

小黄的奇妙探险 - 开发文档

1. 项目背景

“小黄的奇妙探险”是一个基于控制台的迷宫冒险游戏项目。这是我在课程作业中开发的一款小游戏，通过设计迷宫、实现玩家移动和游戏状态管理，结合数据结构的实际应用，来进一步巩固对编程基础的理解。本项目同时融入了撤销与恢复功能，帮助玩家更好地体验游戏，也提升了项目的技术深度。

2. 项目目标

本项目的开发目标可以概括为以下几点：

1. 实现一个功能完整的迷宫游戏：

- 玩家控制小黄在迷宫中移动并找到所有宝藏。
- 游戏中包含陷阱等障碍，需要玩家合理规划路径。

2. 提供撤销与恢复功能：

- 允许玩家撤销最近一步操作，也可以恢复撤销的步骤。
- 确保撤销与恢复操作不会影响游戏状态的一致性。

3. 探索数据结构和动态内存的实际应用：

- 使用栈实现撤销与恢复功能。
- 动态调整内存大小以存储玩家路径，避免固定容量的限制。

通过开发本项目，我希望不仅能实现游戏功能，还能加深对数据结构、动态内存管理和游戏逻辑实现的理解。

3. 开发环境

- 操作系统:** Windows 10
- 开发工具:** Visual Studio Code (VSCode)
- 编程语言:** C

4. 功能实现

4.1 基础功能

1. 地图加载与初始化：

- 游戏包含两个预定义关卡，初始化时加载地图和玩家初始状态。
- 支持关卡选择，玩家可以自由选择挑战关卡。

2. 玩家控制与状态更新：

- 玩家可以使用 **W**、**A**、**S**、**D** 键进行上下左右移动。
- 游戏会实时显示玩家体力消耗、路径信息和当前状态。

3. 游戏结束条件：

- 玩家找到所有宝藏即完成游戏。
- 玩家可以随时主动结束游戏，或因非法输入终止游戏。

4.2 附加功能 - 撤销与恢复

1. 撤销功能：

- 玩家按下 **Z** 键可以撤销最近一步操作。
- 撤销操作会恢复玩家位置、体力消耗及地图状态。

2. 恢复功能：

- 玩家按下 **Y** 键可以恢复最近撤销的操作。
- 恢复操作会同步更新玩家状态和地图信息。

3. 陷阱状态管理：

- 全局变量 `flag` 记录玩家是否进入陷阱。
- 无论撤销还是恢复，陷阱状态会根据玩家位置正确复原。

4. 动态内存分配：

- 为玩家路径和栈操作分配动态内存，确保在长时间游戏中不会因容量限制导致崩溃。

5. 使用说明

5.1 游戏启动

1. 启动程序后，选择关卡：

- 使用 **W** 和 **S** 键上下移动选择。
- 按下回车键确认选择。

2. 选择模式：

- **实时模式 (0)**：逐步输入移动指令。
- **编程模式 (1)**：一次性输入一串指令。

5.2 操作方式

• 移动控制：

- **W**：向上移动
- **A**：向左移动
- **S**：向下移动
- **D**：向右移动
- **I**：原地不动

• 撤销与恢复：

- **Z**：撤销最近一步操作。
- **Y**：恢复最近撤销的操作。

• 游戏结束：

- **Q**: 主动退出游戏。

5.3 注意事项

1. 每次移动会消耗玩家的体力值，进入陷阱会增加额外消耗。
2. 原地不动（按 **I**）不会影响地图状态，但仍记录为操作的一部分。
3. 游戏结束后，会显示玩家的总体力消耗及完整路径信息。

6. 代码结构

6.1 模块划分

1. 地图管理模块：

- `init_map`: 初始化地图关卡。
- `render_map`: 渲染地图信息。

2. 玩家状态模块：

- `init_player`: 初始化玩家状态。
- `move_player`: 更新玩家位置。
- `energy_sum`: 计算并更新体力消耗。

3. 撤销与恢复模块：

- 栈结构 `ur_stack` 用于存储玩家操作记录。
- `push` 和 `pop` 实现栈操作。
- `clear_stack` 和 `free_stack` 负责内存管理。

4. 游戏逻辑模块：

- `game_loop`: 实现游戏主循环。
- `render_info`: 显示玩家当前状态信息。
- `render_start` 和 `choose_level`: 实现关卡选择界面。

6.2 代码分布

本项目代码集中在一个文件中，主要模块用函数进行划分，清晰直观。未来可进一步优化为多个源文件。

7. 技术亮点

1. 数据结构的应用：

- 采用栈实现撤销与恢复功能，利用其先进后出的特性实现操作的有序管理。

2. 动态内存分配：

- 为路径信息和栈操作分配动态内存，避免容量限制，确保游戏的稳定性。

3. 陷阱状态的统一管理：

- 使用全局变量 `flag` 记录玩家是否处于陷阱状态，简化复杂的状态更新逻辑。

4. 人性化设计：

- 提供两种模式供玩家选择，适应不同的操作需求。

8. 未来改进

1. 关卡扩展：

- 支持从外部文件动态加载关卡，增加游戏的多样性。

2. 界面优化：

- 提供更直观的界面显示，如实时高亮玩家位置和路径。

3. 性能提升：

- 优化动态内存管理，减少频繁分配的性能损耗。

9. 总结

“小黄的奇妙探险”通过实现基本的迷宫冒险功能以及撤销与恢复功能，展示了数据结构和动态内存管理在游戏开发中的应用。通过本项目，我不仅强化了编程实践能力，也加深了对游戏逻辑实现的理解。希望未来能进一步完善本项目，为玩家提供更丰富的游戏体验。