Wordle游戏项目报告

程序结构说明

本项目是一个基于Rust语言实现的Wordle猜词游戏，包含以下主要模块：

主要文件结构

- `main.rs`：程序入口点，处理命令行参数和游戏主循环
- `function.rs`：核心游戏逻辑和功能实现
- `solver.rs`：附加的求解器功能，提供游戏辅助
- `builtin_words.rs`：内置单词列表（未展示内容）

架构概述

程序采用模块化设计，将游戏逻辑(`function.rs`)、主程序流程(`main.rs`)和求解器功能(`solver.rs`)分离，提高了代码的可维护性和可读性。游戏状态使用Serde库进行序列化和反序列化，支持游戏的持久化。

游戏主要功能说明

1. 基本游戏玩法

- 玩家有6次机会猜测一个5字母单词
- 每次猜测后，系统会给出颜色反馈：
  - 绿色(G)：字母正确且位置正确
  - 黄色(Y)：字母正确但位置错误
  - 红色(R)：字母不存在于目标单词中

2. 特色功能

- **多种游戏模式**：支持随机模式、指定单词模式和每日挑战模式
- **难度选择**：困难模式增加了猜测限制，必须使用已揭示的提示信息
- **数据持久化**：游戏状态可保存到文件，支持继续游戏

- **统计功能**：记录游戏数据并显示胜率、平均尝试次数等统计信息
- **自定义词集**：支持使用外部词集文件进行游戏

3. 游戏界面示例

```
root@lpl717:~/wordle-lpl24# cargo run -- -r -t -v
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.11s
     Running `target/debug/wordle -r -t -v`
give me your guess (1 times):aeros
AEROS
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Solver mode active.
Type 'left' to show remaining words, Type 'rec' to show recommend words
rec
rec start
------------------
Top 5 recommended words:
1. RAGDE (Score: 3.63)
2. CADRE (Score: 3.59)
3. RADGE (Score: 3.57)
4. RATED (Score: 3.53)
5. CATER (Score: 3.51)
give me your guess (2 times):ragde
AEROS RAGDE
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Solver mode active.
Type 'left' to show remaining words, Type 'rec' to show recommend words
rec
rec start
------------------
Top 5 recommended words:
1. CREAM (Score: 3.55)
2. BREAM (Score: 3.50)
3. CRENA (Score: 3.41)
4. CREAK (Score: 3.38)
5. BREAK (Score: 3.34)
give me your guess (3 times):cream
AEROS RAGDE CREAM
ABCDEFGHIJKLMNOPQRSTUVWXYZ

You are right! The answer is cream
CORRECT 3
```

游戏运行界面，显示猜测历史、颜色反馈和键盘状态

求解器功能实现（附加功能）

1. 求解器算法设计

求解器采用基于信息熵的单词推荐算法：

```
for guess in &search_set {
    let mut partition_sizes = HashMap::new();
    for answer in &remaining_words {
        let state = function::color_state(guess, answer);
        *partition_sizes.entry(state).or_insert(0) += 1;
    }

    let mut score = 0.0;
    let total_size = remaining_words.len() as f64;
    for size in partition_sizes.values() {
        let p = *size as f64 / total_size;
        if p > 0.0 {
            score += p * p.log2();
        }
    }
    scores.push((guess.clone(), -score));
}
```

2. 求解器功能特点

- **剩余单词计算**：根据已有猜测和反馈，筛选可能的答案单词
- **智能推荐**：基于信息熵推荐能最大程度减少不确定性的单词
- **交互式界面**：支持多种命令：
  - `rec`：获取推荐单词
  - `left`：显示剩余可能单词
  - `win`：标记游戏胜利
  - `quit`：退出求解器