

Name: Senal Dampiya Dolage

Student Number: 10609526

Pseudocode for the word game.

Print welcome to the game message.

Define ***selectDifficulty*** function

Used after printing the welcome message.

Try to:

Input difficulty level (Easy, Medium or Hard) store as global variable, *difficulty*.

Except:

Show error message if invalid value entered.

Define ***selectWordList*** function

To be used after ***selectDifficulty()*** is completed.

This function is to be used only once for each round. It will be re-run if game is over and user plays again.

Choose random amount of words from the *candidateWords* list provided, the amount depending on *difficulty* level and is stored in global variable *wordList*.

If difficulty is Easy

wordList will have 6 words.

If difficulty is Medium

wordList will have 7 words.

If difficulty is Hard

wordList will have 8 words.

Use ***printWordList()*** to display the chosen words.

Returns *wordList* (type: []).

Define ***printWordList*** function

Reusable function to show the *wordList* after every user guess is entered.

Is to be used after ***selectWordList()***.

Loop for word in *wordList*:

Print index and word.

Define ***choosePassword*** function:

Select a random choice from *wordList* and store in global variable *password*.

Is to be used after ***selectWordList()*** is completed.

Returns *password* (type: string).

Define **getUserAnswer** function:

To be called after **choosePassword()**.

Define and set *won* variable default value to False.

Define and set variable *guessesRemaining* value depending on *difficulty* to limit users attempts.

If difficulty is Easy

guessesRemaining is 5.

If difficulty is Medium

guessesRemaining is 4.

If difficulty is Hard

guessesRemaining is 3.

Try to:

Loop for getting user guess (limit is *guessesRemaining*):

Input for getting users guess.

Print word chosen by user.

Use **compareWords()** to get matched characters.

If matched characters is equal to length of *password*

Print guess correct message.

won set to True.

exit loop.

Else

Print guess incorrect message.

won set to False.

Print matched characters.

Append matched characters in front of the guessed word.

printWordList() to print the *wordList* again.

Continue to next loop iteration.

Except if guess input is not valid:

Print invalid input error message.

If loop iterations over and all characters didn't match:

game lost

set won status to False

Finally:

show won/loose message according to variable *won*.

use **reRunGame()** to run game again.

Define ***compareWords*** function:

Parameters: password (required, string), user guess (required, string)

Loop password string:

Compare index value of password and user guess word.

If value is same, increment a *matched* variable.

Return matched variable (type: integer).

Define ***reRunGame*** function:

Input asking user to play again or not.

If yes:

Reuse defined functions in proper order again.

selectDifficulty()

selectWordList()

choosePassword()

getUserAnswer()

If no:

Print thank you for playing message.