AI-Based Number Guessing Game Report

Title Page

Problem Statement: Develop an AI-Based Number Guessing Game that allows users to select a difficulty level, receive hints, and track their scores on a leaderboard.

Personal Details:

Name: [Sunny Kumar]

Roll No.: [202401100400195]

Introduction

The AI-Based Number Guessing Game is designed to challenge users to guess a randomly generated number within a limited number of attempts. The game provides hints at specific intervals to assist players in making educated guesses. Additionally, the program features a leaderboard to track high scores.

This game aims to make number guessing more engaging by incorporating an AI-like hint system that dynamically provides clues to guide players toward the correct answer.

Methodology

The approach used to develop the AI-Based Number Guessing Game includes the following steps:

1. **Difficulty Selection**: Players can select from three difficulty levels that determine the range of the secret number.
2. **Number Generation**: The game generates a random number within the chosen range using Python's random module.
3. **Guessing Mechanism**: The player is given a maximum of 20 attempts to guess the number correctly.
4. **Hint System**:
    - After 5 attempts: Parity hint (Even/Odd)
    - After 10 attempts: Divisibility hint
    - After 15 attempts: Range hint (Upper or Lower half)
5. **Scoring & Leaderboard**: Scores are recorded based on the number of attempts

taken, and a leaderboard ranks players based on their best scores.

Code

```
import random

# Dictionary to store player scores
leaderboard = {}

def get_divisor_hint(secret_number):
    """Returns a small divisor of the secret number, if any, else None."""
    for i in range(2, secret_number // 2 + 1):
        if secret_number % i == 0:
            return i
    return None

def choose_difficulty():
    """Allows the player to choose a difficulty level."""
```

```
    print("Choose a difficulty level:")
    print("1. Easy (1-50)")
    print("2. Medium (1-100)")
    print("3. Hard (1-200)")

    while True:
        choice = input("Enter 1, 2, or 3: ")
        if choice == '1':
            return 1, 50
        elif choice == '2':
            return 1, 100
        elif choice == '3':
            return 1, 200
        else:
            print("Invalid choice. Please select 1, 2, or 3.")

def play_game(player_name):
    """Main game logic where the player guesses the secret number."""
    low, high = choose_difficulty()
    secret_number = random.randint(low, high)
    max_attempts = 20
```

```
attempts = 0
hints_given = {"parity": False, "divisor": False, "range": False}

print(f"\nI have selected a number between {low} and {high}.")
print(f"You have {max_attempts} attempts to guess it correctly!")

while attempts < max_attempts:
    try:
        guess = int(input("Enter your guess: "))
    except ValueError:
        print("Please enter a valid number.")
        continue

    attempts += 1

    if guess < secret_number:
        print("Too low!")
    elif guess > secret_number:
        print("Too high!")
    else:
```

```
            print(f"Congratulations {player_name}! You guessed it in {attempts} attempts.")
            score = max_attempts - attempts + 1
            print(f"Your score for this round is: {score}")

            leaderboard[player_name] = max(leaderboard.get(player_name, 0), score)
            break

        # Provide hints at specific attempts
        if attempts == 5 and not hints_given["parity"]:
            print(f"Hint: The secret number is {'even' if secret_number % 2 == 0 else 'odd' }.")
            hints_given["parity"] = True
        elif attempts == 10 and not hints_given["divisor"]:
            divisor = get_divisor_hint(secret_number)
            print(f"Hint: The secret number is {'a prime number' if not divisor else f'divisible by {divisor}' }.")
            hints_given["divisor"] = True
        elif attempts == 15 and not hints_given["range"]:
            midpoint = (low + high) // 2
```

```
            print(f"Hint: The secret number is in the {'lower' if
secret_number <= midpoint else 'upper'} half of the range.")
            hints_given["range"] = True

        print(f"Attempts remaining: {max_attempts - attempts}")
    else:
        print(f"Sorry, you've used all {max_attempts} attempts. The secret
number was: {secret_number}")

def display_leaderboard():
    """Displays the leaderboard sorted by the highest score."""
    if not leaderboard:
        print("No scores yet. Play a game first!")
        return

    print("\n--- Leaderboard ---")
    sorted_leaderboard = sorted(leaderboard.items(), key=lambda x:
x[1], reverse=True)
    for rank, (player, score) in enumerate(sorted_leaderboard, start=1):
        print(f"{rank}. {player}: {score}")
    print("--------------------\n")
```

```
def main():
    """Runs the game loop."""
    print("Welcome to the AI-Based Number Guessing Game!")
    player_name = input("Enter your name: ")

    while True:
        play_game(player_name)
        display_leaderboard()

        play_again = input("Do you want to play again? (yes/no): 
").strip().lower()
        if play_again not in ("yes", "y"):
            print("Thank you for playing! Goodbye.")
            break

if __name__ == "__main__":
    main()
```

```
Welcome to the AI-Based Number Guessing Game!
Enter your name: Sunny Kumar
Choose a difficulty level:
1. Easy (1-50)
2. Medium (1-100)
3. Hard (1-200)
Enter 1, 2, or 3: 1

I have selected a number between 1 and 50.
You have 20 attempts to guess it correctly!
Enter your guess: 25
Too low!
Attempts remaining: 19
Enter your guess: 35
Too high!
Attempts remaining: 18
Enter your guess: 30
Too high!
Attempts remaining: 17
Enter your guess: 27
Congratulations Sunny Kumar! You guessed it in 4 attempts.
Your score for this round is: 17

--- Leaderboard ---
1. Sunny Kumar: 17
--------------------

Do you want to play again? (yes/no): no
Thank you for playing! Goodbye.
```

References/Credits

- Help from the Internet and Python resources.
- Python Random Library Documentation.