**HANGMAN GAME**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

## Bachelor of Technology

In

## Computer Science and Engineering School of Engineering and Sciences

Submitted by

## BEECHA VENKATA NAGA HAREESH

## (AP21110010839)



Under the Guidance of

## MRS. Karnena Kavitha Rani

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur Andhra Pradesh – 522 240 [DECEMBER 2022]**

# Certificate

Date: 22/12/2022

This is to certify that the work present in this Project entitled “**HANGMAN GAME**” has been carried out by **BEECHA VENKATA NAGA HAREESH** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

## Supervisor

(Signature)

## MRS.Karnena Kavitha Rani

Designation,

Affiliation.

Table of Contents

Certificate

Table of Contents iv

Abstract 4

1. Introduction 5

2. Project Implementation: 6

2.1 Flowchart: 7

3. Software Requirements 8

4. Results 13-15

5. Conclusion 16

# Abstract

This is a simple Hangman game using Python language.  This can serve as a mini project to improve their logical reasoning and programming skills. A secret word is chosen at random by the Hangman program from a list of secret words. This functionality will be provided by the random module. In the well-known word game hangman, one player, known as the "chooser," selects a secret word, while another, known as the "guesser," tries to figure out the word one letter at a time. All occurrences of a guessed letter in the word are shown if it does. If not, there is no possibility for the guesser. The guesser wins if they correctly guess the secret word before their chances are up.

**1.INTRODUCTION**

Hangman is a word game in which the player is trying to guess a secret word. The player guesses letters, one at a time, and is told where each such letter appears in the secret word. If a guessed letter does not appear at all, it is considered a mistake. If the player makes six mistakes in total, the game is lost If the player successfully guesses all the letters in the secret word, the player wins.

Form this project we will write a program such a way that a random player can play the game of Hangman without the help of other guessing person

**2.Project Implementation:**

* We have imported the random function to randomly select the word for the list words.
* We have the displayBord function to display the hang man picture and display the guessed letter, missed letter, correct letter.
* We have the function getGuess () it checks the guessed letter is correct or wrong.
* playAgain() function is used to play again the game .
* In every function we have the different while, for, if and if else statement to make decisions

**2.1 Flowchart:**

Diagram

Description automatically generated

**3)Software Requirements**

**Operating system : Windows 7 or Greater**

**Processor :2 Ghz Intel Pentium 4 or equivalent**

**Memory : 1 GB RAM**

**Graphics : Integrated Graphics Chip**

**DirectX : Version 9.0**

**Storage : 50 MB available space**

**CODE:**

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

  +---+

  |   |

  O   |

  |   |

      |

      |

=========""", """

H A N G M A N - Fruits Edition

  +---+

  |   |

  O   |

 /|   |

      |

      |

=========""", """

H A N G M A N - Fruits Edition

  +---+

  |   |

  O   |

 /|\  |

      |

      |

=========""", """

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', 'orange', 'grape', 'pineapple', 'apricot',

             'lemon', 'coconut', 'watermelon', 'cherry', 'papaya', 'berry', 'peach', 'lychee', 'muskmelon']

    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!')

            gameIsDone = 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 + '"')

            gameIsDone = True

    if gameIsDone:

        if playAgain():

            missedLetters = ''

            correctLetters = ''

            gameIsDone = False

            secretWord = getRandomWord()

        else:

            break

**Result: -**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Conclusion**

With this, we come to the end of the article. This was an easy approach to make the game of hangman in python. It can definitely be enhanced by using the Tkinter library or allowing more words as per the coder’s desire. The purpose of this article was to provide a simplistic approach to develop the game of hangman in python.

However, if you have any doubts or questions, do let me know in the comment section below. I will try to help you as soon as possible.