Spencer Davis

Hangman Project Documentation

My Thought Process

The goal was to create a Hangman game that greets the user, selects a random word, displays the word length, handles guesses, tracks progress, and let's the user play the game again if they want.

I have had lots of experience gathering user input, manipulating strings, and looping through data. I chose to code this in python because of the simple and organized layout and ease of gathering user input.

I broke down the game into key components:

- **Word Selection**: A list of 10 words I made up, and one gets randomly chosen.
- **User Interaction**: Functions for input and game state display.
- **Game Logic**: Checking guesses, updating the word, and determining game completion.

My Code Implementation

1. Greeting and Instructions: My code starts out with a simple greeting and instructions for the user.

2. Word Selection: Using random.choice, I selected a word from a list of strings I made.

3. Displaying Word Length: I displayed the word length to the user then showed the word using underscores to represent each letter.

4. Handling User Guesses: I insured the input was a single letter using len() and that the letter wasn't already guessed.

5. Updating the Game State: Replacing underscores with correct letters was tricky. I used a loop to check each character and update the display accordingly.

6. Providing Feedback: For each guess, I displayed the current state of the word, guessed letters, and counts of correct and incorrect guesses.

7. Checking for Game Completion: I checked if there were any underscores left to determine if all the letters have been guessed and the game is over.

8. Offering to Play Again: Prompted the user to play again or quit after each game.

Time Spent: I spent 1 hour and 54 minutes in total on this challenge.