Mini-Project3

Hangman-game

June 21th

Yoji

Agenda

1. Overreview
2. Conding
    i. Word list
    ii. Validation
    iii. Game logic
3. Demonstration
4. Conclusion

Overreview

Objective

Create simple to-do-console-application.

Features

User can add, view, remove tasks.
And also, application suggest task depend on priority and deadline.

Coding - Word list

How did I get word list

I got word list which some made on GitHub
https://github.com/Xethron/Hangman/blob/master/words.txt

How did I read word list

When initialize hangman-class, read word list

```
class hangman:
    def __init__(self) -> None:
        with open("words.txt", "r") as file:
            self.words = file.read().splitlines()
```

Coding - Validation

After user type answer, it has validation.

- Only accept 1-letter alphabet.
- letter has not been guessed

```
if not letter.isalpha() or len(letter) != 1:
    print("Invalid input. Please enter a single letter.")
elif letter in lstguessed:
    letter = ""
    print("Already guessed!!")
```

Coding - Game logic

- Make sure letter in the correct word
  - Update guessed
  - If guessed doesn't have "_", game would end
- Update used_words word list

```
elif letter not in word:
    tries -= 1
    print(f"Sorry, '{letter}' is not in the word.")
    lstguessed.append(letter)
else:
    print(f"Good job! '{letter}' is in the word.")
    lstguessed.append(letter)
    for i in range(len(word)):
        if letter == word[i]:
            guessed[i] = letter

if "_" not in guessed:
    print(f"Congratulations! You guessed the word!: {word}\n")
    break
```

Demonstration

Conclusion

What was the challenges and impressions

Mini-Project3 was easier than Mini-Prject1. It didn't require new function, library and so on. Additionaly, it was solo project so that I didn't have to use GitHub with some branch strategy. I only use main branch.