ECS36B Winter 2022 2022-03-04

Homework 5 due Fri 2022-03-11 at 23:59

Wordle game

In this assignment, you will implement the Wordle game. This game involves a single player who must guess a five-letter English word (the key) by entering successive guesses and getting a score for each guess specifying which letters of the guess are in the right position, which letters are present in the key but are not in the right position, and which letters are not in the key. The guesses must be recognized English words according to a reference list that is provided. We use the convention that a letter in the correct position gets the score **`'*'`**, a letter present in the key but in the wrong position gets the score **`'+'`**, and a letter not in the key gets the score **`'-'`**. For example, if the key is **`track`**, the guess **`shake`** gets the score **`--*+-`** . An example of input and output for the key **`track`** could be as follows:

Input:

```
slate
shake
crate
track
```

Output:

```
--*+-
--*+-
+**+-
*****
4 attempts
```

Note that in some cases, a letter may occur in a guess more times than it occurs in the key. The total number of **`'+'`** and **`'*'`** scores assigned to that letter cannot exceed the number of times it occurs in the key. For example, if the key is **`throw`**, and the guess is **`robot`**, the score should be **`+--*+`** .

A program **`game.cpp`** is provided. It uses an instance of the class **`WordleGame`** whose interface is defined in the file **`WordleGame.h`** (provided). You will have to implement the class **`WordleGame`** in a file **`WordleGame.cpp`** . A file **`words5`** containing a list of acceptable five-letter words (the dictionary) is provided.

The game is played by invoking the **`game`** program with a random positive integer (the index) given as a command line argument. The index is used to select a word from the **`words5`** file. This allows for testing with a reproducible choice of the key while not revealing the key to the player. The key is chosen to be the word at position **`i`** in the dictionary where **`i`** is the value of the index folded into the range **`[1..size_of_the_dictionary]`** using the modulo **`%`**

operator. For example, using the index **42510** selects the key **alley** which appears in position **i=78** in the list.

If a guess is not in the **words5** list, or if the guess is not a five-letter word, the program should print the message **not in dictionary** in place of the score, and continue to read the next guess. Entering an unrecognized word is counted as an attempt.

The **select_key** member function of the **WordleGame** class must find the word corresponding to the given index and store it in the **key** data member.
The **over()** member function of the **WordleGame** class must return **true** if the last guess entered was equal to the key and **false** otherwise.

Implementation

The STL **string** class, STL containers and algorithms should be used. The files **game.cpp**, **WordleGame.h** and **Makefile** are provided and must not be modified.

Testing

Example input files are provided together with corresponding reference output obtained e.g. with the following command:

```
$ ./game 42510 < test42510.in > test42510.out
```

Your implementation of the game should reproduce exactly the reference output files. Use the Unix **diff** command to compare your output file with the reference output file.
Note that the example output files only contain the scores. When playing the game in an interactive session, both input and output would appear interlaced on the terminal, as:

```
$ ./game 42510
audio
*----
abbey
*--**
alley
*****
3 attempts
```

Playing

To play the game interactively, invoke the **game** program with a random (positive) integer as a command line argument.

Submission

Create a tar file **hw5.tar** containing the file **WordleGame.cpp** only and submit it using Gradescope.

Acknowledgements

The file **words5** was obtained by selecting five-letter words from the list of English words provided by O. Astrachan at https://users.cs.duke.edu/~ola/ap/linuxwords under a Creative Commons License CC BY-SA 1.0.