PYTHON MINI PROJECT REPORT

The Second Year Python Mini Project Report Submitted To

Don Bosco Institute of Technology

For The Award of:

DEGREE IN ELECTRONICS AND TELECOMMUNICATION:

Department of Electronics and Telecommunication Engineering

Don Bosco Institute of Technology

Kurla, Mumbai

Academic Year 2021-22

ACKNOWLGEDEMENT

We are thankful to the institute “Don Bosco Institute of Technology” for providing necessary facility to carry out learning through the Python Miniproject successfully.

It is our duty to record my sincere thanks and gratitude towards the institute.

Also, I am highly obliged to our faculty Ms. Poonam Chakraborty ma’am who provided such a great opportunity to complete this miniproject by guiding us throughout and clearing our doubts time to time.

Thank You Again,

Jaipreet Singh 11

Shruti Gokhale 40

Department of Electronics and Telecommunication Engineering
Don Bosco **Institute Of Technology, Mumbai Maharashtra, India**

Contents

Purpose

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a generalpurpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.The better way to learn Python is to learn it and then implement it.

Miniprojects are one way to have hands on experience on python and implementation.

Here in our mini project, we have created a Hangman game using Python code to get the hands on experience on it.

Steps on Hangman Game in Python

1. Import the random module.
2. Define a function to choose a word at random.
3. Allot 7 lives for the user.
4. Create a decision-making process, to check that the user only enters alphabets and not numbers as a name.
5. A while loop is created to check the condition that the letters of words are more than zero and lives are not zero.
6. Create visual representation of the ‘hangman’ as the name suggests i.e.

```
 ____
| /    |
|/    ( )
|      |
|     / \
|
```

7. For each wrong letter one item is added to the hangman representation.
8. Create a list for words to be chosen from named as words which is as

   words = [“ ”, “ ”, ...].

CODE

hangman.py

```
import random

from words import words

from hangman_visual import lives_visual_dict

import string



def get_valid_word(words):

    word = random.choice(words)  # randomly chooses something from the list

    while '-' in word or ' ' in word:

        word = random.choice(words)


    return word.upper()



def hangman():

    word = get_valid_word(words)

    word_letters = set(word)  # letters in the word

    alphabet = set(string.ascii_uppercase)

    used_letters = set()  # what the user has guessed


    lives = 7


    # getting user input

    while len(word_letters) > 0 and lives > 0:

        # letters used

        # ' '.join(['a', 'b', 'cd']) --> 'a b cd'

        print('You have', lives, 'lives left and you have used these letters: ', ' '.join(used_letters))


        # what current word is (ie W - R D)
```

```
        word_list = [letter if letter in used_letters else '-' for letter in word]
        print(lives_visual_dict[lives])
        print('Current word: ', ' '.join(word_list))

        user_letter = input('Guess a letter: ').upper()
        if user_letter in alphabet - used_letters:
            used_letters.add(user_letter)
            if user_letter in word_letters:
                word_letters.remove(user_letter)
                print('')

            else:
                lives = lives - 1  # takes away a life if wrong
                print('\nYour letter,', user_letter, 'is not in the word.')

        elif user_letter in used_letters:
            print('\nYou have already used that letter. Guess another letter.')

        else:
            print('\nThat is not a valid letter.')

    # gets here when len(word_letters) == 0 OR when lives == 0
    if lives == 0:
        print(lives_visual_dict[lives])
        print('You died, sorry. The word was', word)
    else:
        print('YAY! You guessed the word', word, '!!')


if _name_ == '_main_':
    hangman()
```

hangman_visual.py

```
lives_visual_dict = {
    0: """
        _____
       | /   |
       |/    ( )
       |     |
       |    / \\
       |
      """,
    1: """
        _____
       | /   |
       |/    ( )
       |     |
       |    /
       |
      """,
    2: """
        _____
       | /   |
       |/    ( )
       |     |
       |
       |
      """,
    3: """
        _____
       | /   |
       |/    ( )
       |
```

```
            |
            |
        """,
    4: """
            _____
            | /       |
            |/
            |
            |
            |
        """,
    5: """
            _____
            | /
            |/
            |
            |
            |
        """,
    6: """
            |
            |
            |
            |
            |
        """,
    7: "",
}
```

words.py

words = ["aback","abaft","abandoned","abashed","aberrant", ...
,"yawn","yell","zip","zoom"]

reference: https://www.randomlists.com/data/words.json

Screenshot of Code

```
Go  Run  Terminal  Help                                   hangman.py - Visual Studio Code
words.py   hangman_visual.py   hangman.py ×
D: > Pyproject > hangman.py > ...
import random
from words import words
from hangman_visual import lives_visual_dict
import string


def get_valid_word(words):
    word = random.choice(words)  # randomly chooses something from the list
    while '-' in word or ' ' in word:
        word = random.choice(words)

    return word.upper()


def hangman():
    word = get_valid_word(words)
    word_letters = set(word)  # letters in the word
    alphabet = set(string.ascii_uppercase)
    used_letters = set()  # what the user has guessed

    lives = 7

    # getting user input
    while len(word_letters) > 0 and lives > 0:
PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE
```

```
Go  Run  Terminal  Help                                   hangman.py - Visual Studio Code
words.py   hangman_visual.py   hangman.py ×
D: > Pyproject > hangman.py > hangman
    while len(word_letters) > 0 and lives > 0:
        # letters used
        # ' '.join(['a', 'b', 'cd']) --> 'a b cd'
        print('You have', lives, 'lives left and you have used these letters: ', ' '.join(used_letters))

        # what current word is (ie W - R D)
        word_list = [letter if letter in used_letters else '-' for letter in word]
        print(lives_visual_dict[lives])
        print('Current word: ', ' '.join(word_list))

        user_letter = input('Guess a letter: ').upper()
        if user_letter in alphabet - used_letters:
            used_letters.add(user_letter)
            if user_letter in word_letters:
                word_letters.remove(user_letter)
                print('')

            else:
                lives = lives - 1  # takes away a life if wrong
                print('\nYour letter,', user_letter, 'is not in the word.')

        elif user_letter in used_letters:
            print('\nYou have already used that letter. Guess another letter.')
PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE
```

Go Run Terminal Help hangman.py - Visual Studio Code

words.py | hangman_visual.py | hangman.py

D: > Pyproject > hangman.py > hangman

```python
            else:
                lives = lives - 1  # takes away a life if wrong
                print('\nYour letter,', user_letter, 'is not in the word.')

        elif user_letter in used_letters:
            print('\nYou have already used that letter. Guess another letter.')

        else:
            print('\nThat is not a valid letter.')

    # gets here when len(word_letters) == 0 OR when lives == 0
    if lives == 0:
        print(lives_visual_dict[lives])
        print('You died, sorry. The word was', word)
    else:
        print('YAY! You guessed the word', word, '!!')


if __name__ == '__main__':
    hangman()
```

Go Run Terminal Help

words.py | hangman_visual.py | hangman.py

D: > Pyproject > hangman_visual.py > ...

```python
lives_visual_dict = {
        0: """
                _________
                | /       |
                |/       ( )
                |         |
                |        / \\
                |
            """,
        1: """
                _________
                | /       |
                |/       ( )
                |         |
                |        /
                |
            """,
        2: """
                _________
                | /       |
                |/       ( )
                |         |
                |
                |
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Go Run Terminal Help

words.py | hangman_visual.py | hangman.py

D: > Pyproject > hangman_visual.py > ...

```python
                _________
                | /       |
                |/
                |
                |
                |
            """,
        5: """
                _________
                | /
                |/
                |
                |
                |
            """,
        6: """
                |
                |
                |
                |
                |
            """,
        7: "",
    }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Screenshot of Output

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

D:\Pyproject>python hangman
python: can't open file 'D:\\Pyproject\\hangman': [Errno 2] No such file or directory

D:\Pyproject>python hangman.py
You have 7 lives left and you have used these letters:

Current word:  - - - -
Guess a letter: a

Your letter, A is not in the word.
You have 6 lives left and you have used these letters:  A

              |
              |
              |
              |
              |

Current word:  - - - -
Guess a letter: s

Your letter, S is not in the word.
You have 5 lives left and you have used these letters:  S A

              _________
              |  /
              |/
              |
              |
              |

Current word:  - - - -
Guess a letter: d

You have 5 lives left and you have used these letters:  S A D

              _________
              |  /
              |/
              |
              |
              |

Current word:  D - - -
Guess a letter: f
```

C:\Windows\System32\cmd.exe

```
Your letter, F is not in the word.
You have 4 lives left and you have used these letters:  F S A D

              _________
             | /       |
             |/
             |
             |
             |

Current word:  D - - -
Guess a letter: e

You have 4 lives left and you have used these letters:  F S D E A

              _________
             | /       |
             |/
             |
             |
             |

Current word:  D E - -
Guess a letter: c

Your letter, C is not in the word.
You have 3 lives left and you have used these letters:  C F S D E A

              _________
             | /       |
             |/       ( )
             |
             |
             |

Current word:  D E - -
Guess a letter: x

Your letter, X is not in the word.
You have 2 lives left and you have used these letters:  C F X S D E A

              _________
             | /       |
             |/       ( )
             |         |
             |
             |
```

C:\Windows\System32\cmd.exe

```
Your letter, X is not in the word.
You have 2 lives left and you have used these letters:  C F X S D E A

              _________
             | /       |
             |/       ( )
             |         |
             |
             |

Current word:  D E - -
Guess a letter: z

Your letter, Z is not in the word.
You have 1 lives left and you have used these letters:  Z C F X S D E A

              _________
             | /       |
             |/       ( )
             |         |
             |        /
             |

Current word:  D E - -
Guess a letter: h

Your letter, H is not in the word.

              _________
             | /       |
             |/       ( )
             |         |
             |        / \
             |

You died, sorry. The word was DEBT

D:\Pyproject>_
```

CONCLUSION

So here we conclude that hangman is a popular word guessing game where the player attempts to build a missing word by guessing one letter at a time. After a certain number of incorrect guesses, the game ends and the player loses. The game also ends if the player correctly identifies all the letters of the missing word.