高级编程技术实验报告

实验三：Word Game

数据科学与计算机学院　17大数据与人工智能

17341015　陈鸿峥

一、 问题描述及求解思路

实现一个单词游戏(word game)。每轮游戏会发给玩家几张字母牌，通过组合字母牌上的字母组合出单词。

1. 单词分数

get_word_score将单词word转为小写后，遍历其中的每一个字母，并在SCRABBLE_LETTER_VALUES中获取该字母的分数。将分数累加即得该单词的得分。

2. 处理手牌

update_hand用于出了一次完整单词后更新手牌内容及数目。将遍历打出的单词word中的每一个字母，通过get函数获得其是否在手牌中出现，以避免出现错误情况（即0张手牌但没有从字典中清除的情况）。如果出现，则对该字母对应的手牌数减1。

3. 单词合法性

is_valid_word先将word中的字母全部转为小写，然后遍历word_list，判断该小写单词是否在单词表中。另外，还需用类似update_hand的方法判断word中的字母是否全部由原有的手牌构成。

4. 通配符

注意通配符(wildcard)只能替代元音字母。

is_valid_word中添加通配符的判断：如果存在通配符，则遍历所有元音字母，并替换掉通配符，尝试匹配单词表word_list中的单词。若对于所有元音字母替换，都找不到对应的单词，则说明word不是一个合法的单词。如果没有通配符，则代码实现逻辑与上题相同。

deal_hand按照题目叙述的方式进行实施，减少一个元音位置，留给通配符使用。注意要在程序引言区添加WILDCARD变量。处理hand时同样用了get方法字典匹配不成功报错。

5. 出牌实施

游戏规则见题目要求，在此不再赘述。

play_hand函数具体实施如下

1. 初始化分数计数器total_score为0
2. 当手上还有牌时，进入while循环

3. 用display_hand展示当前手牌

4. 请求用户输入，如果输入为!!则结束游戏，否则继续进行

5. 用is_valid_word判断用户单词是否合法，如果合法则得分get_word_score，否则输出错误提示信息

6. 用update_hand更新手牌，回到2（注意即使不合法的单词也会消耗手牌!）

7. 如果手上没牌或得到!!输入，则游戏结束

8. 输出用户最终得分

6. 完整游戏实施

substitute_hand用于替换手牌，方法与生成手牌的方法类似，即利用random.choice方法随机生成。然后将新生成的手牌结如其中，将原指定手牌丢弃。注意是替换字母，而不是替换单一手牌，即如果一个字母对应的手牌有多张，则需要全部替换。

play_game的实施如下

- 用户输入玩牌轮数n_hands
- 在每一手牌开始之前询问用户是否需要替换手牌，如果需要则询问需要替换哪一张牌，且此操作只能被使用一次，用flag_sub记录。
- 每一轮手牌结束之后询问用户是否回放上一轮的手牌，如果回放则不给予用户新的手牌。该操作只能用一次，用flag_replay记录，但不计入用户的玩牌轮数，且不再拥有替换牌的权利。

二、 代码

代码实施及注释请见附件ps3.py。

三、 运行截图

实验运行结果如下面几幅图片所示，且**测试了不同样例，以针对不同分支**，确保结果的正确性和鲁棒性。

图 1: 测试样例全部通过

图 2: 完整游戏

图 3: 自己添加：通过命令行传递参数指明手牌数目，如此图中传递的是10