LANGUAGES AND WORD LISTS

0

Generado por Doxygen 1.8.11

Índice general

Capítulo 1

Índice de clases

1.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

Capítulo 3

Documentación de las clases

3.1. Referencia de la Clase Language

Class fully implemented. It is used to store and manage all the details concerning a given language. It includes and make it publicily available some functions in wordlist.h like those to change the encoding of characters. **All the characters stored in memory use the ISO8859 standard** but characters read from keyboard might follow the UTF standard. These functions allow to change from one to another.

`#include <language.h>`

Métodos públicos

- Language ()
  Basic constructor and initializer.
- Language (std::string language)
  Basic constructor and initializer.
- bool query (std::string word) const
  Query if a given word exists in the given language.
- std::string getLanguage () const
  Returns the ISO690 identifier of the language.
- void setLanguage (std::string lang)
  Loads the chosen language, which must be under <root>/languages folder.
- int getFrequency (char letter) const
  Query the frecuency of appearance in Scrabble of the given letter, according to the chosen language.
- int getScore (char letter) const
  Query the score in Scrabble of the given letter, according to the chosen language.
- std::string getLetterSet () const
  Query the full set of available letters (without repetitions) in a given language.

3.1.1. Descripción detallada

Class fully implemented. It is used to store and manage all the details concerning a given language. It includes and make it publicily available some functions in wordlist.h like those to change the encoding of characters. **All the characters stored in memory use the ISO8859 standard** but characters read from keyboard might follow the UTF standard. These functions allow to change from one to another.

- std::string ISO8859toUTF8(const char * in);
- std::string UTF8toISO8859(const char * in);

Please note that all characters are stored in uppercase

Definición en la línea 27 del archivo language.h.

3.1.2. Documentación del constructor y destructor

3.1.2.1. Language::Language (std::string *language*)

Basic constructor and initializer.

Parámetros

language	Language chosen according to international rules ISO639-1 https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

Nota

If the language chosen does not exists in the folder <root>/languages it throws an exception and stops the program

3.1.3. Documentación de las funciones miembro

3.1.3.1. int Language::getFrequency (char *letter*) const

Query the frecuency of appearance in Scrabble of the given letter, according to the chosen language.

Parámetros

letter	The letter to query

Devuelve

The frequency of appearance of the letter in the Scrabble of the given language. It returns

Valores devueltos

0	if the letter does not appear in the language

Definición en la línea 69 del archivo language.cpp.

```
69                                          {
70     int pos = letterset.find(letter);
71     if ( pos != string::npos) {
72         return frequencies[pos];
73     }
74     else return 0;
75 }
```

3.1.3.2. string Language::getLanguage ( ) const

Returns the ISO690 identifier of the language.

Devuelve

A string with the ID of the language https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

Definición en la línea 65 del archivo language.cpp.

```
65                                      {
66     return language;
67 }
```

3.1.3.3. string Language::getLetterSet ( ) const

Query the full set of available letters (without repetitions) in a given language.

Devuelve

A string containing the letters available in ISO8859

Definición en la línea 84 del archivo language.cpp.

```
84                                       {
85     return letterset;
86 }
```

3.1.3.4. int Language::getScore (char *letter*) const

Query the score in Scrabble of the given letter, according to the chosen language.

Parámetros

letter	The letter to query

Devuelve

The number of points of the letter in the Scrabble of the given language. It returns

Valores devueltos

0	if the letter does not appear in the language

Definición en la línea 76 del archivo language.cpp.

```
76                                          {
77     int pos = letterset.find(letter);
78     if ( pos != string::npos) {
79         return scores[pos];
80     }
81     else return 0;
82 }
```

3.1.3.5. bool Language::query (std::string *word*) const

Query if a given word exists in the given language.

Parámetros

word	The word to be queried

Devuelve

Valores devueltos

true	if the word exists in the recorded language,
false	otherwise

Definición en la línea 61 del archivo language.cpp.

```
61                                      {
62     return wordlist.searchWord(normalizeWord(word));
63 }
```

3.1.3.6. void Language::setLanguage (std::string *lang*)

Loads the chosen language, which must be under <root>/languages folder.

Parámetros

lang	The ID of the language

Nota

If the language chosen does not exists in the folder <root>/languages it throws an exception and stops the program

Definición en la línea 26 del archivo language.cpp.

```
26                                      {
27     bool res, exit=false;
28     ifstream fi;
29     string letters;
30     int number, scs, frecs;
31
32     for (auto & c: l) c = toupper(c);
33     language = l;
34     letterset = "";
35     res = wordlist.load("./languages/"+language+".tree");
36     assert(res);
37     cout « "Opening ./languages/"+language+".scrabble" « endl;
38     fi.open("./languages/"+language+".scrabble");
39     assert(fi);
40     while (!exit)  {
41         fi » scs;
```

```
42          fi » number;
43          while (number >= 0){
44              frecs = number;
45              fi » letters;
46              letterset+=letters;
47              for (int i=0; i<letters.size(); i++) {
48                  frequencies.push_back(frecs);
49                  scores.push_back(scs);
50              }
51              fi » number;
52          }
53          if (number == -2)
54              exit = true;
55      }
56      cout « "OK "«letterset.size()« " Scrabble's letter read " « endl;
57  //    cerr « "["«toUTF(letterset)«"]"«endl;
58      fi.close();
59  }
```

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- include/language.h
- src/language.cpp

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo include/language.h

Fully functional static library to handle languages, which are represented as a full list of allowed words, stored as a tree to make search efficient O(n) being n the number of letters in the word to be loocked up.

```
#include <vector>
#include "wordlist.h"
```

Dependencia gráfica adjunta para language.h:

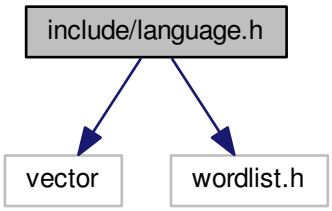
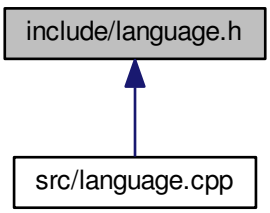


Gráfico de los archivos que directa o indirectamente incluyen a este archivo:



Clases

- class Language

  Class fully implemented. It is used to store and manage all the details concerning a given language. It includes and make it publicily available some functions in wordlist.h like those to change the encoding of characters. ***All the characters stored in memory use the ISO8859 standard*** *but characters read from keyboard might follow the UTF standard. These functions allow to change from one to another.*

4.1.1. Descripción detallada

Fully functional static library to handle languages, which are represented as a full list of allowed words, stored as a tree to make search efficient O(n) being n the number of letters in the word to be loocked up.

Autor

DECSAI

Nota

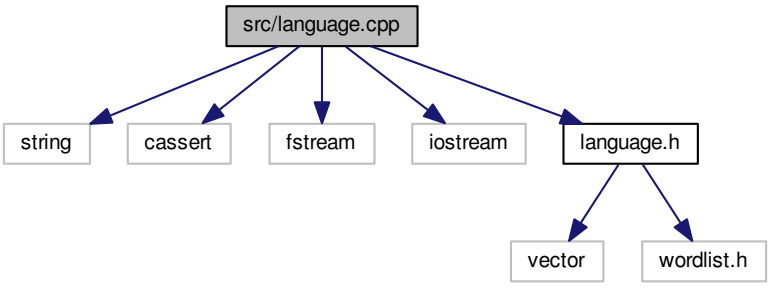
Fully implemented. No further implementation required.

4.2. Referencia del Archivo src/language.cpp

Fully functional static library to handle languages, which are represented as a full list of allowed words, stored as a tree to make search efficient O(n) being n the number of letters in the word to be loocked up.

```
#include <string>
#include <cassert>
#include <fstream>
#include <iostream>
#include "language.h"
```

Dependencia gráfica adjunta para language.cpp:



4.2.1. Descripción detallada

Fully functional static library to handle languages, which are represented as a full list of allowed words, stored as a tree to make search efficient O(n) being n the number of letters in the word to be loocked up.

Autor

DECSAI

Nota

Fully implemented. No further implementation required.

Índice alfabético