

三阶魔方问题的求解

-- 《高级软件工程》课程结项报告

指导老师：罗铁坚

成员：李港 赵天赐

2019 年 12 月 14 日

1. 背景介绍

最早的魔方是匈牙利的一位叫鲁比克（Rubik）的教授于 1974 年发明的，但是这位教授发明它并不是为了投入生产和娱乐。因为他是建筑学教授，为了帮助学生们认识空间立方体的组成和结构以及锻炼学生的空间思维能力和记忆力，鲁比克（Rubik）教授设计了一个立方体切割的实验，这就是魔方最早的概念雏形。他在这个概念的基础上，想制作出一个辅助教学的教具，他用了 6 周的时间设计出了一个可以上下左右旋转并且交叉换位的 $3 \times 3 \times 3$ 正立方体结构，制作出这个教具后，鲁比克（Rubik）教授在其 6 个外表面涂以 6 种不同的颜色，魔方从此诞生。

魔方是一个典型的复合型解密游戏，它有非常庞大的状态空间，但是目标状态有且只有一个，即保证各面的色块保持一致。借助于其独特的空间结构，魔方现在已经成为智力游戏的王牌，至今依然畅销全世界。但是现在对于初学者，如果要自己摸索复原魔方的方法，首先是对个人的空间思维要求很高，需要记住不同的状态，以及对应移动前后色块的变换；其次很少有人能很快上手，往往需要耗费很多的时间来熟悉。而对于一些想娱乐或者放松的人来说，这么耗时耗力，最后还不一定可以复原魔方，不免会对自己的自信心有些打击。

现在学习复原魔方的方法都是配套的公式，可以说只要按照公式来做，最后肯定可以复原魔方。但是公式一般都是印在纸上的，一是不便于理解学习，还有就是一旦中间有什么步骤走错了，很容易前功尽弃。即便网上有一些视频教程，但是都是看别人做，其实跟看公式大同小异，并没有跟玩家进行交互。所以对于学习复原魔方的时候，就需要有一个辅助型工具，一步步的引导玩家怎么复原魔方，帮助玩家掌握复原的公式。

2. 预期目标

用户可以将现实中魔方的状态输入系统中，系统求解之后，用户可以按照系统提示一步一步的跟着求解，也可以自己按照公式求解，同时在系统中记录每一步的操作，以此帮助用户完成魔方复原，并辅助用户记忆公式，最终使得用户可以自己完成魔方复原。

3. 需求分析与工作内容

用户可以按照现实中魔方的状态输入到系统中进行操作，然后可以跟着系统的复原思路一步步求解，也可以按照自己的思路进行操作。另外，用户还可以采用随机打乱按钮，在系统中练习自己的复原魔方能力，同时还可以调整魔方的视角，方便求解。系统的用例图如图 1 所示：

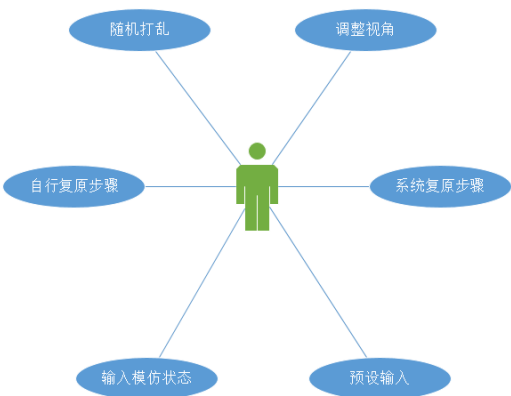


图 3.1 魔方系统用例图

通过分析用户的需求，本项目的工作内容可分为两个模块，即输入模块和复原模块。输入模块包括用户自行输入以及系统预设输入；复原模块包括系统复原、用户复原、调整视角和随机打乱。系统功能模块图如图 2 所示。

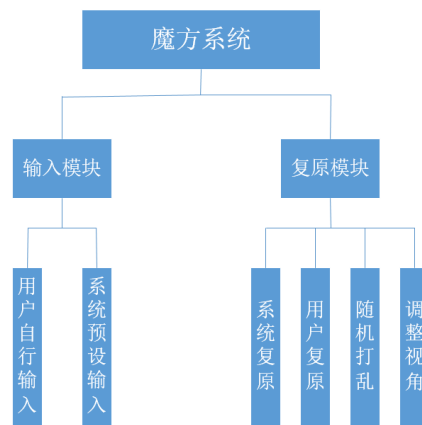


图 3.2 魔方系统功能模块图

4. 概要设计

4.1 相关技术

JavaScript（通常缩写为 JS）是一种高级的、解释型的编程语言[5]。JavaScript 是一门基于原型、函数先行的语言[6]，是一门多范式的语言，它支持面向对象编程，命令式编程，以及函数式编程。它提供语法来操控文本、数组、日期以及正则表达式等，不支持 I/O，比如网络、存储和图形等，但这些都可以通过它的宿主环境提供支持。它已经由 ECMA（欧洲电脑制造商协会）通过 ECMAScript 实现语言的标准化[5]。它被世界上的绝大多数网站所使用，也被世界主流浏览器（Chrome、IE、Firefox、Safari、Opera）支持。

JSP（全称 JavaServer Pages）是由 Sun Microsystems 公司主导创建的一种动态网页技术标准。JSP 部署于网络服务器上，可以响应客户端发送的请求，并根据请求内容动态地生成 HTML、XML 或其他格式文档的 Web 网页，然后返回给请求者。JSP 技术以 Java 语言作为脚本语言，为用户的 HTTP 请求提供服务，并能与服务器上的其它 Java 程序共同处理复杂的业务需求。

Tomcat 是由 Apache 软件基金会属下 Jakarta 项目开发的 Servlet 容器，按照 Sun Microsystems 提供的技术规范，实现了对 Servlet 和 JavaServer Page（JSP）的支持，并提供了作为 Web 服务器的一些特有功能，如 Tomcat 管理和控制平台、安全域管理和 Tomcat 阀等。由于 Tomcat 本身也内含了 HTTP 服务器，因此也可以视作单独的 Web 服务器。但是，不能将 Tomcat 和 Apache HTTP 服务器混淆，Apache HTTP 服务器是用 C 语言实现的 HTTPWeb 服务器；这两个 HTTP web server 不是捆绑在一起的。Apache Tomcat 包含了配置管理工具，也可以通过编辑 XML 格式的配置文件来进行配置。

宝塔面板是一款服务器管理软件，支持 windows 和 linux 系统，可以通过 Web 端轻松管理服务器，提升运维效率。也可以一键配置服务器环境（LAMP/LNMP/Tomcat/Node.js），还可以创建管理网站、FTP、数据库，拥有可视化文件管理器，可视化软件管理器，可视化 CPU、内存、流量监控图表，计划任务等功能。同时，为了方便用户建立网站，宝塔面板上的一键部署源码插件，可一键部署：Discuz, Wordpress, Ecshop, Thinkphp, Z-blog, DedeCMS 等程

序。还有极其方便的一键迁移，两台服务器安装宝塔 Linux 面板 5.2 版本，可实现一键迁移服务器网站、FTP、数据库。

4.2 界面设计

本项目主要有两个页面，即输入模仿状态的初始页面，以及魔方复原的操作页面。其中，在初试页面，用户可以通过点击的方式给色块涂色，也可以采用系统预设的四种涂色方案，提交之后，进入复原界面，用户可以选择自行操作复原，也可以按照系统指示进行复原。另外，用户在复原过程中还可以调整视角，以及随机打乱，再次练习。

5. 详细设计

5.1 前端设计实现

前端页面采用 CSS 样式进行网页样式自定义布局和调整，前端的页面主要包括两部分，即初试页面和复原页面。

首先在初始页面，映入眼帘的六个 3×3 矩阵表示魔方的六个面，然后底下有可选择的颜色，用以表示小方块的颜色，用户可以根据自己现实实际中，魔方的颜色情况进行涂色，也可以选择系统预设的四个涂色方案，来进行魔方复原。初试页面如图 5.1 所示。

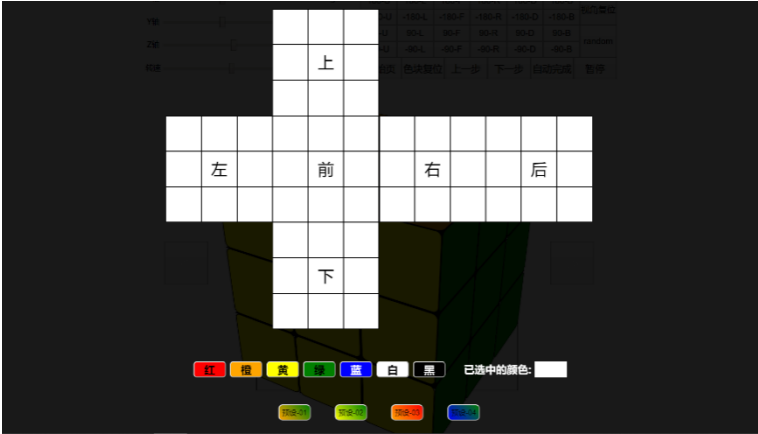


图 5.1 初始页面效果

涂色完毕之后，即可提交，进入到复原界面。在复原界面，用户可以按照系统的复原步骤直接点击下一步或者自动完成观看系统还原的步骤，也可以点击右上方的操作按钮，对魔方做相应的旋转，即自行实现魔方复原。与此同时，用户还可以通过鼠标旋转或者左上角的角度修改的方式来调整魔方的视角，方便用户更好的实现魔方复原。最后在右上角还有一个随机打乱按钮，用户可以采用随机打乱按钮重新开始魔方复原。复原界面如图 5.2 所示。

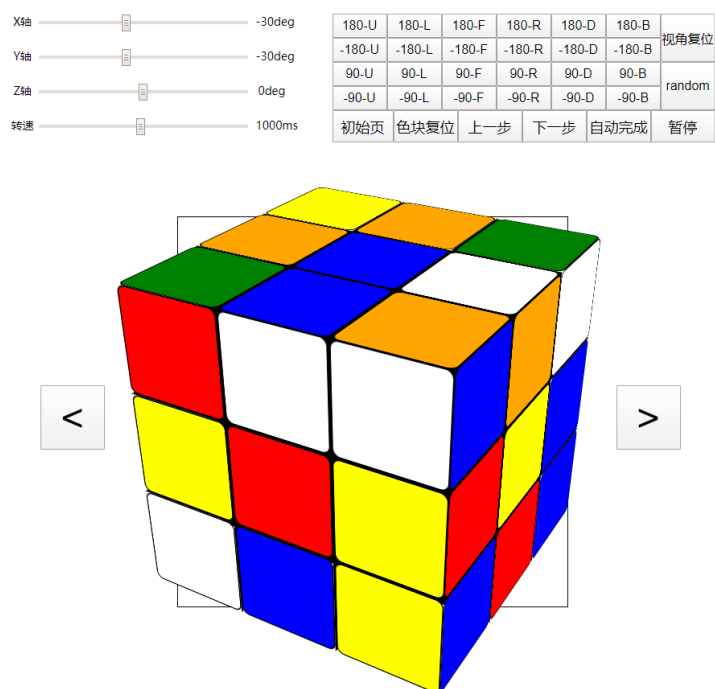


图 5.2 复原页面效果

5.2 后端设计实现

在本项目中，考虑到 DeepCubeA 中的求解思路虽然步骤少，但是不方便人类理解，更是对新手不友好，所以在求解的时候，采用了基于魔方公式规则的方法。首先魔方的公式采用市面上已知的，且行业一致认可的公式，即以此来保证用户所购买魔方的公式与本系统的求解步骤相同。然后，对于魔方求解方法，除了系统自己求解的过程之外，用户还可以根据自己的理解和判断来进行操作。接下来介绍本项目采用的魔方公式以及后端设计实现中其他细节。

5.2.1 魔方公式

第一步：底棱归位（又称底部架十字，底层四个棱块复原的过程）



如图 5.3 所示，魔方底层架十字可以无师自通，只是我们这一步要复原的四个棱块的相对位置顺序要注意，由于我们以白色中心块做底层，按照我们现在的主流魔方的贴纸的帖法（上黄下白，前蓝后绿，左橙右红），如果我们先复原了白蓝这个棱块，那我们在保持白色中心块在底部的情况下，白红的棱块就一点要放在白蓝棱块的右边，白橙棱块放在白蓝棱块的左边，白绿棱块放在白蓝棱块的对面，由于魔方的中心块不会发生变化，所以在复原的过程中，我们是以中心块为参照物的，第一步我们在复原白蓝、白红、白绿、白橙这四个棱块的时候，我们可以先把白色面旋转到顶层，和黄色中心块同一个平面，然后再把他对应的另一个颜色（蓝或红或绿或橙）经过旋转最上层，使之和对应的中心块的颜色同色，这样我们再旋转 180 度，对应的棱块就正确复原到底部了。



图 5.3 第一步底棱归位

第二步：底角归位（复原魔方第一层四个角块）

先将目标角块调至顶层侧面，再转动能与之相连形成顺色整体的面，使目标角与底棱连成一个 $(1 \times 1 \times 2)$ 的归位整体，再转至正确的位置。具体公式如图 5.4 所示。

	
公式2-1: (RUR')	公式2-2: $(F'U'F)$
记忆技巧: 白色朝右, 第一步就旋转右层	记忆技巧: 白色朝前, 第一步就旋转前层




		
用两次公式2-1	用两次公式2-2	用三次公式2-1
$(RUR')U'(RUR')$	$(F'U'F)U(F'U'F)$	$(RUR')(RUR')U'(RUR')$

图 5.4 底角归位

第三步: 中棱归位 (复原魔方中层四个棱块的步骤)

魔方中间层共有四个棱块, 也只是四个棱块需要复原, 当你需要的棱色块不在顶面, 而在中间层棱块的位置, 但颜色反了, 碰到这种情况或者类似这种情况, 我们就用 3-1 或者 3-2 的公式把最上面一层的其他颜色的棱块转移到该位置, 我们要的那个蓝红棱块就自然换到顶层了。具体公式如图 5.5 所示。




		
公式3-1: $(U'F'UF)(URU'R')$	公式3-2: $(URU'R')(U'F'UF)$	公式3-1+U2+公式3-1或者 公式3-2+U2+公式3-2

图 5.5 中棱归位

第四步: 顶棱面位 (也称顶层架十字, 顶层四个棱块的顶面颜色和顶层中心块颜色一样)


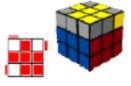
这一步我们只用一个公式就可以完成顶部十字, 在用公式的过程前, 在使用公式之前, 请注意魔方上层的位置, 如果把已经面位的两个棱块的位置旋转放置在左上和右上位置, 才可使用公式完成。具体公式如图 5.6 所示。

		
相对顶棱面位	相邻顶棱面位	无顶棱面位
公式4: $F(RUR'U')F'$	两遍公式4	两遍公式4+U+公式4

图 5.6 顶棱面位

第五步: 顶角面位 (魔方的四个顶角的顶面色全部调至顶面的步骤)

当我们完成了顶层十字以后, 我们来完成顶角的面位 (即顶层角块的翻色), 我们还是先观察一下, 现在我们的魔方的四个角块是什么状况, 如果是已经有一个角的黄色在顶层, 其他三个角的顶面颜色不是黄色, 首先把顶层面是黄色的那个角块移动到前右的位置, 再来对照图示, 看是是哪种情况, 我们对应用那个公式来完成即可完成顶角面位的步骤, 如果不是公式给出的两种情况, 那一定是下图中的其他 5 种情况其中的一种情况, 按照这 5 种情况的对应图示放好自己的魔方, 然后按照对应的操作步骤来完成。具体公式如图 5.7 所示。

			
公式5-1: $(RU' 2R')U' (RU' R')$		公式5-2: $(R' U2R)U (R' UR)$	






				
公式5-1+ 公式5-2	公式5-1+ 公式5-2	公式5-1+U2+ 公式5-2	公式5-1+U+ 公式5-1	公式5-1+ 公式5-1

图 5.7 顶角面位

第六步：顶角归位（面位的四个顶角的其他两面颜色和对对应面的中心块颜色同色）

这一步在复原顶角块的时候，先观察有无两个侧面颜色一样的情况，如果有，同色的侧面向自己，用公式 6 即可完成，如果不是，可随使用一次公式 6，就一定会出现有两个角块的侧面颜色一样的情况，再用一次公式 6 即可完成。具体公式如图 5.8 所示。



公式6: $(RB' RF2) (R' BRF2) R2$

图 5.8 顶角归位

第七步：顶棱归位（已经面位的四个顶棱的另一面的颜色和所在的另四个面的中心块颜色同色）

魔方顶层的四个角块正确归位以后，观察顶层的四个棱块，正常情况下，现在的魔方状态应该是下图四种情况中的一种，按照如图 5.9 所示的公式进行旋转即可完成魔方复原。

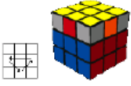



			
公式 7: $(RU' R) (URUR) (U' R' U' R2)$	公式 7 两次	公式 7+U+公式 7	公式 7+U'+公式 7

图 5.9 顶棱归位

5.2.2 魔方复原操作

为了方便用户可以熟悉魔方复原的步骤，本系统提供了两个魔方复原的方法，即用户自行操作复原和系统自动复原。首先，对于系统自动复原，就是用户提交输入状态之后，直接点击右上角的自动完成按钮，来观看系统完成魔方的动画演示。然后对于用户自行复原操作，用户可以选择点击右上角的旋转按钮，分别对应不同的面旋转相应的角度，以此来完成魔方复原。

5.2.3 魔方动画渲染

为了更直观的让用户看到魔方的旋转过程，需要在每一步操作后都要对魔方界面进行渲染，跟新魔方的颜色状态，并且实现了旋转的视觉效果，使用户看起来更加方便，也更容易与现实世界中的魔方对应起来，最终达到辅助用户学习魔方复原的目的。

6. 测试与发布

6.1 前端界面测试

该环节通过在不同浏览器上访问网站来查看网页布局是否有错位现象，此外还要检查魔方复原动画中用到的各种 javascript 代码是否能够顺利运行，其结果如表 6.1 所示。

表 6.1 前端界面测试结果表

项目\浏览器	Chrome	Edge	Safari	IE
页面布局	正常	正常	异常	异常
JavaScript	正常	正常	异常	异常

6.2 魔方复原测试

该环节主要测试魔方的自动复原情况，检验系统的公式法能否完美的复原魔方。经过测试，该系统可以完美的复原魔方。同时，对于用户自行求解操作的过程，也进行了测试，一切功能均正常，用户可以正常使用系统进行魔方复原。

6.3 项目发布

项目发布在阿里云服务器上，具体信息如表 6.2 所示。

表 6.2 云服务器部署信息

云服务器	阿里云 ECS
操作系统	Centos7
运行环境	Tomcat9
网站网址	http://www.campustao.cn:8080/ligang/

另外，对于云服务器的管理，还用到了宝塔面板，这是一个服务器可视化资源管理工具，虽然 linux 指令操作是每个计算机专业出身的必备技能，但是可视化更直观并且难度低，降低了运维人员的门槛。宝塔面板的服务器资源可视化界面如图 6.1 所示。

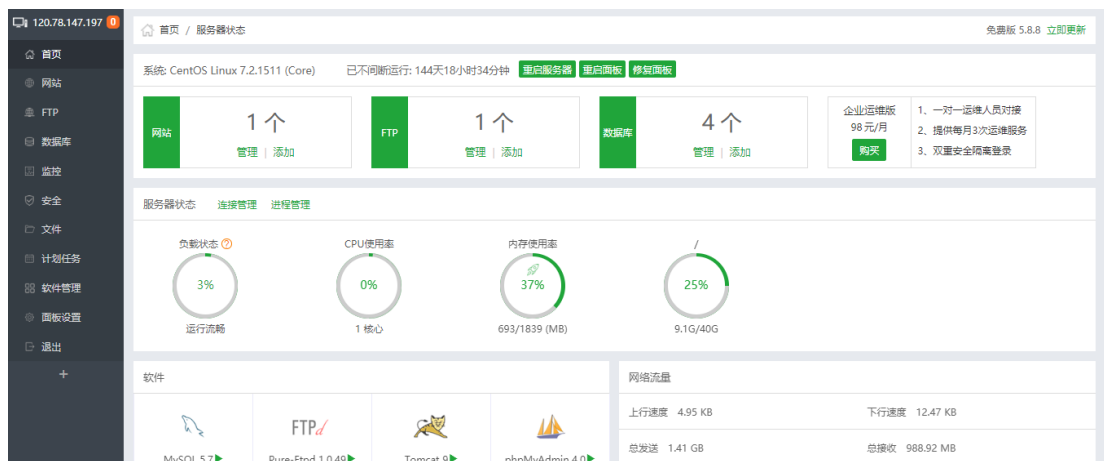


图 6.1 服务器资源管理首页

7. 总结

通过数周的努力，我们二人完成了魔方系统的开发、测试和部署等工作，实现了现实模拟状态的输入，魔方的系统求解，和人工操作以及随机打乱，还有视角调整。对于罗老师开课之初所提出的思维升级，对我们也产生了深远的影响，对于本项目没有采用深度学习的求解策略，也是经过反复的讨论和思考的。

首先，我们在按照课程指导文件进行环境配置的时候，总是出错，在网络上也查不到类似的问题，而且问其他同学都没有得到答案，我们认为可能是老师的课程指导文件中对于配置环境的步骤可能少了一些，虽说配置环境听起来很简单，但是如果涉及的环境变量很多，而且对服务器的硬件还有要求的话，那对我们新手的确是有些不友好。

在经过若干尝试未果之后，我们俩商议决定采用 B 计划，也就是现在的项目成果，使用公式法来求解魔方复原。其实在一开始我们就有考虑公式法，毕竟机器学习出来的求解方法并不是很好解释，并且也没有通用性，用户可能只知道当前状态需要怎么办，但是遇到别的状态，还需要通过机器求解，然后才能知道。但是公式法已然广为流传，拥有很好的群众基

础，并且其正确性在过去几十年的实践中早已得到了验证，而且公式法对于提高用户自信心有很大的作用。另外，本项目中的动画展示效果，能让用户更直观的看清楚公式的执行过程，避免用户因为字母符号而眼花缭乱不知所措，并且用户在系统中人工操作的时候，系统会记录用户的操作过程，当出现异常时，可以回退到上一步，简单易用，从而可以很好的帮助用户学会公式法求解魔方。

在此过程中，切身体验到软件开发和测试发布的流程，对软件工程有了全新的认识。此外，在开发过程中遇到的各种问题增加了我们对网站设计开发和发布的经验，强化了团队分工协作的能力，为今后的工作增长了经验。

最后，由于水平有限，本系统也存在着一些问题，在以后的学习工作中，可以按照以下几点进行升级改造：

- ✓ 改善前端界面不能随浏览器变化而自动兼容的情况
- ✓ 找寻更简单快捷易理解的求解方式，帮助用户快速上手
- ✓ 升级服务器配置，提高系统的并发访问量
- ✓ 系统测试采用自动化测试，提高测试用例的覆盖率