背单词助手
概要设计

计 53 范舒翼 2015011302

计 53 吴昊哲 2015011297

1、目标

提供一个能跨平台的，具备背单词、词汇量测试、生词统计、查词功能的英语单词学习软件。

2、总体功能概述

（1）用户系统：用户可以选择注册新用户或登录已有账户，注册新用户时会要求用户选择默认背单词模式和英语水平，这些信息会被存储在用户文件中。

（2）背单词: 可以通过选择题与自觉模式两种方式背单词，用户可以根据自身的情况选择背多少单词。背单词方式根据用户注册时的设置初始化，并且可以根据用户的需求进行调整。用户每背完一个单词，可以选择向该单词加入例句，并且在该单词背诵错误时，会显示例句。在用户背单词结束后将根据正确率自动调整下一次背单词时单词的难度。

（3）词汇量测试：测试开始前，会要求用户选择测试方式（选择题或判断题），测试开始时，首先会测试难度较低的单词，根据用户答题情况调整单词难度。最终将反馈给用户所测得的词汇量级别。

（4）生词统计： 用户给定需要查询的文件，并输入文件名。根据用户的英语水平和单词背诵情况智能分析，自动筛选出生词，给出生词的解释（重复的生词仅出现一遍）。

（5）查词：用户可根据自己的实际需求查询单词，且会保留用户的最后 15 条查询信息，用户可以手动清除历史查询记录。

3. 整体架构

（1）概述：软件分为 3 个模块，数据库模块、功能模块、交互模块。其中数据库模块负责单词库与用户数据的导入与导出；功能模块实现了各项功能的核心算法；交互模块负责接收输入信息，将信息传递给功能模块，并接收功能模块要输出的信息，对其进行输出。

（2）数据库模块：数据库模块由 Single_Word, DataBase, User 三个类组成。其中 Single_Word 负责每个单词的难度、释义、例句的存储，并向外部提供获取这些信息以及对该单词添加例句的接口。Database 负责从 dict.txt 文件中读入单词,并将其存储在一个由 Single_Word 组成的 vector 中，向外部提供根据索引获取单词以及根据单词获取相关信息的接口。User 则负责用户数据的导入与导出，存储用户信息，并提供获取与改变用户信息的接口。

（3）功能模块:

<1>背单词模块：背单词模块由基类 Memory_Strategy 与子类 Memory_Strategy_Shanbay 和 Memory_Strategy_Towords 组成，其中基类负责从单词库中抽取一个单词的队列，以及更改当前队首单词的信息，决定是否将队首 pop，基类同时也为与交互模块交互信息提供了接口。子类负责对对应的单词生成题目格式与答案。

<2>词汇量测试模块：词汇量测试模块由基类 Test_Strategy 与子类 Test_Strategy_Multi 和 Test_Strategy_TF 组成，其中基类负责对当前这次词汇量测试进行初始化，实时维护用户当前体现出来的英语水平，并抽取一个符合用户水平的单词，将用户的反馈和正确答案进行比对，根据结果更改相应参数。子类则根据指定单词生成相应题面。

<3>生词统计模块：生词统计模块由基类 NewWords_Strategy 与子类 NewWords_Strategy_File 组成，基类根据得到的文章，将单词从文章中分离出来，对每个单词进行分析，并将统计结果提交给交互模块。子类根据文件名从相应的文件中读入要查询的文章。

<4>查词模块：查词模块因较为简单，可扩展性相对不高，因此直接在 Console 中实现。

（4）交互模块:

交互模块由 Console 与 Output 组成，Console 负责调用 Output 类给出提示信息，根据用户的选择，执行相应功能。具体的实现任务由 Console 调用功能模块完成。Output 有基类和子类，Output 负责将信息反馈给用户，其中 Output_Screen 的实现是将输出信息打印到屏幕上。

4. 详细设计

（1）数据库模块

<1>模块描述、功能

见整体架构

<2>Single_Word 类

接口：该类提供的接口有 Get_Word() 获得存储的英文，Get_Chinese() 获得存储的释义，Get_Difficulty() 获得难度，Add_Example()给该单词加入例句，Get_Example()获得该单词例句。

<3>Database 类

接口:Get_English(int number)根据索引返回对应的英文单词，Exist(string)查找某单词是否存在,Get_Chinese(string)返回某个单词的中文释义，如果不存在返回空串，Get_Difficulty(string)返回某个单词的难度，如果不存在返回 0，Get_Examples（string）返回某个单词的所有例句，Add_Example(string a, string b)给某个单词 a 添加给定的例句 b，Get_Words_Size()返回单词库的单词数。

有关算法:查找时由于数据库中单词是按照字典序有序存储，在查询时使用二分查找算法，每次查找的复杂度为 O(lgN)。

<4>User 类

接口:Get_Difficulty() 获得用户设定的难度，Get_Memory_Strategy()获得用户设定的背单词策略，返回一个 int,1 表示扇贝模式，2 表示百词斩模式。Get_Search_History()获得用户的查询历史，Get_Memorized_Times(string)获得用户背诵该单词的次数以及正确次数，Whole_Memorized_Words()返回用户背诵过的所有单词以及背诵情况，Add_History(string) 在用户文件中增加搜索历史，Change_Memory_Strategy_Number(int)在用户文件中改变记忆策略，Change_Difficulty_of_User(int) 在用户文件中改变难度，Change_Memorized_Tiems(string, int a, int b)在用户文件中改变 string 的背诵正确次数 a 与背诵次数 b。

（2）功能模块

<1>模块描述、功能

见整体架构

<2>Memory_Strategy 类及其子类

基类接口： Get_Words_Queue() 获得此次背单词的题目。After_Factory()在背完单词后改变和该单词有关的信息，并决定是否将该单词 pop 出队列。Run()在背完所有单词后动态调整用户设定难度。Init(int)在背单词前接收外界传入的今日背单词数目，并执行一些初始化工作。Exist()判断是否还需要背单词。First_Word()返回当前需要背诵的单词

子类接口:Get_Query()生成并返回题面。Check()检查输入是否合法。Work（）显示用户答题后的提示信息。

<3>Test_Strategy 类及其子类

基类接口：Init()对类进行初始化，将已经测试的题数清零，并将被测者的能力设为初始状态。End()给出本次词汇量测试是否结束。Get_Word()根据类中存储的被测者的能力值选择一个合适的考查单词并返回英文。Work(bool)根据被测者的回答正确与否对被测者的能力进一步评估，调整接下来的单词难度，并返回用户答题的结果信息。Set_Answer(int)可以设置当前这道题的标准答案。Get_Answer()可以返回标准答案。Get_Level()用来返回用户当前体现出来的水平。

子类接口:Test_Word(string)根据用户的输入给出本题回答是否正确。Get_Query(string)根据给定的英文单词，生成相应的题面并返回。Check(string)判断给定的用户输入是否为合法输入。

<4>NewWords_Strategy 类及其子类

基类接口：Run(string)根据用户输入的文件名调用虚函数GetText()从文件中获取文章，把单词分离出来，根据用户设置的英语水平和单词背诵情况判断是否为生词，把所有要输出的内容返回给Console类

子类接口：Get_Text(string)根据文件名从相应文件中读取内容并返回。

有关算法：为避免生词重复出现，基于set标准库来判断是否出现过。

（3）交互模块

<1>模块描述、功能

见整体架构

<2>Console类

接口： set_up()执行用户的登录以及 User 和一些策略类的初始化,Run()实现了词典运行时的UI并且执行了各项功能。

<3>Output类

子类接口:Print(string)输出string。

（4）跨平台设计

由于在Windows下的中文编码为GBK格式，在linux与Mac下中文编码为UTF8，因此为了解决中文输出的问题，将Console中需要输出的中文或各个策略类中需要传给Console进行输出的中文存入txt文件中，在预编译时，系统的不同会导致最终编译不同的代码，读入不同的txt文件。以Console为例，在windows下，会将Console_Windows.txt打开，将内容读入vector<string>get_out_console中，在linux和Mac下，则会打开Console_linux.txt，由此实现了中文输出的跨平台。

5.安全性设计

为了避免用户在输入时给出的输入不合法导致程序崩溃或者出现其它的错误，每次获取用户输入信息后进行合法性测试，如果不满足要求会给出提示信息让用户重新输入，直到得到合法输入再作为参数传给相应的类。