

## Computer programming

### Bachelor's Degree in Electrical and Computer Engineering

**2022-2023**

#### Guess Game

The aim of this work is to implement the game “Guess Game”. A secret word will be displayed to the user. The objective of the game is to discover the secret word by entering letters.

The game ends successfully if the word is discovered without exceeding the possible number of wrong guesses.

#### Requirements Table:

Reference	Description
G001	The program has a header with the application's name and the complete names of the elements that make up the group. (See example I1)
G002	The program has a selection menu that allows the user to select one of the following options by entering a number: (See example I1) <ul style="list-style-type: none"> <li>• 1 to start playing.</li> <li>• 2 to define the number of possible wrong guesses.</li> <li>• 3 to define difficulty (word length).</li> <li>• 4 to print the help.</li> <li>• 5 to exit the program.</li> </ul>
G003	Whenever option '4' is introduced, the program should print the menu with the available options. You should not reprint the header.
G004	The program can only end with the entry of option '5' in the menu. All errors or failures must be handled internally, without the program ending its execution.
G005	When the user enters an invalid menu option, feedback should be given via an error message. (see example I1)
R001	The number of attempts is limited between 3 and 15. Any value outside this range must generate an error message and return to the main menu. The default value should be 5, ie in case no value is defined by the user.



R002	<p>The difficulty level is related to the length of the secret word. It is limited to 3 levels and must be entered as follows:</p> <ul style="list-style-type: none"><li>E - Define easy level (6 letter words)</li><li>M - Define medium level (8 letter words)</li><li>H - Define difficult level (10 letter words)</li></ul> <p>Any other value entered must be rejected with an error message. The default value is easy level.</p>
F001	<p>There must be a word database with at least 10 entries. The file available in the moodle can be used as an example.</p>
F002	<p>When the game option is selected, the secret word must be chosen from the repository at random.</p>
F003	<p>When the game option is selected, the screen should be cleaned and underscores ('_') must be wrote for each letter of the secret word. The number of remaining attempts must also be shown and asked a test letter to continue the game (See example I2).</p>
F004	<p>When a letter is entered, it must be checked if it exists in the secret word.</p> <ul style="list-style-type: none"><li>• If it exists, the corresponding underlined stroke must be replaced by the letter, continuing the game. (See example I3).</li><li>• If it does not exist, one attempt must be deducted from the number of possible attempts.</li></ul>
F005	<p>The game ends when all the secret word is discovered or when the number of attempts runs out. For each case, the program should indicate the result in a message and wait for the user to press a key to return to the main menu.</p>
F006	<p>When the game ends successfully, the result must be saved in a text file ("GuessGame.txt") in a new line, without erasing existing data. This file must be located in the same folder as the executable file.</p> <p>Example: (...) <i>Word: transpirar, length: 10, remaining attempts: 2</i></p>

## Example of the Interface

Referência	Exemplo
I1	<pre>***** *****  HANGMAN GAME  ***** ***** * Joao Santos - P2 ** Luis Oliveira - P2 *  *****  MENU  ***** 1 - Play 2 - Set number of retries 3 - Set difficulty level 4 - Help 5 - Exit</pre>
I2	<pre>***** ***** - - - - - ***** Remaining plays available: 3 Enter letter:</pre>
I3	<pre>***** ***** _ A _ _ A _ ***** Remaining plays available: 3 Enter letter:</pre>

## Material to Submit

- A compressed file containing: a) \*.c file with properly structured, indented and commented source code; b) \*.exe file.
- The identification of the files must contain the first and last name of each element of the group and the laboratory class to which it belongs. For example: JoaoSantosP2LuisOliveiraP2.c.

## Observations



- Deadline: 23:55 (GMT) on 11 June 2023.
- The works must be delivered on the Moodle platform, in the respective upload link. Other forms of delivery will not be considered.
- Groups that do not meet the delivery deadlines are penalized by 10% for each day of delay.
- All students will be called for individual defense of the work, on a date considered appropriate.
- If copied programs or parts of programs are found, all students involved will have a grade equal to 0.

April 2023