

# 数独乐乐软件系统建模报告

## 1. 引言

### 1.1 项目背景

这是一个用 Svelte 和 TailwindCSS 构建的简易数独游戏。

**项目功能：**

- 支持数独游戏的基本玩法
- 支持撤销/重做操作
- 支持自定义创建数独题目
- 响应式界面，适配多种设备
- 友好的用户交互体验
- 支持多种难度选择
- 提供自动求解和提示功能

### 1.2 建模目的

(1) 明确需求，统一理解

**可视化需求表达：**

通过用例图、用户故事等模型，将模糊的“游戏功能”转化为明确的交互流程（如“用户请求提示→系统返回一个正确数字”），避免开发团队与利益相关者之间的理解偏差。

**功能边界清晰化：**

例如，通过建模明确“撤销/重做”功能需记录操作历史栈，而“自

动求解”需独立算法模块，防止需求蔓延。

## **(2) 设计可扩展的系统架构**

### **模块化设计指导**

类图直接反映代码结构，确保高内聚低耦合。

### **技术选型依据**

序列图中“自动求解耗时操作”提示需使用 Web Worker 避免阻塞 UI，影响技术方案选择。

## **(3) 降低开发风险，提高效率**

### **提前暴露问题**

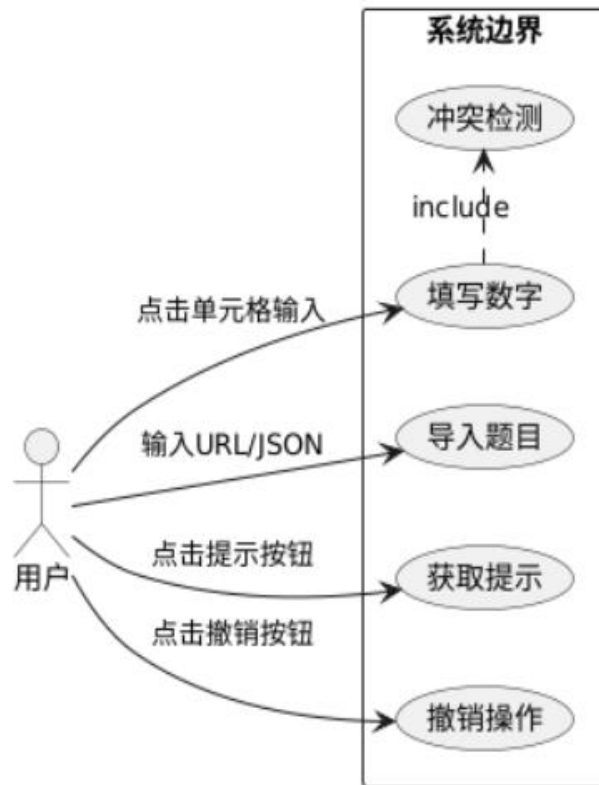
如状态图中“游戏从 Solved 状态返回 Playing”可能触发未重置计时器的漏洞，建模阶段即可发现并修正。

### **开发路线图**

根据模型优先级划分开发阶段（如先实现核心验证逻辑，再开发撤销功能）。

## **2. 系统建模**

### **2.1 用例图**



说明：

### （1）参与者

用户：唯一的外部参与者，直接与系统交互的所有操作均由其发起。

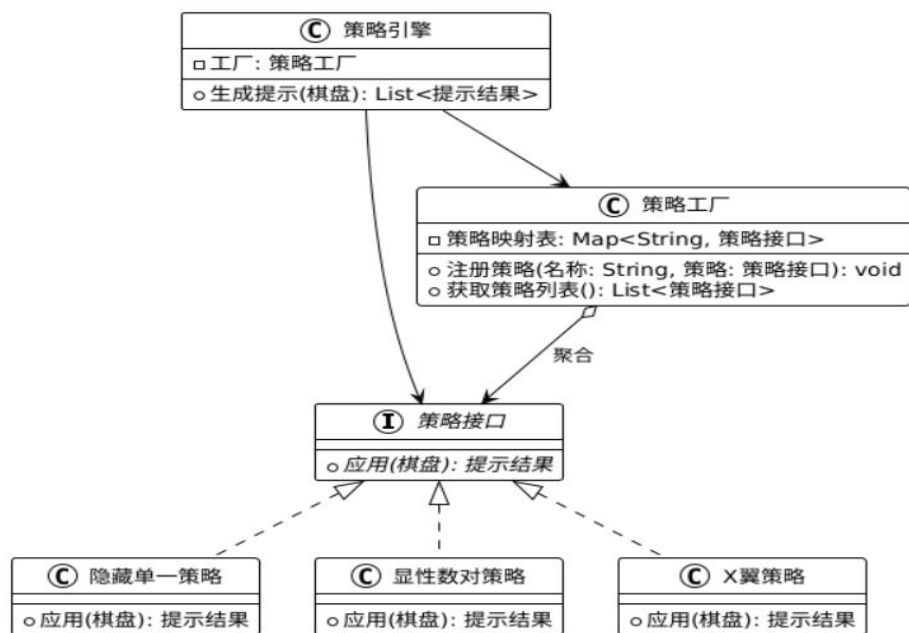
### （2）系统边界

实线框内为数独乐乐系统的核心功能范围，包含用户可直接触发的所有交互用例。

### （3）核心用例及关系

用例名称	说明	关联关系
点击单元格输入	用户选择棋盘上的单元格,进入数字填写状态	扩展为填写数字
填写数字	用户输入 1-9 的数字,系统需实时验证合法性 (如重复冲突)	包含冲突检测
获取提示	用户点击提示按钮,系统自动填充一个正确数字	依赖冲突检测结果
撤销操作	用户撤销上一步操作(如删除填写的数字), 依赖操作历史记录	与填写数字为组合关系
导入题目	用户通过 URL 或 JSON 导入自定义题目, 需验证格式有效性	独立功能

## 2.2 类图



说明:

### (1) 核心类及其职责

类名	职责	关键属性/方法
策略引擎	作为策略模式的上下文，协调策略的调用与结果返回	<ul style="list-style-type: none"> <li>策略工厂（依赖注入）</li> <li>生成提示(棋盘): 聚合各策略结果</li> </ul>
策略工厂	管理所有策略的注册与获取，实现动态扩展	<ul style="list-style-type: none"> <li>策略映射表（维护策略名称与实现类）</li> </ul>

类名	职责	关键属性/方法
		<ul style="list-style-type: none"> <li>注册策略(): 支持新增策略</li> </ul>
		<ul style="list-style-type: none"> <li>获取策略列表(): 供引擎调用</li> </ul>
策略接口	定义策略的统一行为(抽象层)	<ul style="list-style-type: none"> <li>应用(棋盘): 返回提示结果</li> </ul>
具体策略实现类	实现不同数独解题算法:	
<ul style="list-style-type: none"> <li>隐藏单一策略</li> </ul>	检测某数字在行/列/宫格中的唯一可能位置	覆写应用()方法
<ul style="list-style-type: none"> <li>显性数对策略</li> </ul>	发现同行/列/宫格中两个单元格的候选数对, 排除其他可能性	覆写应用()方法
<ul style="list-style-type: none"> <li>X翼策略</li> </ul>	高级技巧, 通过两行两列的交叉匹配排除候选数	覆写应用()方法

## (2) 类关系解析

- 依赖关系：

- 策略引擎依赖策略工厂获取策略实例。
- 所有具体策略依赖策略接口实现契约。

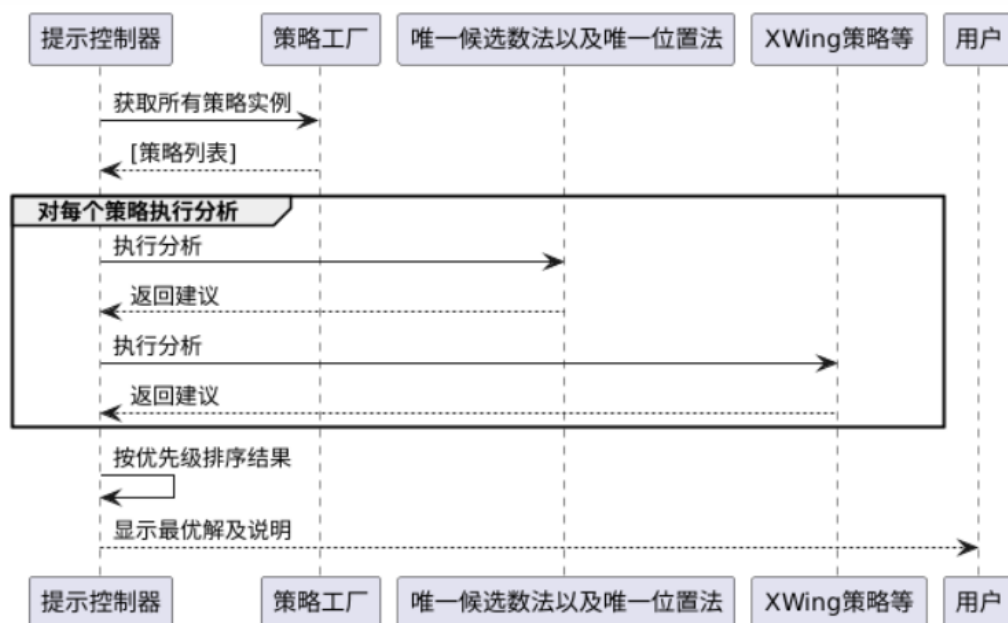
- 聚合关系：

- 策略工厂通过策略映射表聚合多个策略接口实例。

- 接口与实现：

- 策略接口是抽象层，具体策略通过覆写应用()方法提供不同算法。

## 2.3 序列图



说明：

### (1) 参与对象

对象	角色
用户	触发提示请求的终端用户
提示控制器	协调提示生成流程的中枢模块
策略工厂	管理所有策略实例的工厂类（维护策略列表及优先级）
具体策略实现	包括唯一候选数法、唯一位置法、XWing 策略等算法实例

## （2）交互流程

### 1. 用户发起提示请求

- 。 用户点击界面“提示”按钮，触发提示控制器。

### 2. 获取策略列表

- 。 提示控制器向策略工厂请求所有可用策略实例。
- 。 策略工厂返回已注册的策略列表。

### 3. 执行策略分析

- 。 提示控制器遍历策略列表，依次调用每个策略的执行分析()  
方法，传入当前棋盘状态。
- 。 各策略独立分析棋盘并返回建议结果。

### 4. 排序与筛选结果

- 。 提示控制器按策略优先级（如基础策略优先）和结果有效性排序。

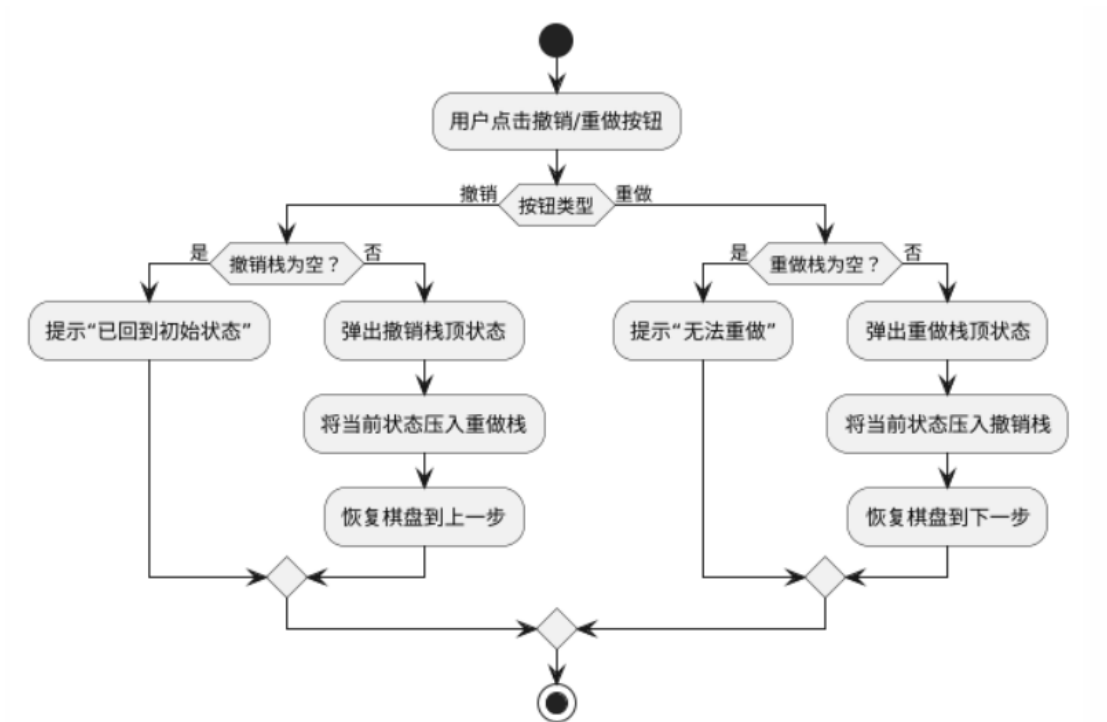


- 。排除无效建议。

## 5. 返回最优解

- 。提示控制器将最高优先级结果返回给用户界面。
- 。界面高亮目标单元格并显示说明。

## 2.4 状态图



说明：

### (1) 核心状态与转换

状态/条件	说明
初始状态	用户未进行任何操作时的棋盘状态
撤销栈不为空	存在可撤销的操作历史（栈存储每一步的棋

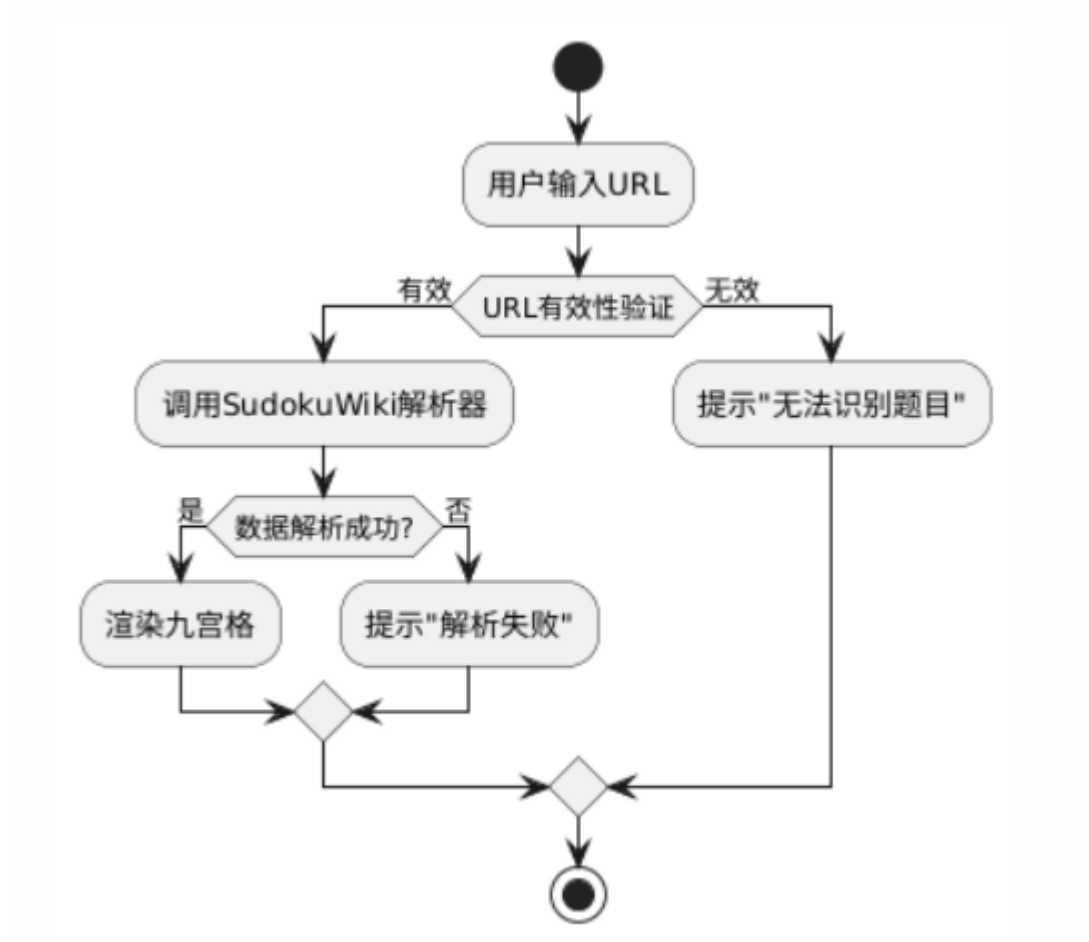
状态/条件	说明
	盘快照)
重做栈不为空	存在可重做的操作历史(栈存储被撤销的步骤)
“已回到初始状态”提示	撤销栈为空时触发, 阻止无效撤销
“无法重做”提示	重做栈为空时触发, 阻止无效重做

## (2) 交互流程

### 用户触发操作

- 。 点击撤销按钮: 尝试回退到上一步。
- 。 点击重做按钮: 尝试恢复到下一步。

## 2.5 题目导入活动图



说明：

### （1）核心流程

描述了用户通过 URL 导入数独题目的完整过程，涵盖验证、解析、渲染及异常处理。

### （2）步骤解析

步骤	说明	技术实现示例
用户输入 URL	用户粘贴或输入数独题目的外部链接（SudokuWiki）。	- 前端：<input type=“URL”> 控件 + 输入校验。

步骤	说明	技术实现示例
URL 有效性验证	系统检查 URL 格式是否合法。	<pre> Javascript if (!/^https:\/\/\/.test(URL)) alert("无效 URL"); </pre>
调用解析器	若 URL 有效，向后端发送请求，调用 SudokuWiki 解析器。	<p>后端 API：POST /api/parse-puzzle，参数为 URL。</p>
数据解析成功	解析器尝试从 URL 提取数独数据	<p>成功：返回 { success: true, board: [...] }。</p>
渲染九宫格	前端将解析后的数据渲染为可交互的数独棋盘。	<p>Svelte：{#each board as row} 动态生成网格。</p>
异常处理	失败时提示用户具体原因（URL 无效、解析失败等）。	<p>错误码映射：404 → “链接不存在”，500 → “服务器错误”。</p>

### 3. 总结

本报告通过系统化的 UML 建模，全面覆盖数独乐乐的需求分析、架构设计与核心流程，具体成果包括：

### （1）需求精准化

- 。用例图明确了用户与系统的 6 大交互场景（如提示、撤销、题目导入），功能边界清晰，避免开发歧义。
- 。通过活动图细化异常流程（如 URL 解析失败），确保鲁棒性设计。

### （2）架构可扩展性

- 。类图中策略模式的应用（策略引擎+工厂）使提示功能支持无缝扩展新算法，符合开闭原则。
- 。状态图揭示的双栈结构（撤销/重做）为历史管理提供标准化实现模板。

### （3）开发效率提升

- 。序列图明确提示功能的优先级调用链（基础策略→高级策略），指导代码模块化开发。
- 。状态图暴露的边界问题（如重置计时器）提前规避潜在缺陷。

### （4）技术验证

- 。序列图中 Web Worker 的潜在需求验证了前端性能优化方向。
- 。类图与 Svelte 状态管理的映射关系确保技术选型合理性。