

Lab project 3 : Wordle game

This is a python implementation of the Wordle Game.

Based on the online word game created and developed by Welsh software engineer Josh Wardle. Players have six attempts to guess a five-letter word, with feedback given for each guess in the form of coloured tiles indicating when letters match or occupy the correct position.

The current version of this project is for educational purposes.

ENSAT 2024/2025

The project has the following modules:

- `display.py` : Module for displaying colored text in the terminal for the Wordle game.
- `word_choice.py` : Module for getting all words from `words.txt` and choosing a random word.
- `validate_guess.py` : Module for validating input guess.
- `wordle.py`: Module for the wordle game

5-letter words of the English dictionary are in `wordle_package/assets/words.txt`. Each line of the document has a word.

Download the skeleton for the lab and implement the functions for the game. The repository for the lab can be found [here](#).

word choice

The module `word_choice` has two functionalities. The first is to get all the words from `words.txt` and the second is to pick a random word from the list of words.

1. In order to get all the words from the text file, you can use:

- `open()` to read a file
- `os.path` to lookup the location of the file

```
def read_words_file() -> List[str]:  
    """Read all valid 5-letter words.  
    Return: a list of words strings  
    """  
    pass
```

2. To pick a random word from a list of words, you can use:

- `random.choice()` to pick an element from a list

```
def choose_random_word(words: List[str]) -> str:  
    """Choose the starting word for the game.  
    Return: a random word on the list
```

```
"""  
pass
```

validate guess

The main goal for the `validate_guess` module is to validate the input from the user.

1. The first function checks whether an input is the correct format. The input needs to be a 5 letter word in lower case

```
def check_guess_valid(guess: str) -> bool:  
    """Check if the guess is a valid 5-letter word  
    Return: True if the guess is valid, False otherwise  
    """  
    pass
```

2. The second function is a while loop that asks the user to input a valid word. In the loop, you must check that the word is valid, not previously guessed and exists in `words.txt`. The function returns a valid guess of the user.

```
def get_valid_guess(all_words: List[str], guesses: List[str]) ->  
str:  
    """Get a valid guess from the user  
    Must be:  
    1. 5 letter word (lower case)  
    2. not previously guessed  
    3. exists in words.txt  
    """  
    # Ask user for a word  
    pass
```

Wordle

Moving now to the main part of the game. The `wordle` module has two functions:

1. The first function checks if the guess is the same as a word:

```
def check_guess_correct(word: str, guess: str) -> bool:  
    """Check the guess against the word  
    Return: True if the guess is correct, False otherwise  
    """  
    pass
```

2. The second is a function that returns the feedback for the guess.

- GREEN for a correct letter in the correct place
- YELLOW for a correct letter in the wrong place
- RED for a wrong letter

The function should return a list such that each position has one of these three colors with their correct indication.

```
def feed_back_word(word: str, guess: str) -> List[str]:  
    """Check the guess against the word and return the feedback  
    Return: a list with the feedback for each letter in the guess  
    """  
    # Initialize the feedback list  
    # It will contain the colors for each letter in the guess  
    # The default color is RED for all the letters  
    # GREEN for the correct letter in the correct position  
    # YELLOW for the correct letter in the wrong position  
    pass
```

Display

The `display` module contains all the prompts for the game.

1. Write functions for displaying the game instructions, the game start, the prompt for winning and losing

```
def game_instructions():  
    """Print the game instructions in the terminal."""  
    pass  
  
def game_start_display():  
    """Print the starting message for the game."""  
    pass  
  
def display_win(word: str, attempt: int) -> None:  
    """Display for winning"""  
    pass  
  
def display_lost(word: str) -> None:  
    """Display for losing"""  
    pass
```

2. Write a function that takes the guess and the feedback list and returns the guessed letter colored in their corresponding feedback position.

- You can `COLORS` dictionary for ANSI color formatting

```
def display_word_feedback(guess: str, feedback: List[str]) -> str:
    """Display the coloured feedback for a guess."""
    pass
```

The main loop for the game

Import all the modules to the main script and use all the aforementioned functionalities and implement the wordle game.

1. The starting parameters should be:

```
attempt = 0
previous_guesses = list()
```

2. Fetch all the word from `words.txt` and pick a random `word`
3. In each attempt, ask the user for a valid guess. Then check if they guessed correctly. If not show the feedback.
4. Prompt for winning or losing the game.