

Curso 2023 – 2024

PARAULOGIC

Objective

The student must demonstrate having acquired the programming knowledge developed during the course. Development of search and traversal algorithms, development of subprograms, use of arrays, solving problems using the STL library and treatment of sequential files.

Statement

Implement a program that allows playing PARAULOGIC, the player must form a series of words from a set of given letters. The set of letters is selected randomly from a text file. The game is described in greater detail later.

This delivery will be made up of:

1. Corresponding project along with the source code.
2. PDF documentation with the following sections:
 - a) Cover with the title of the practice, the name of the author, name of the subject.
 - b) Introduction that summarizes the statement of the practice.
 - c) Design. Where the top-down design that has led to the proposed solution is described.
 - d) Test set with the description of the tests, tests and data used to verify the proper functioning of the program.
 - e) Conclusions that synthesize the experience achieved, describe the concepts assimilated and highlight the points that have been most difficult to resolve.

Program details

The program to be developed consists of implementing the PARAULOGIC game. The program will load a text file called Palabras.txt, or will receive it as a parameter, in which case it must use the indicated file (that is, both cases must be implemented). The file will contain words in lowercase, one per line and without blank spaces. It is only played with words less than or equal to 7 letters, other words are ignored.

The user is asked to form a word with the given letters. And you will have as many attempts as you want.

- If the word is not found in the dictionary ☐ it must be indicated to the user.

- If the word is found in the dictionary:

or but does not contain the central letter ☐ you must indicate this to the user.

- If the word is found in the dictionary and contains the middle letter:

or but does not contain some of the given letters ☐ you must indicate this to the user.

or and if it contains the given letters, it must indicate this to the user and add 5 points to the score.

- If the user repeats a word, it should not add points. In this case, the user will be informed that it has already been entered.

- If the user finds all the words they can make with that set of letters, the user is prompted and the game ends.

In addition to proposing words, you can ask for help using commands:

- @help: It must explain the game and the possible help
- @exit: Exit the program.
- @hint: Given the set of letters with which to form a word, show all the possible combinations of the first two letters and the number of times they appear in the dictionary taking into account the maximum size of the word.

Example:

- They begin with: al --> appears: 7 times
 - They begin with: az --> appears: 3 times
 - They begin with: ca --> appear: 12 times
 - They begin with: cl --> appear: 1 times
 - They begin with: co --> appears: 3 times
 - Starts with: cu --> appears: 4 times
 - Starts with: ja --> appears: 1 times
 - Starts with: la --> appears: 1 times
 - Starts with: lo --> appears: 2 times
 - Starts by: lu --> appears: 1 times
 - Starts with: oc --> appears: 2 times
 - Starts with: uz --> appears: 1 times
 - Starts with: za --> appears: 3 times
 - Starts with: zo --> appears: 5 times
 - Starts with: zu --> appears: 4 times
-
- **@found:** Shows all the words found/matched with that set of letters.
 - **@show:** Shows the number of possible words that exist with that set of letters, taking into account that it will always contain the central letter. In addition to also showing the number of

missing combinations.

- **@rendicion:** Shows all the possible words that can be formed with that set of letters and contain the central letter.
- **@puntuacion:** Shows the score obtained to the user. Keep in mind that if a word is repeated, it should not add points.

Additional options

Any additional option will be valued extra. Complying only with the minimum requirements implies that the maximum achievable grade is 7. To obtain a better grade, it is necessary to make additional options.

Some possibilities are:

- Allow the user to play one game after another.
- Treat the letters ñ/ç and capital letters correctly.
- Allow two players (or more players) to play, each player indicating a word alternately. At the end of the game, it shows the winner (most correct words).
- Allow the user to choose the maximum size of letters with which they will be able to form words, and therefore indicate the size with which they will play.

Add additional help, such as:

- **@ranking:** shows the score of the current game and previous games, ordered from best to worst score.
- **@shufle:** rearrange letters keeping the position of the center letter.
- **@restart:** end current game and load new game with a new set of letters.
- **@recover:** shows the set of letters to play.
- **@language:** the user can choose the language they are going to play with. Keep in mind that the dictionary must be consistent with the chosen language.
- **@difficulty:** allow the user different levels of difficulty. Like, for example, including a level where incorrect words subtract points.
- **@attempts:** limit the number of attempts, always taking into account the total number of words that can be formed with that given set of letters.
- Any other help command proposed by the student.

Important Reminder

- The PARAULOGIC game must have size 7, and must be configurable with a constant called WORD_SIZE. `const int WORD_SIZE = 7;`
- The game must be able to load any file on any computer, regardless of the name.
- For the implementation the student must be able to use strings, vectors and maps.