Hangman Game

CODE:

# Hangman Game

import random

from word\_list import words

# (0-6 wrong guesses)

# It shows the hangman progressively being drawn

hangman\_art =   {0:("   ",

                    "   ",

                    "   "),

                 1: (" o ",

                     "   ",

                     "   "),

                 2: (" o ",

                     " | ",

                     "   "),

                 3: (" o ",

                     "/| ",

                     "   "),

                 4: (" o ",

                     "/|\\",

                     "   "),

                 5: (" o ",

                     "/|\\",

                     "/  "),

                 6: (" o ",

                     "/|\\",

                     "/ \\")}

def display\_man(wrong\_guesses):

    # Shows the hangman as the user inputs wrong guesses

    print("-------------------------------------")

    for line in hangman\_art[wrong\_guesses]:

        print(line)

    print("-------------------------------------")

def display\_hint(hint):

    # Show the word with guessed letters and underscores

    print(" ".join(hint))

def display\_answer(answer):

    # Show the correct answer at the end of the game

    print(" ".join(answer))

def play\_game():

    # Pick a random word for the round

    answer = random.choice(words).lower()

    # Start with underscores instead of letters

    hint = ["\_"] \* len(answer)

    wrong\_guesses = 0

    guessed\_letters = set()

    is\_running = True

    while is\_running:

        # Show hangman + the current guessed word

        display\_man(wrong\_guesses)

        display\_hint(hint)

        # Get the player's guess

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

        # If input is invalid (not a single letter)

        if len(guess) != 1 or not guess.isalpha():

            print("\*\*Invalid\*\*")

            continue

        # If player already guessed this letter

        if guess in guessed\_letters:

            print(f"{guess} is already guessed")

            continue

        # Add the letter to the guessed set

        guessed\_letters.add(guess)

        # If guess is correct reveal it in the word

        if guess in answer:

            for i in range(len(answer)):

                if answer[i] == guess:

                    hint[i] = guess

        else:

            # Wrong guess leads to draw more hangman

            wrong\_guesses += 1

        # Check for win condition (no underscores left)

        if "\_" not in hint:

            display\_man(wrong\_guesses)

            display\_answer(answer)

            print("YEAH YOU WIN!")

            is\_running = False

        # Check for lose condition (if hangman is fully drawn - 6 wrong guesses the game is over)

        elif wrong\_guesses >= len(hangman\_art) - 1:

            display\_man(wrong\_guesses)

            display\_answer(answer)

            print("DARN YOU LOST!")

            is\_running = False

def main():

    # Main loop: keeps running until player says they don't want to play

    while True:

        play\_game()

        again = input("Do you want to play again? press (y): ").lower()

        if again != "y":

            print("Thanks for playing Hangman!")

            break

# This makes sure the main function runs first

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Explanation

Main Goal:

The Hangman game challenges the player to guess a secret word one letter at a time. With each wrong guess, a stick figure (the hangman) is drawn step by step. The game ends when the player either guesses the full word or the hangman is fully drawn.

1. **if \_\_name\_\_ == "\_\_main\_\_"**
   * Program entry point: starts the main menu loop.
2. **main()**
   * Calls play\_game() to start a round. After a round is completed, it asks the user whether to play again (via input) and then either calls play\_game() again or exits.
3. **play\_game()**
   * Calls random.choice(words) to pick the secret word.
   * Initializes hint, wrong\_guesses, and guessed\_letters.
4. **play\_game()``s game loop (repeats until win/lose):**  
   a. **display\_man(wrong\_guesses)** — prints the current hangman stage.  
   b. **display\_hint(hint)** — prints the current word hint (letters + underscores).  
   c. **input("Enter a letter: ")** — gets the player’s guess.  
   d. **validation logic** (not a separate function) — ensures a single alphabetic character and not previously guessed.  
   e. **guessed\_letters.add(guess)** — record the guess   
   f. If guess is correct: update the hint list (loop through answer and replace underscores).  
   g. If guess is wrong: wrong\_guesses += 1.  
   h. **Win check:** if "\_" not in hint → call display\_man(wrong\_guesses), **display\_answer(answer)**, print win message, set loop flag to stop.  
   i. **Lose check:** if wrong\_guesses >= 6 call display\_man(wrong\_guesses), **display\_answer(answer)**, print lose message, set loop flag to stop.
5. **Return to main()** after play\_game() ends
   * main() prompts: input("Do you want to play again? press (y): "). If 'y' → go to step 3; otherwise print goodbye and exit.