

Hangman Game - Lab 1

Objective:

The objective of this program is to create a Hangman game where a random word is chosen from a list. The player must guess the letters one by one until they either guess the full word or run out of lives.

Code Explanation:

1. A list of words is defined, and one random word is chosen.
2. An empty display list is initialized with underscores ('_') to represent each letter of the chosen word.
3. The player inputs a guessed letter, which is checked against the chosen word.
4. If the guessed letter is correct, it replaces the corresponding underscore(s) in the display.
5. If the guessed letter is wrong, the player loses a life. A hangman stage is displayed for each incorrect guess.
6. The game continues until the player either guesses all letters (win) or runs out of lives (lose).

Code:

```
import random
import hangman_stages

List = ['furiously', 'intelligence', 'technology', 'information',
'strawberry']
lives = 8
chosen_word = random.choice(List)
# print(chosen_word)

display = []
# for displaying dashes "_"
for i in range(len(chosen_word)):
    display += "_"
print(display)

game_over = False
while not game_over:
    guessed_letter = (input('Guess a letter: ')).lower()

    for position in range(len(chosen_word)):
        letter = chosen_word[position]
        if letter == guessed_letter:
```

```
        display[position] = guessed_letter
    print(display)

    if guessed_letter not in choosen_word:
        lives -= 1
        print(f"You guessed {guessed_letter}, that's not in the word. You
lose a life.")
        if lives == 0:
            game_over = True
            print("You lose!!")
        if "_" not in display:
            game_over = True
            print("You win!!!")
    print(f"You have {lives} lives left.")
    print(hangman_stages.stages[lives])
```

Output/Usage:

The program displays a series of underscores representing the hidden word. The player keeps guessing letters until they win by completing the word or lose all their lives, with the hangman figure displayed after each wrong guess.