Rust—Wordle大作业报告

Part 1.代码结构功能

大作业代码全部位于src/main.rs文件下，起始为use代码块，使用的库包括标准库std，命令行clap，随机数rand，数据输入text.io，Json库serde\_json，颜色输出console。其次为结构体（类）定义，代码中均有相应注释：

Args类：命令行参数定义，使用clap宏用于解析

WordDict类：重命名的String与i32元组，代表使用单词及其出现次数

WordEntropy类：重命名的String与i64元组，代表词库中单词及其信息熵（扩大10^9倍后截断小数点后的结果）

（以上类均定义Ord与Eq宏以及PartialEq,PartialOrd特型满足BinaryHeap的排序需求）

Game/User类：与需要读取导入的Json文件结构一致，用于加载Json文件，User由整型total\_rounds以及Game类向量games构成

最后为函数定义，代码中亦有相应注释：

information\_entrophy：导入String的集合，基于该集合及信息熵算法计算集合中每一个词的信息熵，返回自动按信息熵大小排序的二叉堆（优先队列）

random\_spawn：基于轮数day，随机种子seed和词集大小size生成随机的index

read\_to\_list：从文件中按行读取单词加载至集合中，包含非法判断

read\_from\_file\_user：加载Json文件并将信息存入User类对象

read\_from\_file\_config：加载Json文件并将信息存入Args类对象

game\_round\_normal: 交互模式的一局游戏，内含以下代码块：

导入文件词库并作合法性检查

答案指定（制定答案模式、随机模式、用户输入）

加载状态Json文件

answer输入—判断—结果输出

信息熵集更新

将该轮信息导入Json文件

返回该轮具体信息（获胜情况，猜测次数等）

game\_round\_test: 测试模式的一局游戏，所含功能与交互模式除不含信息熵计算排序部分以及输出至终端信息不同外基本一致，故不再赘述

game\_round\_automatic：自动对词集进行平均猜测次数的计算以及输出

game\_round：根据is\_tty决定game\_round的具体模式（test，normal）

main：主函数，内含以下代码块：

命令行及config文件解析参数

开启一轮游戏

加载State文件前几轮信息

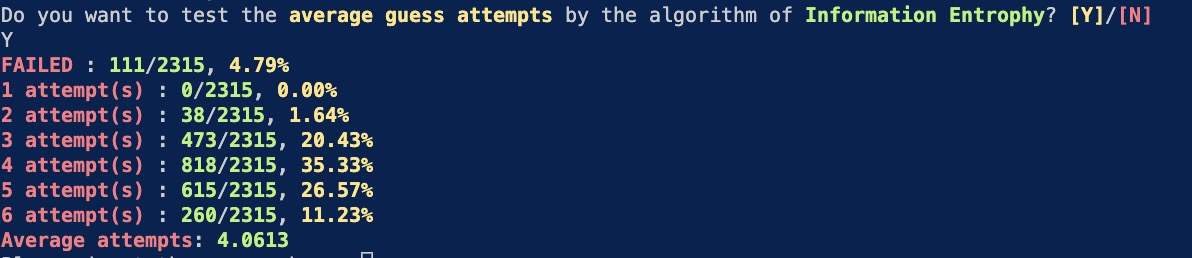
根据该轮游戏返回的信息计算Statistics并按模式分别输出

判断是否开始下一局游戏

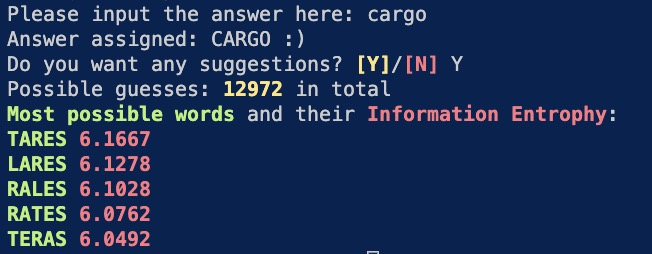
Part 2.游戏主要功能说明

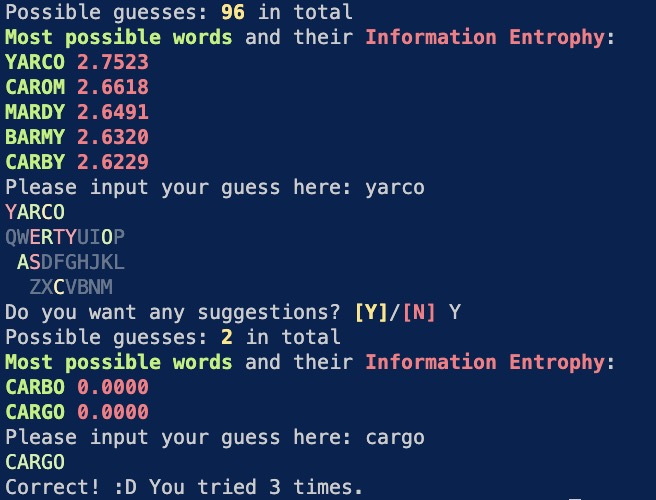
游戏实现基础部分的所有要求，对于需要的输入均有较为详细的说明，故仅展示部分截图。

首先为提高功能：

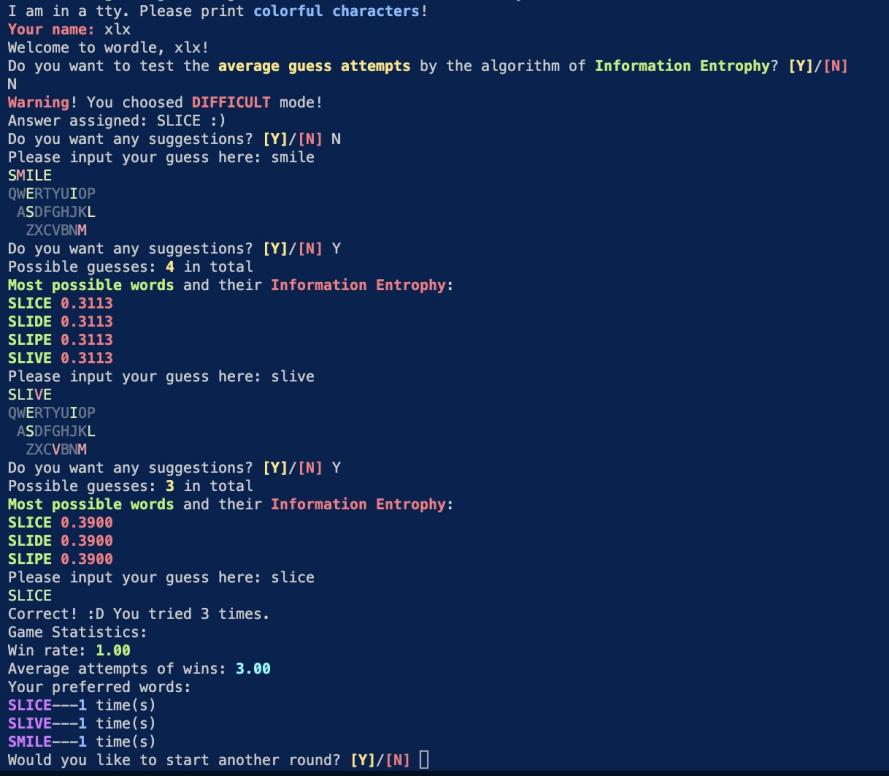


首先询问是否需要测试该词库的平均猜测次数，以上为标准答案词库和候选词库的平均猜测次数，经过数小时的计算得出。

制定答案后，给出可能的猜测词数（初始显然为12972即总词库词数），最有可能作为初始猜测词的五个单词及其信息熵，图中可见为tares, lares,rales, rates, teras五个单词。



距离制定答案为cargo，第一次猜测tares，剩余答案仅剩96个单词，其中yarco为这96个单词中信息熵重新计算后的最大者，猜测yarco后仅剩2个单词，最后猜出cargo为正确答案。



其次展示基础部分的功能：

（-t -r --seed 114514 --day 7 -D）

自动设置答案Slice

困难模式警告

输入单词后输出结果

+键盘区各字母状态

结束输出胜率

最常使用单词——

slice/slive/smile

Part 3.扩展功能实现

扩展功能使用3B1B的算法计算信息熵，具体过程为对每一个目标词遍历全部词库，判断其属于共243种不同颜色状态之中的哪一种状态，然后利用信息熵公式将每一种状态的信息熵相加得到结果。由于未用并行加速，耗时稍长，故将愿词库12972个单词的信息熵打表排序后存入文件（见src/acceptable.json）避免默认词库时较长的加载计算时间。



Part 4.感想

Rust相较C++关于所有权的明确规定导致出现变异错误的几率远高于C++（悲），但鲜有运行错误出现，且针对错误有人性化的解决方案提供，减小了程序猿debug的严重负担（？），但其缺乏面向对象的封装手段也会造成一定的困扰。

在实现大作业的过程中，能够体会到自己通过阅读文档和搜集教程来掌握Rust不同库功能的过程，也有帮助其他同学解决库功能学习上困扰的经历，通过wordle游戏的制作算是掌握了Rust较为基础的一些操作，但对于UI和Web的制作还是缺乏相应的经验，在今后的开发过程中应早做准备。