Text-Based Hangman Game in Python

Mini Project Report

Amit Bikram Mishra
B.Tech, Computer Science and Engineering
Roll: 27900122002
Ideal Institute of Engineering

May 2025

Abstract

The purpose of this mini project is to design and implement a text-based Hangman game using the Python programming language. The game allows users to guess a randomly selected word by entering one letter at a time. A limited number of incorrect guesses are permitted, after which the game ends. This report outlines the algorithm, code structure, and testing of the game.

1 Introduction

Hangman is a classic word-guessing game played between two or more participants. The objective is to guess a hidden word by suggesting letters within a certain number of attempts. In this version, the computer selects a word at random, and the player guesses the word letter-by-letter.

Python, being an interpreted and beginner-friendly language, is well-suited for developing simple games like Hangman. This project emphasizes basic concepts such as loops, conditionals, input validation, and list manipulation.

2 Algorithm

The logic of the Hangman game is described through the following steps:

1. Initialize a list of potential words.
2. Randomly choose a word from the list.
3. Initialize a variable to hold the number of allowed incorrect guesses (e.g., 6).
4. Create a list to represent the guessed state of the word, initially filled with underscores.
5. Begin a loop that continues while attempts remain and the word is not fully guessed.
   - Display the current guessed state and letters guessed so far.
   - Take a single-letter input from the user.
   - Check the validity of the input.
   - If the guessed letter is in the word:
     - Reveal all occurrences in the display list.
   - If not, reduce the number of attempts.
6. End the game with a win or loss message depending on the result.

3 Code Implementation

The following Python code implements the above algorithm.

```python
import random

words = ['python', 'hangman', 'programming', 'challenge', 'development']
word_to_guess = random.choice(words)
guessed_letters = set()
attempts_remaining = 6
display_word = ['_' for _ in word_to_guess]

print("Welcome to Hangman!")

while attempts_remaining > 0 and '_' in display_word:
    print("\nWord:", ' '.join(display_word))
    print("Guessed letters:", ', '.join(sorted(guessed_letters)) or 'None')
    print("Attempts remaining:", attempts_remaining)

    guess = input("Enter a letter: ").lower()

    if not guess.isalpha() or len(guess) != 1:
        print("Invalid input. Enter a single letter.")
        continue
    if guess in guessed_letters:
        print("You've already guessed that letter.")
        continue

    guessed_letters.add(guess)

    if guess in word_to_guess:
        for i, letter in enumerate(word_to_guess):
            if letter == guess:
                display_word[i] = guess
        print("Correct!")
    else:
        attempts_remaining -= 1
        print("Wrong guess.")

if '_' not in display_word:
    print("Congratulations! You guessed the word:", word_to_guess)
else:
    print("Game Over. The word was:", word_to_guess)
```

Listing 1: Text-Based Hangman Game in Python

4 Testing and Sample Output

Test Case 1: The word is `python`.

Input: p, y, x, t, h, o, n

Output: Congratulations! You guessed the word: python

Test Case 2: The word is `challenge`.

Input: a, e, i, o, u, l, n, g, c, h

Output: Congratulations! You guessed the word: challenge

Test Case 3: Exceeded maximum wrong guesses.

Output: Game Over. The word was: programming

5 Conclusion

This project demonstrates how a simple game like Hangman can be implemented using Python's basic features. It helped reinforce concepts like loops, conditionals, input validation, list and string operations, and user interaction. The modularity and readability of Python allowed the logic to be implemented efficiently with minimal code.

References

- Python Official Documentation: `https://docs.python.org/3/`
- Real Python – Game Development with Python: `https://realpython.com`
- Stack Overflow Community for Debugging Support