ULB Digital and IT essentials VUB

COURSE: FOUNDATIONS

TASK 1: THE GAME- GUESS A WORD

SUBMITTED BY:

DEEPA MARY VARGHESE

13/03/2025

INTRODUCTION:

Create a word guessing game using python programming language. I chose 9 random words for the game. The game is programmed in such a way that the system selects a random word for the user to guess. I have elaborated all the functions used in the programme.

METHODOLOGY:

Step 1: I had decided the words I want the user to guess.

Step 2. I started writing the programme in different sections.

Step 3: Identified the functions and their role in the programme.

Step 4: Drafted the programme step by step.

Step 5: Run the programme.

Step 6: Debugging the programme.

THE GAME:

1. Importing the random module:

- The **random** module is imported to use the **random.choice** function, which help in randomly selecting a word from a list of words.
- I choose this module because I had decided to have few random words that I wanted the player to guess. The random words that I chose are mentioned in the programme.

Python code:

```
import random
```

2. Defining the choose_word function:

- This function creates a list called **secret_word** containing several words.
- It then returns a randomly chosen word from this list using **random.choice.**
- This function contains all the random words that I want the user to guess. However, when the player starts the game, any one of the words will be randomly chosen by the system to guess.

Python code:

```
def choose_word():
    secret_word = ["Happy", "Hello", "Spring", "Autumn", "Winter", "Summer", "Season", "Rain", "Snow"]
    return random.choice(secret_word)
```

3. Defining the display_word function:

- This function takes two arguments**: secret_word** (the word to be guessed from the list of words) and **guessed_letters** (a list of letters that have been guessed so far).
- It constructs and returns a string where each letter in **secret_word** is shown if it has been guessed (i.e., it's in **guessed_letters**), otherwise, it shows an underscore _.
- This is the function I had made a mistake initially while writing the code. I wanted the user to know how many letters are there in the random word that has been choosen. I wanted the game to display (------) as the number of letters. I did not want a single line(______) to be displayed. Hence, I used **.join()** function to add a space between each letter. This will also help the user to guess the number of letter. I also wanted the user to only use lower case letter and hence **.lower()** function.

Python code:

```
def display_word(secret_word, guessed_letters):
    display = " ".join(letter if letter.lower() in guessed_letters else "_" for letter in
secret_word)
    return display
```

4. Defining the take_guess function:

- This function handles the main logic of the guessing game.
- It initializes **is_word_guessed** with a randomly chosen word from **choose_word,** converted to lowercase.
- It initializes **guessed_letters** as an empty list to keep track of guessed letters and **attempts** to 6, representing the number of allowed incorrect guesses.
- It prints the initial state of the game, showing underscores for each letter in the word to be guessed.
- This is the part of the code where I have written the print function for the game to start. Here, I decided the maximum number of limits to 6. In this part the user will also be shown how many letters are there in the word in the format (-----).

Python code:

```
def take_guess():
    is_word_guessed = choose_word().lower()
# Randomly choose a word and convert to lowercase
    guessed_letters = []
# Initialize an empty list to track guessed letters
    attempts = 6
# Number of allowed incorrect guesses

# Print the initial game state
    print("Welcome to the word guessing game")
    print("******************")
    print("Number of letters:", display_word(is_word_guessed, guessed_letters))
```

5. Main guessing loop:

- From this part of the programme, loop starts. I want the user to only give lower case string characters and no integers or numbers. It is the main loop part of the programme for the game.
- The loop continues as long as **attempts** is greater than 0.
- It prompts the user to input a guess (a single letter).
- It checks if the input is valid (a single alphabetic character); if not, it prints a corresponding message (You must enter a single letter) and continues to the next iteration.
- It checks if the letter has already been guessed; if so, it prints a corresponding message (you already guessed that letter) and continues to the next iteration.
- It adds the guessed letter to **guessed_letters**.
- If the guessed letter is not in **is_word_guessed**, it decrements **attempts** by 1, prints a message indicating the letter is not in the word, and shows the remaining attempts. The number of attempts decreases only if when the user guesses a wrong letter and not the repeated letter.
- If the guessed letter is in **is_word_guessed**, it counts the occurrences of the letter in the word and prints a message indicating how many times the letter occurs in the word.
- **len(guess) != 1 or not guess.isalpha() function ensures only a single alphabet is accepted as an input.**
- It updates and prints the current status of the guessed word using **display_word.**
- If there are no underscores left in the current status (i.e., the word is completely guessed), it prints a congratulatory message and breaks out of the loop thus, ending the loop.

Python code:

```
    while attempts > 0:
        guess = input("Guess a letter: ").lower()
 # Prompt user to input a guess and convert to lowercase


        if len(guess) != 1 or not guess.isalpha():
            print("You must enter a single letter.")
            continue
# Check if the input is valid

        if guess in guessed_letters:
            print("You already guessed that letter.")
            continue
# Check if the letter has already been guessed

        guessed_letters.append(guess)
# Add the guessed letter to the list

        if guess not in is_word_guessed:
            attempts -= 1
# Decrement attempts if the guess is incorrect
# Check if the guessed letter is in the word

            print("No letter '{guess}' occurs in the word.")
            print("You have {attempts} attempts remaining.")
        else:
```

```
      occurrences = is_word_guessed.count(guess)
# Count occurrences of the guessed letter in the word

      print("Letter '{guess}' occurs {occurrences} time(s).")

    current_status = display_word(is_word_guessed, guessed_letters)
# Update the current status of the guessed word
    print("Is word guessed:", current_status)

 # Check if the word has been completely guessed
    if "_" not in current_status:
      print("Congratulations! You guessed the word.")
      break
```

6. Game Over condition:

- If the loop exits because **attempts** reached 0, it prints a message indicating that the user ran out of attempts and reveals the word.

Python code:

```
  else:
    print("You ran out of attempts! The word was: {is_word_guessed}")
```

7. Running the game:

- The **take_guess** function is called to start the game.
- This function is important for the programme to run. I did not initially type this code and the programme did not run. I had to go through the entire programme and then understand which function I had to type to start the game.

Python code:

```
# Run the game
take_guess()
```

CHALLENGES:

1. Identifying the functions for the programme.
2. Indexation
3. Syntax errors due to indexation and missing special characters.
4. I had watched few you tube videos and referred to the course material to understand the functions better.

APPENDIX:

THE CODE:

```
import random

def choose_word():
    secret_word = ["Happy", "Hello", "Spring", "Autumn", "Winter", "Summer", "Season",
"Rain", "Snow"]
    return random.choice(secret_word)

def display_word(secret_word, guessed_letters):
    display = " ".join(letter if letter.lower() in guessed_letters else "_" for letter in secret_word)
    return display

def take_guess():
    is_word_guessed = choose_word().lower()
    guessed_letters = []
    attempts = 6

    print("Welcome to the word guessing game")
    print("******************")
    print("Number of letters:", display_word(is_word_guessed, guessed_letters))

    while attempts > 0:
        guess = input("Take guess: ").lower()

        if len(guess) != 1 or not guess.isalpha():
            print("You must enter a single letter.")
            continue

        if guess in guessed_letters:
            print("You already guessed that letter.")
            continue

        guessed_letters.append(guess)

        if guess not in is_word_guessed:
            attempts -= 1
            print("No letter '{guess}' occurs in the word.")
            print("You have {attempts} attempts remaining.")
        else:
            occurrences = is_word_guessed.count(guess)
            print("Letter '{guess}' occurs {occurrences} time(s).")

        current_status = display_word(is_word_guessed, guessed_letters)
        print("Is word guessed:", current_status)

        if "_" not in current_status:
            print("Congratulations! You guessed the word.")
            break
    else:
        print("You ran out of attempts! The word was: {is_word_guessed}")

# Run the game
take_guess()
```

EXAMPLE:

```
Welcome to the word guessing game

******************

Number of letters: _ _ _ _ _ _

Take guess: f

No letter 'f' occurs in the word.

You have 5 attempts remaining.

Is word guessed: _ _ _ _ _ _

Take guess: g

No letter 'g' occurs in the word.

You have 4 attempts remaining.

Is word guessed: _ _ _ _ _ _

Take guess: i

No letter 'i' occurs in the word.

You have 3 attempts remaining.

Is word guessed: _ _ _ _ _ _

Take guess: a

No letter 'a' occurs in the word.

You have 2 attempts remaining.

Is word guessed: _ _ _ _ _ _

Take guess: e

Letter 'e' occurs 1 time(s).

Is word guessed: _ _ _ _ e _

Take guess: w

No letter 'w' occurs in the word.

You have 1 attempts remaining.

Is word guessed: _ _ _ _ e _
```

```
Take guess: s

Letter 's' occurs 1 time(s).

Is word guessed: s _ _ _ e _

Take guess: u

Letter 'u' occurs 1 time(s).

Is word guessed: s u _ _ e _

Take guess: m

Letter 'm' occurs 2 time(s).

Is word guessed: s u m m e _

Take guess: r

Letter 'r' occurs 1 time(s).

Is word guessed: s u m m e r

Congratulations! You guessed the word.
```