



中山大學

SUN YAT-SEN UNIVERSITY

面向对象课程报告

OODA 逆向分析

姓 名 朱思源

学 号 24214593

学 院 计算机学院

专 业 计算机技术

2025 年 1 月 17 日

1 需求规格

1.1 愿景

开发一个具有不同难度级别、自定义谜题和社交分享功能的在线数独游戏平台，为用户提供多样化且富有挑战性的解题体验，支持答案提示、撤回与重做、笔记功能和计时功能等，以优化用户解题思路并提升解题效率。

1.2 用例分析

- 开始游戏：用户能够依据自身需求选定难度级别以生成随机棋盘，或者输入特定序列创建自定义棋盘，与此同时启动计时程序。
- 暂停游戏：用户触发暂停按钮后，棋盘与计时器进入冻结状态，点击恢复按钮则恢复至先前状态。
- 填写答案：用户于选定的空白单元格内填入数字时，将激活验证机制，其中涵盖即时错误反馈以及游戏结束判定提示等内容。
- 笔记记录：用户选定空白单元格后进入笔记模式，进行候选值标注，再次点击操作按钮则退出该模式。
- 选择单元格：用户选取单元格时，该单元格将以高亮形式呈现，并同时展示其所在行列及九宫格信息，若单元格内存在数字，则同步提示相同数字在其他位置的分布情况。
- 获得提示：用户选定空白单元格并点击提示按钮后，系统生成提示信息并在所选单元格上予以显示。
- 题目加载：用户进入系统界面后，可选择系统预设难度或进行自定义题目操作（通过 [sudokuwiki.org](https://www.sudokuwiki.org) 题目链接导入并验证其合规性），经系统处理实现数独问题可视化后进入解题界面；若题目不符合规则，则重新加载页面。
- 回溯结果：用户点击“undo”按钮后，系统依据游戏进程中用户的输入顺序，对解题关键步骤、错误节点或整体思路进行梳理复盘，并以可视化图表或文本形式呈现供用户查阅，同时对用户的查看操作予以记录。

1.3 领域模型

- Game：管理整局游戏状态（光标位置、难度、棋盘、计时等），协调游戏流程（开始、暂停等）。

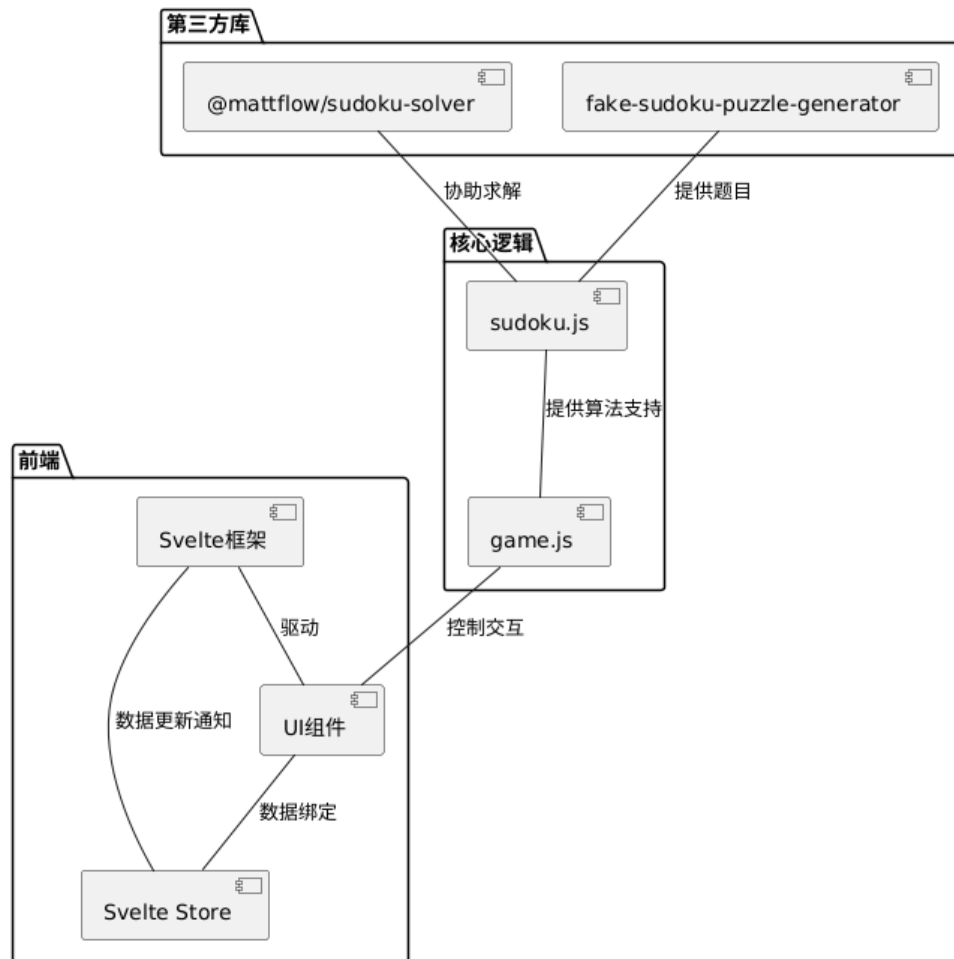


图 1: 技术架构

- Grid: 负责生成与管理游戏棋盘（题目棋盘、用户棋盘）及获取提示等操作。
- Candidates: 记录和管理每个单元格的候选值（添加、删除）。
- Cursor: 记录光标当前位置。
- Difficulty: 记录游戏难度级别。
- Hints: 管理提示次数。
- Keyboard: 记录和管理键盘状态。
- Modal: 控制模态窗口的显示与隐藏，用于展示设置、分享、提示等信息。
- Note: 管理笔记模式状态。
- Setting: 保存用户设置（如定时器显示、候选数字高亮等）。
- Timer: 负责游戏计时（开始、暂停、重置）。

2 软件设计规格

2.1 系统技术架构

如图1。本项目基于 Svelte 构建多层架构。前端借助 Svelte 框架，其响应式机制确保 UI 随数据即时更新，组件化设计将 UI 拆分为 Board、Controls 等独立组件，提升复用与维护性。Svelte Store 负责管理 grid、candidates 等全局状态，实现数据与 UI 双向绑定，降低耦合。核心逻辑依靠 JavaScript 模块，如 sudoku.js 处理数独生成与求解等算法，game.js 控制游戏流程。

2.2 对象模型

见图2.2。

2.3 设计改进建议

1. 抽象层优化

- 封装数独核心逻辑（求解、生成、验证）为独立类（如 SudokuPuzzle 类），提升模块化与复用性。
- 将候选数字、用户输入、提示、计时器等功能抽象为服务类（GridService、HintService、TimerService 等），通过接口与实现分离便于扩展和替换。

2. 依赖管理改进

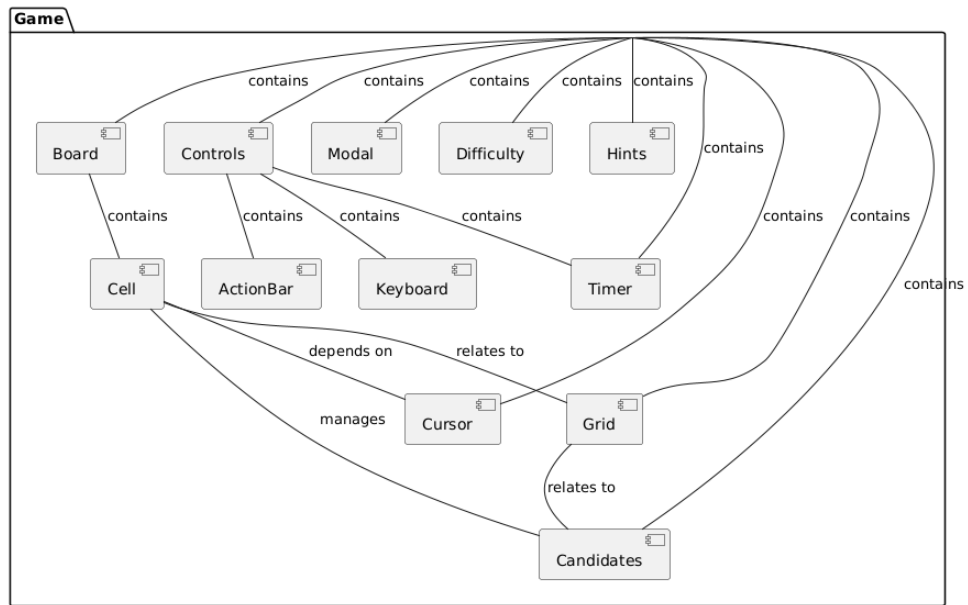


图 2: 对象模型

- 使用依赖注入管理类间依赖，如在 Game 类构造函数中注入相关服务类。
- 明确模块边界，避免逻辑分散在多个文件中，方便单元测试和维护。