

Project Report

Project : HANGMAN GAME

Name : Rishav Dhakad(submitted by)

E-mail id : ce19b079@smail.iitm.ac.in

Contact no. : 8824675946

Trainer : Ms. mallika Srivastava

Submitted to : 1. Ms. mallika Srivastava

2. Mr. Mayur Dev Sewak

Introduction : The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it. Hangman is a popular word game in which one player (the "chooser") chooses a secret word and another player (the "guesser") attempts to guess the word one letter at a time. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. If the guesser figures out the secret word before he or she runs out of chances, he or she wins. If not, the player who chose the word wins.

Project Setup :

- 1.The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it.
- 2.The Game: Here, a random word (a fruit name) is picked up from our collection and the player gets limited chances to win the game.
- 3.When a letter in that word is guessed correctly, that letter position in the word is made visible. In this way, all letters of the word are to be guessed before all the chances are over.
- 4.For convenience, we have given length of word + 2 chances. For example, word to be guessed is mango, then user gets $5 + 2 = 7$ chances, as mango is a five letter word.

Code in text form-

```
import random
```

```
hang = [""
```

```
H A N G M A N - Fruit Edition
```

```
+---+
```

```
| |
```

```
|
```

```
|
```

```
|
```

```
|
```

```
===== "", ""
```

```
H A N G M A N - Fruits Edition
```

```
+---+
```

```
| |
```

```
O |
```

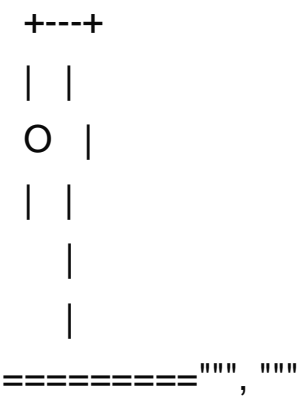
```
|
```

```
|
```

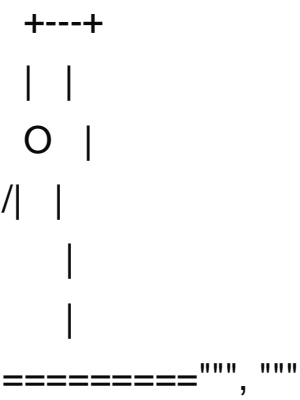
```
|
```

```
===== "", ""
```

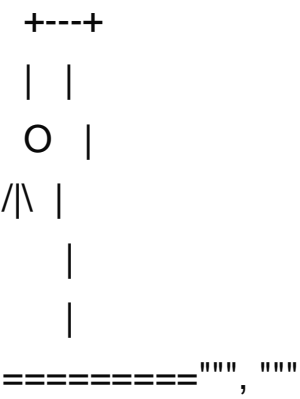
H A N G M A N - Fruits Edition



H A N G M A N - Fruits Edition



H A N G M A N - Fruits Edition



H A N G M A N - Fruits Edition

```

+---+
|   |
O   |
/\  |
/   |
    |
=====

```

H A N G M A N - Fruits Edition

```

+---+
|   |
O   |
/\  |
/\  |
    |
=====

```

```

def getRandomWord():
    words = ['apple', 'banana', 'mango', 'strawberry', 'cherry', 'orange',
'grape', 'pineapple', 'apricot',
            'lemon', 'coconut', 'watermelon', 'cherry', 'papaya', 'berry', 'peach',
'lychee', 'muskmelon', 'avocado', 'almond']

    word = random.choice(words)
    return word

```

```

def displayBoard(hang, missedLetters, correctLetters, secretWord):
    print(hang[len(missedLetters)])
    print()

    print('Missed Letters:', end=' ')
    for letter in missedLetters:
        print(letter, end=' ')
    print("\n")

    blanks = '_' * len(secretWord)

    for i in range(len(secretWord)): # replace blanks with correctly guessed
letters
        if secretWord[i] in correctLetters:
            blanks = blanks[:i] + secretWord[i] + blanks[i+1:]

    for letter in blanks: # show the secret word with spaces in between each
letter
        print(letter, end=' ')
    print("\n")

def getGuess(alreadyGuessed):
    while True:
        guess = input('Guess a letter: ')
        guess = guess.lower()
        if len(guess) != 1:
            print('Please enter a single letter.')
        elif guess in alreadyGuessed:

```

```
    print('You have already guessed that letter. Choose again.')
elif guess not in 'abcdefghijklmnopqrstuvwxyz':
    print('Please enter a LETTER.')
else:
    return guess
```

```
def playAgain():
    return input("\nDo you want to play again? ").lower().startswith('y')
```

```
missedLetters = ""
correctLetters = ""
secretWord = getRandomWord()
gameIsDone = False
```

```
while True:
    displayBoard(hang, missedLetters, correctLetters, secretWord)
```

```
    guess = getGuess(missedLetters + correctLetters)
```

```
    if guess in secretWord:
        correctLetters = correctLetters + guess
```

```
    foundAllLetters = True
    for i in range(len(secretWord)):
        if secretWord[i] not in correctLetters:
            foundAllLetters = False
            break
```

```

if foundAllLetters:
    print('\nYes! The secret word is "' +
          secretWord + '"! You have won!')
    gamelsDone = True
else:
    missedLetters = missedLetters + guess

    if len(missedLetters) == len(hang) - 1:
        displayBoard(hang, missedLetters,
                      correctLetters, secretWord)
        print('You have run out of guesses!\nAfter ' + str(len(missedLetters))
              + ' missed guesses and ' +
              str(len(correctLetters)) + ' correct guesses, the word was "' +
              secretWord + '"')
        gamelsDone = True

if gamelsDone:
    if playAgain():
        missedLetters = ""
        correctLetters = ""
        gamelsDone = False
        secretWord = getRandomWord()
    else:
        Break

```

Screenshot of code-




```
File Edit Selection View Go Run Terminal Help 130.py - Desktop - Visual Studio Code
qr_code.py 126.py 127.py 128.py 129.py 130.py 131.py qr_code.svg
code > .vscode > 130.py > ...
26 0 |
27 |
28 |
29 |
30 ===== """
31 H A N G M A N - Fruits Edition
32
33 +---+
34 | |
35 o |
36 /|
37 |
38 |
39 ===== """
40 H A N G M A N - Fruits Edition
41
42 +---+
43 | |
44 o |
45 /|\
46 |
47 |
48 ===== """
49 H A N G M A N - Fruits Edition
50
```

```
File Edit Selection View Go Run Terminal Help 130.py - Desktop - Visual Studio Code
qr_code.py 126.py 127.py 128.py 129.py 130.py 131.py qr_code.svg
code > .vscode > 130.py > ...
69 def getRandomWord():
70     words = ['apple', 'banana', 'mango', 'strawberry', 'cherry', 'orange', 'grape', 'pineapple', 'apricot',
71             'lemon', 'coconut', 'watermelon', 'cherry', 'papaya', 'berry', 'peach', 'lychee', 'muskmelon', 'avoca
72
73     word = random.choice(words)
74     return word
75
76
77 def displayBoard(hang, missedLetters, correctLetters, secretWord):
78     print(hang[len(missedLetters)])
79     print()
80
81     print('Missed Letters:', end=' ')
82     for letter in missedLetters:
83         print(letter, end=' ')
84     print("\n")
85
86     blanks = '_' * len(secretWord)
87
88     for i in range(len(secretWord)): # replace blanks with correctly guessed letters
89         if secretWord[i] in correctLetters:
90             blanks = blanks[:i] + secretWord[i] + blanks[i+1:]
91
92     for letter in blanks: # show the secret word with spaces in between each letter
93         print(letter, end=' ')
```

```
File Edit Selection View Go Run Terminal Help 130.py - Desktop - Visual Studio Code
qr_code.py 126.py 127.py 128.py 129.py 130.py 131.py qr_code.svg
code > .vscode > 130.py > ...
94 print("\n")
95
96
97 def getGuess(alreadyGuessed):
98     while True:
99         guess = input('Guess a letter: ')
100         guess = guess.lower()
101         if len(guess) != 1:
102             print('Please enter a single letter.')
103         elif guess in alreadyGuessed:
104             print('You have already guessed that letter. Choose again.')
105         elif guess not in 'abcdefghijklmnopqrstuvwxyz':
106             print('Please enter a LETTER.')
107         else:
108             return guess
109
110
111 def playAgain():
112     return input("\nDo you want to play again? ").lower().startswith('y')
113
114
115 missedLetters = ''
116 correctLetters = ''
117 secretWord = getRandomWord()
118 gameIsDone = False
119
```

```
code > .vscode > 130.py > ...
128 foundAllLetters = True
129 for i in range(len(secretWord)):
130     if secretWord[i] not in correctLetters:
131         foundAllLetters = False
132         break
133 if foundAllLetters:
134     print('\nYes! The secret word is "' +
135           secretWord + '"! You have won!')
136     gameIsDone = True
137 else:
138     missedLetters = missedLetters + guess
139
140     if len(missedLetters) == len(hang) - 1:
141         displayBoard(hang, missedLetters,
142                     correctLetters, secretWord)
143         print('You have run out of guesses!\nAfter ' + str(len(missedLetters)) + ' missed guesses and ' +
144               str(len(correctLetters)) + ' correct guesses, the word was "' + secretWord + '"')
145         gameIsDone = True
146
147 if gameIsDone:
148     if playAgain():
149         missedLetters = ''
150         correctLetters = ''
151         gameIsDone = False
152         secretWord = getRandomWord()
153
```

Screenshot of the successfulluy running project-

```
code > .vscode > 130.py > ...
PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL
Guess a letter: python -u "c:\Users\91882\Desktop\code\.vscode\130.py"
Please enter a single letter.
Guess a letter: m

H A N G M A N - Fruits Edition
+---+
|   |
0   |
|   |
|   |
+---+

Missed Letters: m

-----
Guess a letter: a

H A N G M A N - Fruits Edition
+---+
|   |
0   |
|   |
|   |
+---+

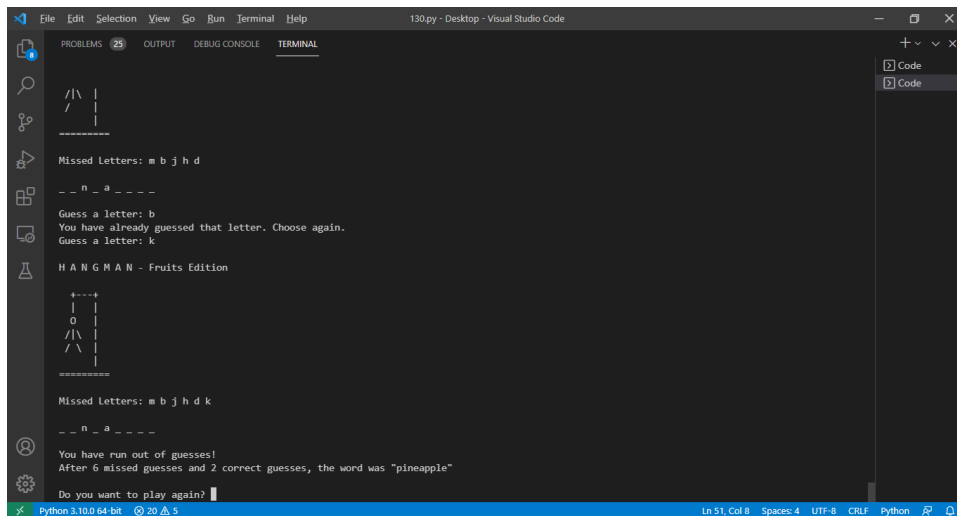
Missed Letters: m
```

```
File Edit Selection View Go Run Terminal Help 130.py - Desktop - Visual Studio Code
PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL
Missed Letters: m
_ _ _ _ a _ _ _ _
Guess a letter: n
H A N G M A N - Fruits Edition
+---+
|   |
0   |
|   |
|   |
+---+

Missed Letters: m
_ _ n _ a _ _ _ _
Guess a letter: b
H A N G M A N - Fruits Edition
+---+
|   |
0   |
|   |
|   |
+---+
```

```
File Edit Selection View Go Run Terminal Help 130.py - Desktop - Visual Studio Code
PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL
_ _ n _ a _ _ _ _
Guess a letter: a
You have already guessed that letter. Choose again.
Guess a letter: j
H A N G M A N - Fruits Edition
+---+
|   |
0   |
/ \ |
|   |
+---+

Missed Letters: m b j
_ _ n _ a _ _ _ _
Guess a letter: h
H A N G M A N - Fruits Edition
+---+
|   |
0   |
/ \ |
|   |
+---+
```



```
File Edit Selection View Go Run Terminal Help 130.py - Desktop - Visual Studio Code
PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL
/|\ |
/ | |
-----
Missed Letters: m b j h d
_ _ n _ a _ _ _ _
Guess a letter: b
You have already guessed that letter. Choose again.
Guess a letter: k
H A N G M A N - Fruits Edition
+----+
| | | |
| 0 | |
|/|\ | |
|/ \ | |
| | |
-----
Missed Letters: m b j h d k
_ _ n _ a _ _ _ _
You have run out of guesses!
After 6 missed guesses and 2 correct guesses, the word was "pineapple"
Do you want to play again?
Python 3.10.0 64-bit 20 5 Ln 51, Col 8 Spaces: 4 UTF-8 CRLF Python
```

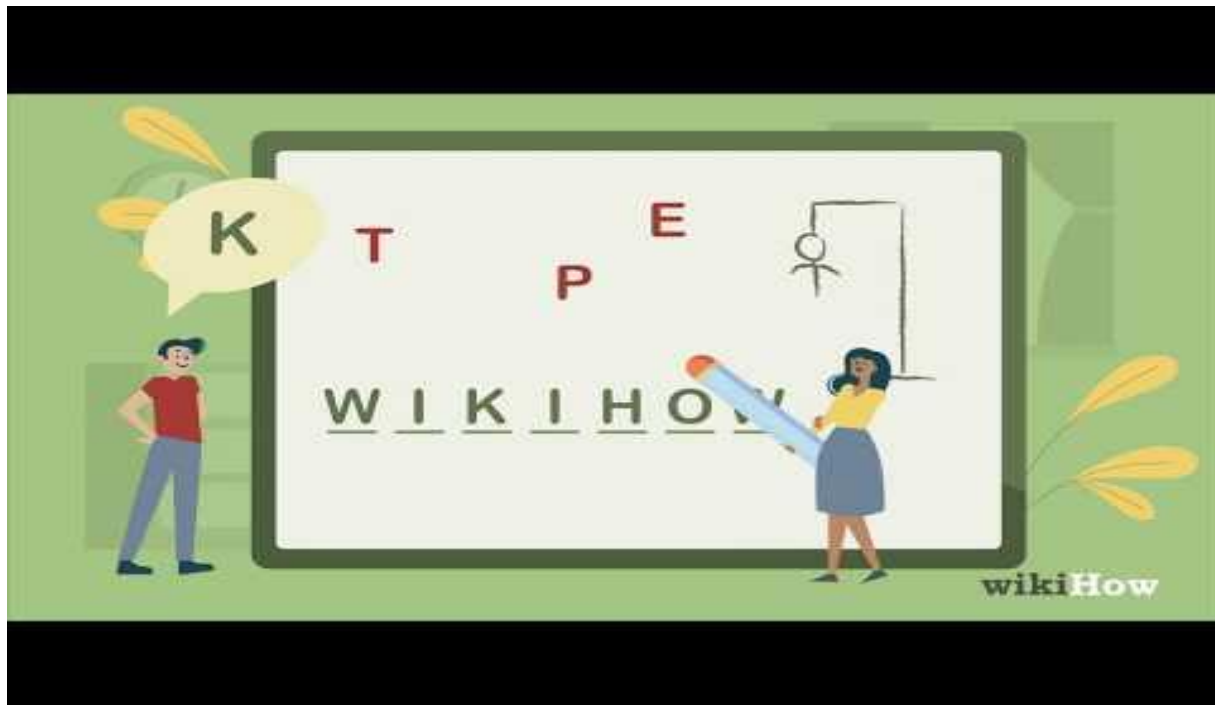
CONCLUSION-

In this game we might guess names of cities, or athletes, or fictional characters, or top forty song titles the list is endless. We will be writing a program to play a “guess a word letter-by-letter” version of hangman as shown above. We will also be doing some statistical analysis of the words used in the Hangman game.

REFERENCES-

1. <https://www.wikihow.com/Play-Hangman>

2. [How to Play Hangman](#)



THANK YOU

