

Artificial Intelligence

guess the word

SID:1940180 | 2022

# Difficulty Easy

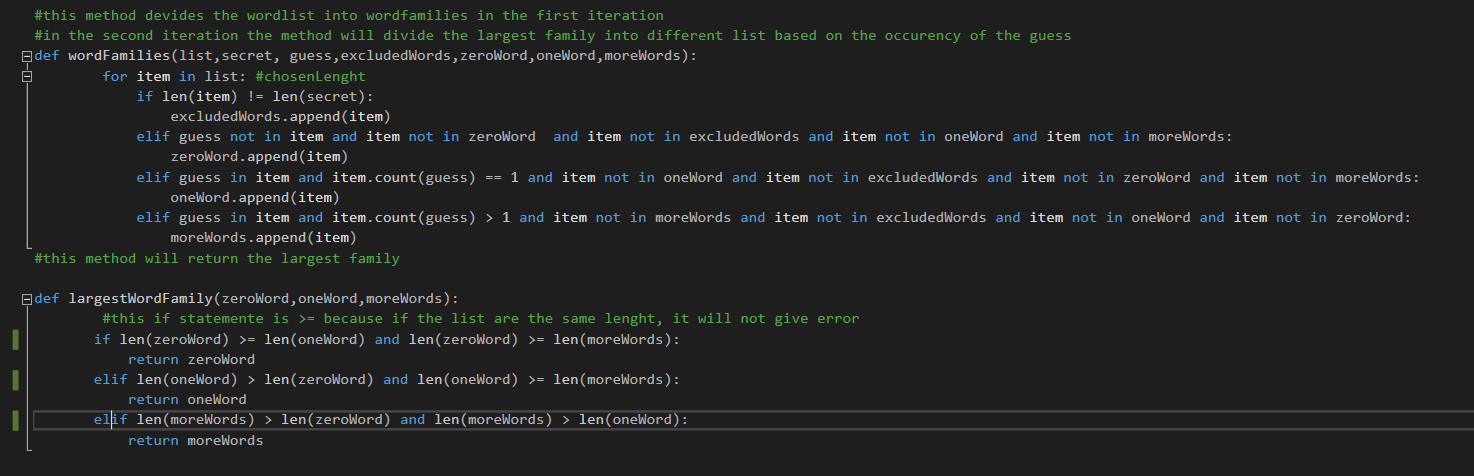
In the guess word game, the user should find the secret word. To make the cheating version of the game, the easy mode will change the word every time the user makes a guess. The file will be divided into different lists based on the frequency of the guess in the word. Then, the algorithm will choose the list with more elements and will randomly choose one word from this list, being aware of the positions of the discovered letter.

Figure 1 Largest Word Family List

Figure 1 shows the *wordFamilies* and the *largestWordFamily* method. The first method will check if the length of the word is the same as the length of the secret word, and then it discards the words with different sizes. Consequently, the algorithm will append the item to other lists based on the word’s guess frequency. The second method will return the list with more items. After the first iteration, the algorithm will work on the largest list in the previous iteration.

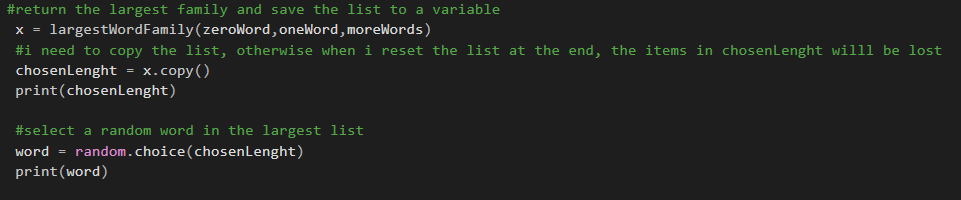


Figure 2 Word Chosen

Figure 2 shows the word chosen. Once the largest family is defined, the program will select a random word from that list. For debugging purposes, the largest list and the word are displayed in each iteration.

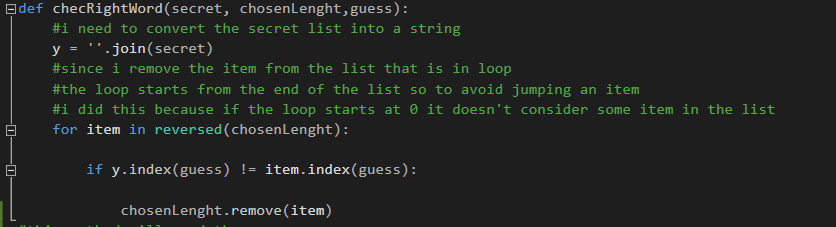


Figure 3 Index comparison

Figure 3 shows the index comparison. This method will compare the guess index of the secret word that is revealed to the user and compare it to the words in the largest list. If the letter is in a different position, the word will be removed from the list.

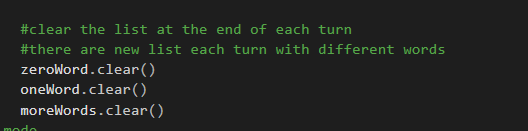


Figure 4 List clear

Figure 4 shows the List clear. These functions will remove all items in the list at each iteration. This is the final step to make the algorithm work.

In conclusion, the algorithm works as requested, dividing the words based on their frequency and finding the list with more items.

# Difficulty Hard

The hard mode of this game tries to make the algorithm more challenging to beat. The downside of the easy mode is that the algorithm can show a word with high-frequency letters, giving more information to the user and winning the game. In the Hard mode, the algorithm will consider two more variables: the number of guesses left and the empty spaces in the secret word.

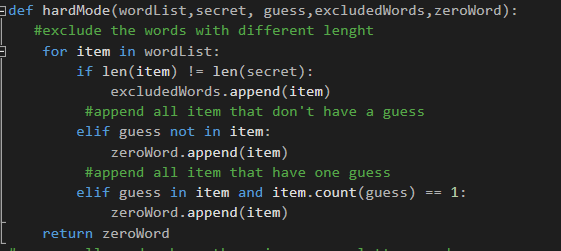


Figure 5 Hard mode list

Figure 5 shows the hard mode list. This method will create a list that contains words with zero or one guess frequency. Therefore, the user can only discover one letter in the worst-case scenario.

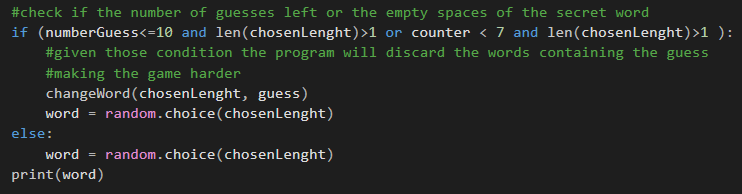


Figure 6 Word Chosen

Figure 6 shows the word chosen. The algorithm will consider the number of guesses left and the empty spaces of the secret word. Consequently, the computer will discard the word that has the guess, making the user lose an attempt.

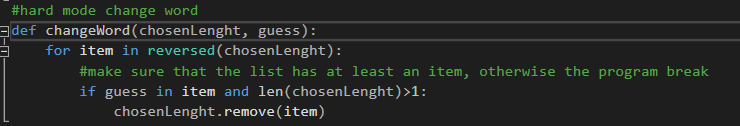


Figure 7 remove word

Figure 7 shows how the computer will remove a word if the conditions showed in figure 6 are met.

In conclusion, this algorithm is harder to beat because the word will change more times based not only on a random choice but it will consider how many guesses the user has left and the empty spaces of the secret word. Moreover, the list is bigger, giving the user more chances to fail more attempts.

# Appendix

Initially, the program will ask the user to choose the difficulty level of the game. 

Figure 8 Program interface

Figure 8 shows the program interface. The user has to choose the difficulty lever, and if the input is wrong, the program will ask for the correct input to start the game.

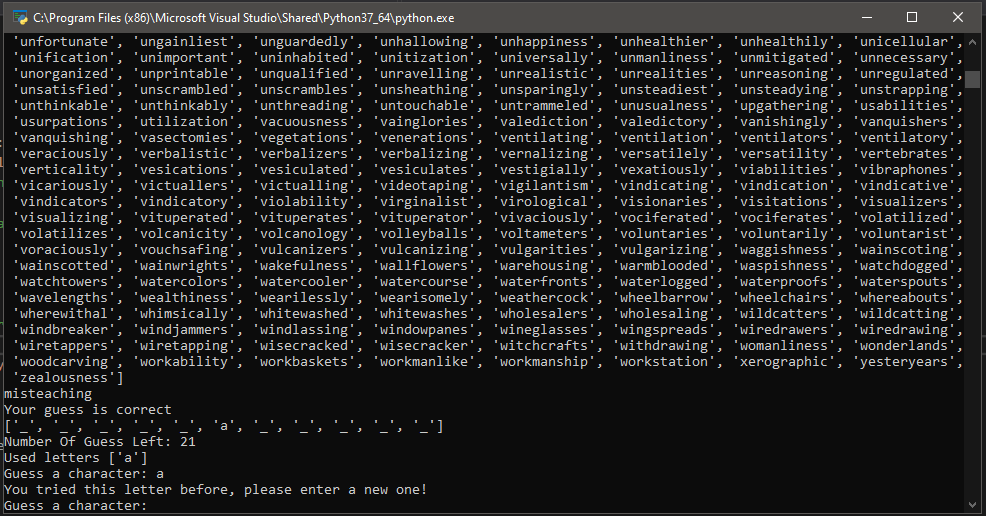


Figure 9 Guess check

Figure 9 shows the guess check. The user can’t enter the same letter twice, and number of guess is the same.

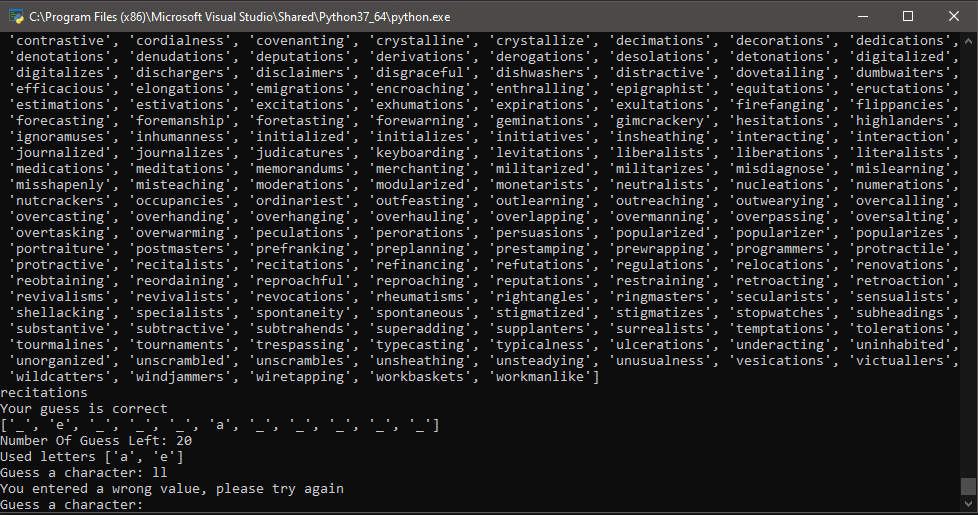


Figure 10 Guess check

Figure 10 shows the Guess check when the user inputs 2 characters.

In conclusion, the program runs until the user wins the game or the number of guesses left is zero. For debugging purposes, the program will prompt the random word at each turn, and the list with all possible words at each turn.