Hangman Game Using Python

Name: Shivakumar Gandrakota

Overview

The Hangman game is a classic word-guessing game where players try to guess a hidden word by suggesting letters within a certain number of attempts. This document outlines the functionalities and features of a simple Hangman implementation in Python.

Functionalities

Core Features

1. **Random Word Selection**:
   - The game randomly selects a word from a predefined list of words.
2. **User Input**:
   - Players can input their guesses for letters.
3. **Display Progress**:
   - The current state of the word is displayed, showing correctly guessed letters and underscores for remaining letters.
4. **Hint Feature**:
   - At the start of the game, players receive a hint showing the first letter of the word.
5. **Incorrect Guess Tracking**:
   - The game tracks the number of incorrect guesses and informs the player of their remaining attempts.
6. **Win/Lose Conditions**:
   - The game ends when the player either successfully guesses the word or exhausts all attempts.
7. **Already Guessed Letters**:
   - Players are informed if they attempt to guess a letter that has already been guessed.

Enhanced Features (Possible Additions)

1. **Hangman Visuals**:
   - Display ASCII art representing the hangman based on the number of incorrect guesses.
2. **Word List Expansion**:
   - Include a larger set of words or allow the player to input their own words.
3. **Difficulty Levels**:
   - Implement different difficulty levels that affect the number of attempts or word length.

4. **Score Tracking**:
    - Keep track of scores or attempts over multiple games.
5. **User Interface Enhancements**:
    - Use libraries like `colorama` for colorful output or create a graphical interface using `tkinter`.
6. **Replay Option**:
    - Ask the player if they want to play again after the game ends.

Code

```
import random


class Hangman:
    def __init__(self):
        self.words = [
            "apple", "banana", "cherry", "dog", "elephant",
            "grape", "orange", "pineapple", "strawberry", "watermelon"
        ]
        self.word_to_guess = random.choice(self.words)
        self.guessed_letters = set()
        self.incorrect_guesses = 0
        self.max_attempts = 6

    def display_word(self):
        display = ""
        for letter in self.word_to_guess:
            if letter in self.guessed_letters:
                display += letter + " "
            else:
                display += "_ "
```

```
        return display.strip()

    def display_hint(self):
        return f"The first letter is: {self.word_to_guess[0].upper()}"

    def play(self):
        print(self.display_hint())

        while True:
            print(self.display_word())
            guess = input("Guess a letter: ").lower()

            if len(guess) != 1 or not guess.isalpha():
                print("Please enter a single letter.")
                continue

            if guess in self.guessed_letters:
                print("You already guessed that letter.")
                continue

            self.guessed_letters.add(guess)

            if guess not in self.word_to_guess:
                self.incorrect_guesses += 1
                print(f"Incorrect guess! Attempts remaining: {self.max_attempts -
self.incorrect_guesses}")
```

```
        if self.incorrect_guesses >= self.max_attempts:
            print(f"Game over! The word was {self.word_to_guess}.")
            break

        if set(self.word_to_guess) <= self.guessed_letters:
            print(f"Congratulations! You guessed the word: {self.word_to_guess}.")
            break

# Start the game
hangman_game = Hangman()
hangman_game.play()
```

Output

The first letter is: A

_ _ _ _ _

Guess a letter: a

A _ _ _ _

Guess a letter: b

Incorrect guess! Attempts remaining: 5

A _ _ _ _

Guess a letter: p

A p p _ _

Guess a letter: l

A p p l _

Guess a letter: e

Congratulations! You guessed the word: apple.