Project Documentation

- I have built a hangman game using python
- I used python because it's very powerful and simple to use
- I defined all the attributes using the __init__ function

```
def __init__(self, word_list, word, num_lives=5):
        # TODO 2: Initialize the attributes as indicated in the
docstring
        # TODO 2: Print two message upon initialization:
        # 1. "The mistery word has {num_letters} characters"
        # 2. {word_guessed}
        self.word_list = word_list
        self.word = random.choice(word_list)
        self.num_letters = len(set(self.word))
        self.num_lives = num_lives
        self.word_guessed = []
        self.list_letters = []
        for i in range(len(self.word)):
            self.word_guessed.append("_")
        print(f"{self.word_guessed}")
        print(f"The mystery word has {self.num_letters} unique
letters.")
```

- I added to task 3 from milestone 1 by creating a program that asks the user to enter a different letter if they enter the same letter twice by using the else statement so when a letter that's already tried is entered it gets added to a list called list_letters I done this using the append function to add the letter to the list then I used the elif statement to check if the letter is in list_letters and then I used the print function to print the message that the letter has already been tried
- I used the while loop along with the len function to check if the letter entered is greater than 1 if it is it will print "please enter just one character"

```
def ask_letter(self):
        '''
```

```
        Asks the user for a letter and checks two things:
        1. If the letter has already been tried
        2. If the character is a single character
        If it passes both checks, it calls the check_letter method.
        '''
        # TODO 1: Ask the user for a letter iteratively until the
user enters a valid letter
        # TODO 1: Assign the letter to a variable called `letter`
        # TODO 1: The letter has to comply with the following
criteria: It has to be a single character. If it is not, print
"Please, enter just one character"
        # TODO 2. It has to be a letter that has not been tried yet.
Use the list_letters attribute to check this. If it has been tried,
print "{letter} was already tried".
        # TODO 3: If the letter is valid, call the check_letter
method
        while True:
            letter = str(input("Enter a letter: "))
            if len(letter) > 1:
                print("Please, enter just one character.")
            elif letter in self.list_letters:
                print(f"{letter} was already tried.")
            else:
                self.list_letters.append(letter)
                Hangman.check_letter(self, letter)
            if self.num_letters == 0:
                print("Congratulations! You won!")
                break
            if self.num_lives == 0:
                print(f"You lost! The word was {word}.")
                break
            else:
                pass
```

- I used the for loop and if statement and in to check if the letter was in the word.
- If the letter was in the word it would print a message saying your guess was correct
- I also used a minus to minus a guess from num_letters everytime you take a guess
- Otherwise I used the elif statement and not if to check if the letter was not in the word.
- If the letter was not in the word I used a minus to minus a life from num_lives
- If the letter was not in the word it would print a message saying your guess was incorrect
- Finally if num_letters = to 0 it would print a message saying you won the game and if the num_lives reaches 0 it would print a message saying you lost the game

```
def ask_letter(self):
        '''
        Asks the user for a letter and checks two things:
        1. If the letter has already been tried
        2. If the character is a single character
        If it passes both checks, it calls the check_letter method.
        '''
        # TODO 1: Ask the user for a letter iteratively until the user
enters a valid letter
        # TODO 1: Assign the letter to a variable called `letter`
        # TODO 1: The letter has to comply with the following criteria: It
has to be a single character. If it is not, print "Please, enter just one
character"
        # TODO 2. It has to be a letter that has not been tried yet. Use
the list_letters attribute to check this. If it has been tried, print
"{letter} was already tried".
        # TODO 3: If the letter is valid, call the check_letter method
        while True:
            letter = str(input("Enter a letter: "))
            if len(letter) > 1:
                print("Please, enter just one character.")
            elif letter in self.list_letters:
                print(f"{letter} was already tried.")
```

```
            else:
                self.list_letters.append(letter)
                Hangman.check_letter(self, letter)
            if self.num_letters == 0:
                print("Congratulations! You won!")
                break
            if self.num_lives == 0:
                print(f"You lost! The word was {word}.")
                break
            else:
                pass
```

- The last thing I done was create the program to play the game
- The first part of the code was already provided
- I used the if statement and I entered my word_list and used random.choice to select a word from the list and then the play_game function to play the game

```
def play_game(word_list):
    game = Hangman(word_list, word, num_lives=5)
    game.ask_letter()
    pass

if __name__ == '__main__':
    word_list = ['apple', 'banana', 'orange', 'pear', 'strawberry',
'watermelon']
    word = random.choice(word_list)
    play_game(word_list)
```