:



4 心得

4.1 王伟杰

学习这个课程之前，我对C++的项目管理以及各种软件架构模式没有任何了解，之前在其他课程上和组员合作完成项目时也体会到了很多合作开发的困难。在学习这门课时，我不仅在课堂上学到了很多团队开发和程序框架相关的知识，还亲身体验了团队项目工程的合作开发。在我们完成MVVM项目框架的时候，我深刻体会到了这个模式的便捷之处，每个人都只专注于自己的模块，不需要等待其他人的代码，而且在多次迭代的过程中，项目成员也可以对整个项目的代码进行一遍完整的开发，让大家都能学到很多东西。

在第一轮迭代中，我负责了Common层和App层以及xmake项目管理的任务。通过这个阶段的工作，我对C++和MVVM模式有了更深入的了解。在第二轮迭代中，我负责了View层的开发以及文档的撰写。这使我能够进一步加深对用户界面和文档编写的理解。通过GitHub的版本控制和分支管理，我能够与其他组员无缝协作，确保代码的顺利合并和项目的进展。

4.2 戴卿

之前我对项目管理和软件架构模式有一定的了解，但在实际的团队开发过程中还存在一些不足。通过参与这个课程的学习和项目实践，我对团队协作和软件架构有了更深入的认识和体验。

在第一轮迭代中，我负责了ViewModel层和Model层的任务。通过这个阶段的工作，我学会了如何合理地管理数据和处理业务逻辑，以及如何与其他层进行良好的交互。在第二轮迭代中，我转变了角色，负责了Common层和App层的开发工作。通过这个过程，我进一步提高了对C++编程语言的理解和应用能力。

参与这个项目让我更深入地了解了MVVM架构模式，并在实践中加以应用。通过模块化的开发和清晰的分工，我们能够更加高效地开发和测试各自的模块，大大提高了工作效率。同时，我也学到了如何撰写清晰而详细的文档，以便更好地与团队成员进行沟通和交流。

4.3 钟睿昕

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。

4.4 总结

学习这门短学期课程之前，我们对C++的项目管理和软件架构模式都没有太多的了解。然而，在课程的学习过程中，我们不仅在课堂上获得了大量的理论知识，还有机会亲自参与了一个真实的团队项目开发，从中学到了许多宝贵的经验。

首先，通过这门课程，我们学习到了不同的项目管理工具和方法。我们采用了GitHub作为项目管理工具，利用其版本控制和分支管理功能，使得团队成员可以并行开发各自的模块，并轻松地解决代码冲突和合并问题。这种协作方式大大提高了我们的工作效率，使得每个成员都能专注于自己负责的部分，而不会互相干扰或产生冲突。

其次，我们学习了MVVM（Model-View-ViewModel）软件架构模式，并将其应用到项目开发中。MVVM模式将用户界面（View）与数据和逻辑（Model）分离，并通过ViewModel来连接二者。这种模式的使用使得我们可以独立地开发和测试各个模块，而不需要等待其他成员的代码完成。这种并行开发的方式极大地提高了我们的协作效率，同时也增加了项目的稳定性和可维护性。

在多次迭代的过程中，我们有机会对整个项目的代码进行一遍完整的开发。这使得我们能够更好地理解整个项目的结构和功能，并在每次迭代中不断改进和优化代码。这种全局的开发经历让我们受益匪浅，不仅加深了对C++编程语言和OpenGL图形库的理解，还锻炼了我们的问题解决能力和团队协作能力。

5 对课程的建议

- 希望能多讲一些样例，现在的课堂展示代码样例感觉偏少且较复杂，理解起来稍有困难。
- 作业布置前可以将要求更明确一些，比如项目开题时的报告可以要求的更加详细一点。
- 可以对项目难度及复杂度给出要求，将代码任务量和实现难度纳入考核中。