

数独乐乐

用例分析

1. 用例：生成棋盘

用例名称：生成棋盘

参与者：玩家

简要描述：玩家通过菜单栏选择难度等级，系统根据所选难度生成符合规则的数独初始棋盘

前置条件：1. 游戏已成功启动；2. 玩家处于游戏主界面或当前游戏结束状态

后置条件：1. 成功：系统生成唯一解的数独棋盘，展示给玩家，游戏进入可操作状态；2. 失败：未生成棋盘，系统提示错误信息，玩家停留在原界面

基本事件流：

1. 玩家点击页面菜单栏
2. 系统弹出难度选择弹窗，展示可选等级：Very easy（极易）、Easy（简单）、Medium（中等）、Hard（困难）
3. 玩家选择其中一个难度等级
4. 系统根据所选难度生成含唯一解的数独初始棋盘（难度对应空格数量/推理复杂度）
5. 系统渲染新棋盘并展示，用例结束

特殊要求：生成的棋盘必须保证唯一解，且空格数量、推理难度与玩家选择的等级严格匹配

2. 用例：输入数字

用例名称：输入数字

参与者：玩家

简要描述：玩家选中数独空格后输入数字，系统验证数字是否符合数独规则并更新显示

前置条件：1. 数独棋盘已成功生成，游戏处于进行中；2. 数独谜题尚未解决；3. 玩家已选中待填充的空格

后置条件：1. 输入有效：选中空格更新为输入的数字，显示正常颜色；2. 输入无效：选中空格不更新或显示红色数字

基本事件流：

1. 玩家点击棋盘上的一个空格，该空格被选中（显示选中状态，边框高亮）
2. 玩家通过键盘输入数字 1-9 或点击页面数字键盘选择数字
3. 系统验证输入数字是否符合数独规则（当前行、当前列、所属九宫格内无重复数字）
4. 若验证通过，系统更新选中空格的显示内容为输入的数字（正常颜色）
5. 用例结束

异常事件流：输入数字无效

1. 偏离步骤：基本事件流的步骤3
2. 系统验证输入数字不符合数独规则
3. 系统将选中空格内的数字显示为红色（或保持原空白状态）
4. 用例结束

3. 用例：候选值笔记

用例名称：候选值笔记

参与者：玩家

简要描述：玩家开启笔记功能后，可在选中单元格添加/移除候选值（1-9），辅助解题，支持一键清空

前置条件：1. 数独谜题尚未完成；2. 玩家已开启笔记功能；3. 玩家已选中目标单元格

后置条件：所选单元格的候选值列表被正确更新（添加、移除或清空）

基本事件流：

1. 玩家选择一个未填充数字的单元格（或已填充但支持修改笔记的单元格）
2. 玩家通过页面数字键盘或键盘输入，选择 1-9 中的一个数字作为候选值
3. 系统判断该数字是否已在当前单元格的候选值列表中：
4. 若未存在，将其添加到单元格的笔记区域（小字体显示在单元格角落）
5. 若已存在，将其从候选值列表中移除
6. 若玩家点击「清空笔记」按钮，系统清空当前单元格的所有候选值
7. 用例结束

4. 用例：分享数独

用例名称：分享数独

参与者：玩家

简要描述：玩家点击分享按钮后，可选择数独识别码、推特、脸书、邮件、二维码等方式，分享当前数独棋盘给他人

前置条件：1. 数独棋盘已正确初始化；2. 游戏处于进行中或已结束

后置条件：1. 分享成功：系统生成并发送分享内容，玩家返回原游戏界面，可继续当前游戏；2. 分享失败/取消：无分享行为，玩家继续当前游戏

基本事件流：

1. 玩家点击页面的「分享」按钮
2. 系统弹出分享方式选择界面，列出所有支持的分享选项

3. 玩家选择其中一种分享方式（如二维码）
4. 系统根据选择的方式生成对应的分享内容（如二维码图片、含识别码的链接、预设邮件文本）
5. 玩家确认分享内容并触发发送操作（如扫码保存、发送邮件、发布社交动态）
6. 分享完成后，系统关闭分享界面，返回原游戏界面
7. 用例结束

异常事件流：玩家取消分享

1. 偏离步骤：基本事件流的步骤6
2. 玩家点击分享界面的「取消」按钮
3. 系统关闭分享界面，返回原游戏界面，不生成也不发送分享内容
4. 用例结束

5. 用例：管理系统个性化设置

用例名称：管理系统个性化设置

参与者：玩家

简要描述：玩家点击设置按钮后，可在弹窗页面中配置个性化选项（如是否显示计时器、是否启用高亮提醒、切换黑暗模式等），配置后保存生效

前置条件：玩家已进入数独游戏页面（主界面或游戏界面）

后置条件：1. 点击保存：系统保存所有配置项，配置即时生效，关闭设置弹窗；2. 点击取消：系统不保存任何修改，关闭设置弹窗，配置保持原有状态

基本事件流：

1. 玩家点击页面的「设置」按钮
2. 系统弹出设置弹窗，展示所有可配置选项及当前生效状态
3. 玩家对设置项进行个性化配置（如开启计时器、关闭高亮提醒、切换黑暗模式等）

4. 玩家点击弹窗中的「保存」按钮
5. 系统验证配置项合法性（默认均合法），保存所有修改到本地存储
6. 配置即时生效（如界面切换为黑暗模式、计时器开始显示）
7. 系统关闭设置弹窗，返回原页面
8. 用例结束

异常事件流：玩家取消设置

1. 偏离步骤：基本事件流的步骤4
2. 玩家点击设置弹窗中的「取消」按钮
3. 系统不保存任何配置修改，直接关闭设置弹窗
4. 返回原页面，所有配置保持修改前的状态
5. 用例结束

6. 用例：高亮提醒

用例名称：高亮提醒

参与者：玩家

简要描述

述：玩家已启用高亮提醒设置的前提下，选中单元格时，系统根据预设规则自动高亮相关单元格（同行/同列/同九宫格、相同数字、冲突数字）

前置条件：1. 玩家正在游戏中（棋盘已生成）；2. 玩家已在系统设置中启用「高亮提醒」功能；3. 玩家已选中目标单元格

后置条件：系统按预设规则完成高亮效果更新，选中状态取消后高亮效果消失

基本事件流：

1. 玩家点击棋盘上的任意单元格，该单元格被选中
2. 系统读取玩家的高亮提醒配置规则

3. 系统按规则应用高亮效果：
4. (可选) 高亮与选中单元格同行、同列、同九宫格的所有单元格
5. (可选) 高亮与选中单元格内数字相同的所有单元格
6. (可选) 高亮与选中单元格内数字存在冲突的所有单元格
7. 玩家取消选中 (如点击其他区域) 后，系统取消所有高亮效果
8. 用例结束

7. 用例：计时器

用例名称：计时器

参与者：玩家

简要描述：玩家启用计时功能后，游戏初始化时自动开始计时；玩家可点击计时器的暂停/开始按钮控制计时状态，暂停时隐藏所有单元格数字，开始时恢复显示

前置条件：1. 玩家已在系统设置中启用「计时功能」；2. 数独棋盘已成功初始化，游戏处于进行中

后置条件：1. 计时中：计时器持续显示累计时间，单元格数字正常显示；2. 暂停中：计时器停止计时，单元格数字隐藏，游戏暂停

基本事件流：

1. 数独棋盘生成完成，游戏进入可操作状态时，计时器自动开始计时（从 00:00 开始累计）
2. 计时器实时显示当前累计时间（格式：时:分:秒）
3. 玩家点击计时器上的「暂停/开始」按钮
4. 若当前为计时状态：系统停止计时，隐藏棋盘上所有单元格的数字，按钮切换为「开始」
5. 若当前为暂停状态：系统恢复计时，重新显示所有单元格的数字，按钮切换为「暂停」
6. 游戏通关或玩家退出游戏时，计时器停止计时
7. 用例结束

8. 用例：校验数独答案

用例名称：校验数独答案

参与者：玩家

简要描述：玩家填写完数独棋盘所有数字后，系统校验棋盘是否符合数独规则，根据校验结果给出对应提示（通关/错误）

前置条件：1. 数独棋盘已生成，游戏处于进行中；2. 玩家已填写完棋盘上的所有空格

后置条件：1. 校验成功：系统弹窗提示通关，提供「生成新游戏」入口，计时器停止；2. 校验失败：系统标红所有填写错误的单元格，计时器继续运行，玩家可继续修改答案

基本事件流：

1. 玩家填写完数独棋盘的所有空格
2. 系统校验整个棋盘的数字是否符合数独规则（所有行、列、九宫格内无重复数字）
3. 若校验通过（答案正确）：
 4. 系统弹窗提示：「恭喜通关！」，并显示本次游戏耗时
 5. 弹窗底部提供「生成新游戏」按钮
 6. 计时器停止计时
7. 若校验失败（答案错误）：
 8. 系统将所有填写错误的单元格标红（或标注错误标识）
 9. 系统弹窗提示：「答案存在错误，请修改后重新校验」
 10. 计时器继续运行，玩家可继续修改棋盘数字
 11. 用例结束

9. 用例：提示单元格数字

用例名称：提示单元格数字

参与者：玩家

简要描述：玩家选中单元格后，点击提示按钮可以获取当前单元格的正确答案。成功获取后，可用提示次数减 1

前置条件：1. 数独棋盘已生成，游戏处于进行中；2. 玩家选中一个单元格

后置条件：1. 提示成功，填入正确答案；2. 提示失败，不做任何处理

基本事件流：

1. 玩家选中数独棋盘的一个单元格
2. 系统根据当前棋盘求解选中单元格的正确答案
3. 若求解成功：
 4. 系统在选中位置填入正确答案
 5. 系统剩余可用提示次数减 1
6. 若求解失败：
 7. 系统不填入数字
 8. 系统剩余可用提示次数不变
9. 用例结束

10. 用例：资源集成

用例名称：资源集成

参与者：玩家

简要描述：玩家通过输入 sudokuWiki 的有效URL 或 sencode，导入自定义数独棋盘并生成游戏

前置条件：1. 游戏已成功启动；2. 玩家拥有指向有效数独资源文件的URL 或 sencode

后置条件：1. 成功：系统生成URL 或 sencode对应的自定义数独棋盘，进入可操作状态；2. 失败：未生成棋盘，系统提示错误，玩家停留在原界面

基本事件流：

1. 玩家点击菜单栏的「Enter Code」选项
2. 系统弹出URL 或 sencode输入弹窗
3. 玩家输入URL 或 sencode并点击确认
4. 系统根据URL 或 sencode下载并解析数独资源文件
5. 系统生成对应的数独棋盘并渲染展示，用例结束

异常事件流：URL 或 sencode无法解析

1. 偏离步骤：基本事件流的步骤4
2. 系统捕获解析失败异常，禁用开始游戏按钮
3. 玩家可选择重新输入URL或点击取消关闭弹窗
4. 用例结束

11. 用例：撤销（Undo）

用例名称：撤销（Undo）

参与者：玩家

简要描述：支持玩家撤回最近一次有效的数字输入操作，可连续撤销历史输入

前置条件：1. 游戏处于进行中；2. 玩家已进行至少一次有效的数字输入操作

后置条件：1. 成功：最新一次数字输入被撤销，单元格恢复输入前状态，界面同步更新；2. 失败：无撤销记录时，撤销按钮置灰不可点击

基本事件流：

1. 玩家点击界面的「撤销」按钮
2. 系统回溯至前一次操作状态，移除最后输入的数字

3. 系统更新棋盘界面显示，用例结束

异常事件流：无可用撤销记录

1. 偏离场景：前置条件不满足（无有效输入记录）
2. 系统将「撤销」按钮置灰，禁止点击
3. 无界面状态变更，用例结束

12. 用例：重做（Redo）

用例名称：重做（Redo）

参与者：玩家

简要描述：玩家执行撤销操作后，可重新应用最近一次被撤销的数字输入，可连续重做未被覆盖的撤销记录

前置条件：1. 游戏处于进行中；2. 玩家已执行至少一次撤销操作，且无后续新输入操作

后置条件：1. 成功：最近一次被撤销的数字输入重新应用，界面同步更新；2. 失败：无可用重做记录时，重做按钮置灰不可点击

基本事件流：

1. 玩家点击界面的「重做」按钮
2. 系统重新应用最后一次被撤销的数字输入操作
3. 系统更新棋盘界面显示，用例结束

异常事件流：无可用重做记录

1. 偏离场景：前置条件不满足（无撤销记录/存在后续新输入）
2. 系统将「重做」按钮置灰，禁止点击
3. 无界面状态变更，用例结束

13. 用例：下一步提示

用例名称：下一步提示

参与者：玩家

简要描述：系统基于数独推理策略，为玩家提示下一个可填写单元格及对应推理依据

前置条件：1. 游戏处于进行中；2. 数独谜题尚未完成

后置条件：1. 成功：高亮显示所有仅剩单个候选值的单元格，展示对应推理策略及线索；2. 失败：无符合条件单元格时，系统提示无有效线索

基本事件流：

1. 玩家点击界面的「下一步提示」按钮
2. 系统筛选并高亮显示所有仅剩一个候选值的单元格
3. 玩家点击任意一个高亮单元格
4. 系统展示该单元格的候选值及推理依据（所用数独策略）
5. 用例结束

异常事件流：无仅剩单个候选值的单元格

1. 偏离步骤：基本事件流的步骤2
2. 系统弹窗提示：「当前无符合条件的候选值单元格，无法提供提示」
3. 无界面高亮变更，玩家可继续解题，用例结束

14. 用例：探索回溯

用例名称：探索回溯

参与者：玩家

简要描述：玩家在多候选值单元格（分支点）选择数字探索解题，遇错误时可一键回溯至最近分支点，删除探索过程的输入并更新候选值

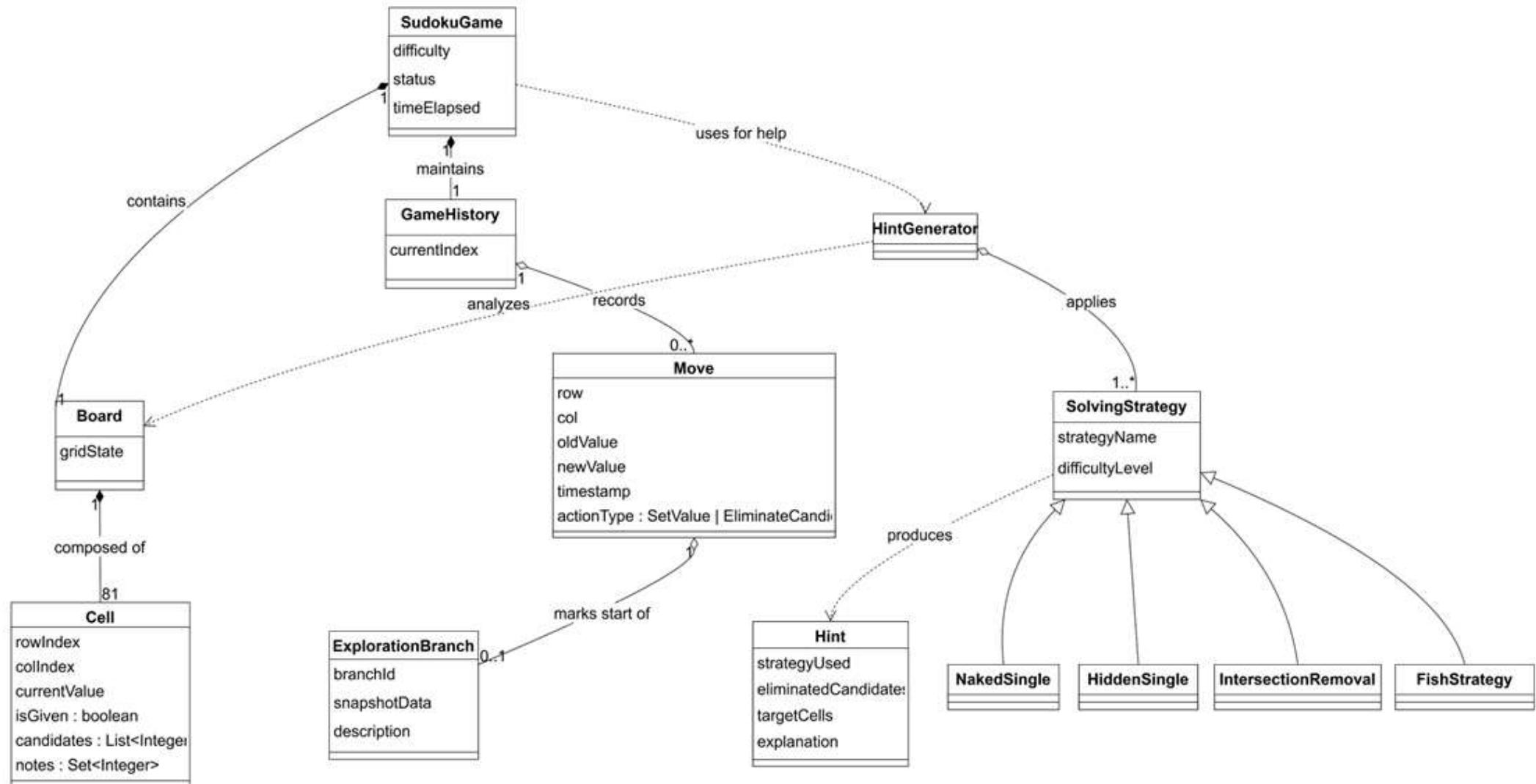
前置条件：1. 游戏处于进行中；2. 数独谜题尚未完成；3. 玩家已选中多候选值的空格（候选值数量>1），且已添加候选值笔记

后置条件：1. 探索中：系统存储分支点信息，按玩家选择的候选值推进解题；2. 回溯后：恢复至最近分支点状态，删除探索输入，更新候选值笔记

基本事件流：

1. 玩家选中候选值数量>1的空格
2. 系统自动存储该空格的位置及未选择的候选值（记录分支点）
3. 玩家从候选值中选择一个填入空格，开始探索解题
4. 系统自动填充当前存在唯一候选值的单元格
5. 若探索过程中出现候选值违反数独规则的情况，玩家点击「回溯」按钮
6. 系统恢复至最近分支点状态，删除探索过程中填入的所有数字
7. 系统将分支点空格中已探索的候选值从笔记中移除并更新界面，用例结束

领域模型



系统技术架构

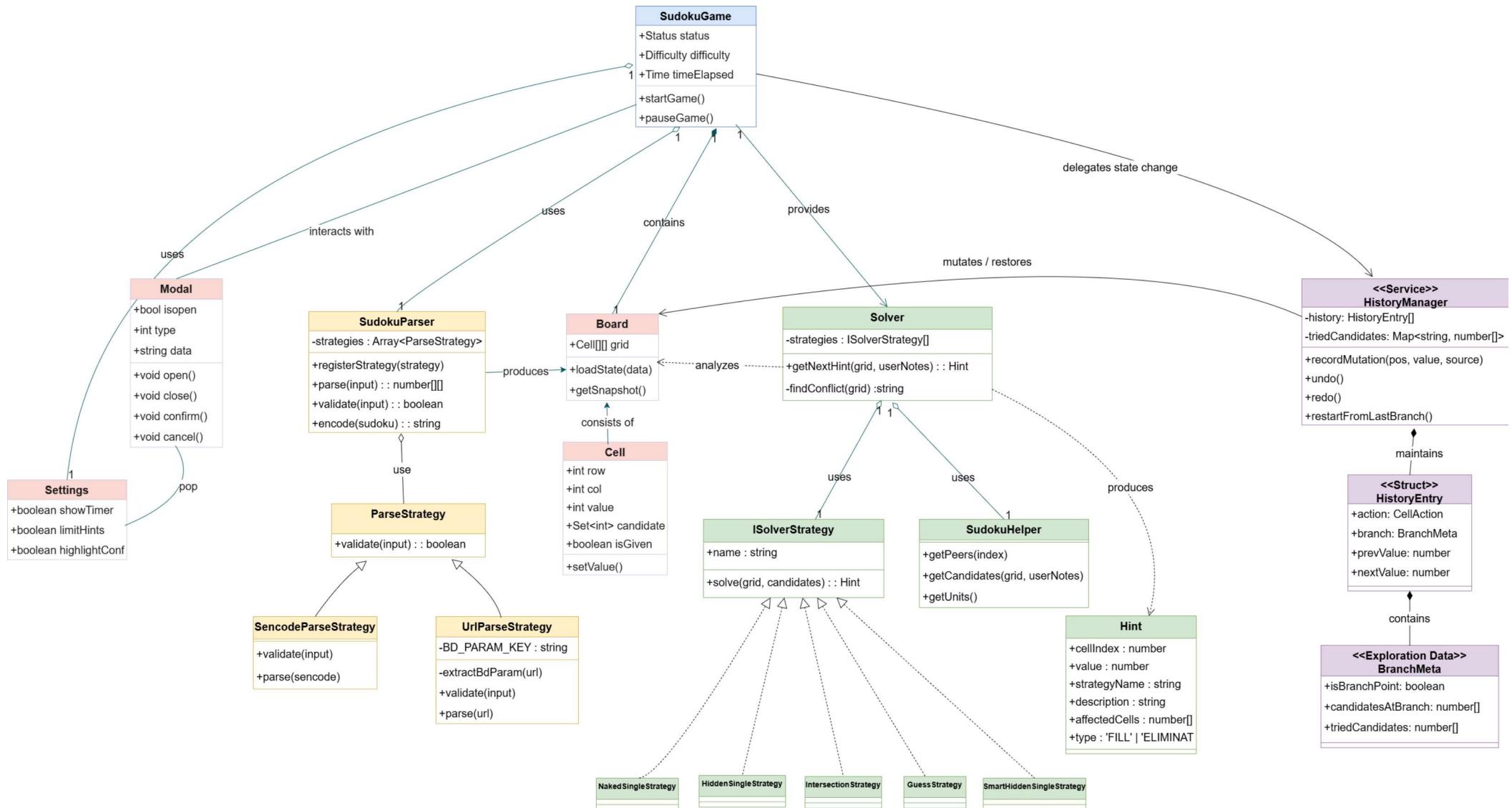
本项目为一款前端主导的轻量级数独游戏应用，基于Svelte框架构建，整体采用分层架构模式。应用以组件化、模块化设计为核心，将UI渲染、状态管理、游戏核心逻辑及工具函数分层解耦，实现高效开发与灵活扩展。核心功能模块为：

- 数独核心逻辑模块：封装棋盘生成、合法性验证、自动求解等核心算法，为游戏运行提供底层逻辑支撑。
- 状态与交互管理模块：基于事件驱动模型，实现游戏状态的实时维护与用户操作的精准响应。
- 配置与可视化扩展模块：提供个性化配置项与可视化功能支持，适配不同用户的使用习惯与体验需求。

整体架构模式采用分层架构模式，自上而下分为应用层、组件层、工具层，各层级职责边界清晰，通过接口化交互实现低耦合、高内聚的架构设计：

- 应用层
 - 游戏状态维护：基于Svelte提供的Store机制（含writable、derived等API），实现全局游戏状态的统一管理、实时更新与跨组件共享，确保状态一致性与可追溯性。
 - 数独核心逻辑：封装棋盘生成、数据合法性验证、自动求解等核心算法，提供标准化接口供上层调用，支撑游戏核心玩法实现。
 - 用户事件处理：统一监听并处理用户点击、数值输入、操作取消等交互事件，转化为对应业务逻辑指令，驱动游戏状态变更与UI反馈。
- 组件层
 - 游戏棋盘组件：负责数独棋盘的可视化渲染，绑定用户交互事件（单元格选中、数值输入等），实现游戏核心操作界面功能。
 - 配置设置组件：提供个性化显示参数配置功能，支持计时器开关、主题切换等选项的灵活控制。
 - 分享功能组件：封装棋盘分享能力，实现游戏状态（棋盘数据）与分享码的双向编码、解码，支持快速分享游戏进度。
- 工具层
 - 数独算法工具：独立封装棋盘生成、合法性验证、自动求解等核心逻辑，提供标准化工具接口，供应用层调用。
 - 分享编码工具：实现棋盘数据与分享码的双向转换（编码/解码），保障分享功能的高效、准确实现。
 - 时间管理工具：封装计时器核心功能，支持计时启停、重置、时间格式化展示，满足游戏计时需求。

对象模型



设计说明

高度可扩展性

SudokuParser和Solver两个核心模块均采用了策略模式。如果未来需要支持新的输入格式（如通过图片识别数独）或增加更多的数学解题算法，只需新增策略类而无需修改核心控制逻辑。这种设计符合开闭原则。

单一职责原则

每个类只负责一项职责。如 SudokuParser 负责资源解析，Solver 负责根据当前棋盘生成提示，Timer 管理时间，Settings 管理设置等。通过职责分离，简化了类的复杂性，便于维护和扩展。

状态管理与业务逻辑解耦，便于维护

使用备忘录模式实现的撤销/重做/回溯功能，能在不暴露对象内部细节的前提下，捕获并保存对象的状态，后续需要时可将对象恢复到被保存的状态。

策略模式

预留策略扩展点，在 ParseStrategy 接口中定义标准化方法（parse、validate），后续新增解析类型（如本地文件、二维码内容）时，仅需新增策略类并注册至工厂，无需修改核心流程。统一管理 URL 解析、sencode 解析等策略实例，根据玩家输入类型（URL/sencode）自动匹配对应解析策略，避免应用层直接创建具体策略对象，降低耦合。

依赖倒置原则

- 定义抽象接口：在工具层定义数据解析、数独算法等抽象接口，明确方法契约。
- 底层实现面向接口编程：URL 解析、sencode 解析类实现数据解析接口；各类数独算法类实现数独算法接口。