

Project Report on Number Guessing Game

Submitted by:

Name: Shiksha Agrawal

Branch: CSEAIML

University Roll No: 202401100400175

Methodology

The Number Guessing Game is a simple console-based Python application where the user tries to guess a randomly generated number within a limited number of attempts. The methodology for this project is outlined as follows:

1. Game Initialization:
 - The game starts with a welcome message and prompts the user to press 'S' to begin.
2. Difficulty Level Selection:
 - The user selects a difficulty level: Easy (E), Medium (M), or Hard (H).
 - Each level determines the range of numbers and the maximum number of attempts:
 - Easy: Range 0-50, Max Attempts: 3
 - Medium: Range 0-200, Max Attempts: 5
 - Hard: Range 0-500, Max Attempts: 8
3. Random Number Generation:
 - The program uses Python's 'random' module to generate a number within the selected range.
4. Guessing and Feedback:
 - The user guesses the number within the defined number of attempts.
 - After each guess, the program provides feedback:
 - Too low if the guess is less than the target number.
 - Too high if the guess is greater than the target number.

- Correct if the guess matches the target number.

5. End of Game:

- If the user guesses the number correctly, the game congratulates the user and shows the number of attempts.
 - If the user fails to guess within the allowed attempts, the game reveals the correct number.
-

Code

```
import random
```

```
# Starting the game
```

```
def guess():
```

```
    print("Welcome to the 'Guessing a number' Game")
```

```
    command = input('Press S to Start the game: ')
```

```
    s = command.upper()
```

```
    if s != 'S':
```

```
        print("Invalid input. Game exits.")
```

```
        return # Exit the function if input isn't S
```

```
# Take input of the difficulty level
```

```
choice = input("Choose Your Difficulty Level:
```

```
    1) Easy  (Press E)
```

```
    2) Medium (Press M)
```

```
    3) Hard  (Press H)
```

```
    ")
```

```
# Initializing attempts
```

```
guess_correctly = False
```

```

attempt = 0

c = choice.upper()
if c == 'E':
    print("The Number is now between 0-50.")
    number = random.randint(0, 50)
    max_attempts = 3 # Set maximum attempts for easy level
elif c == 'M':
    print("The Number is now between 0-200.")
    number = random.randint(0, 200)
    max_attempts = 5 # Set maximum attempts for medium level
elif c == 'H':
    print("The Number is now between 0-500.")
    number = random.randint(0, 500)
    max_attempts = 8 # Set maximum attempts for hard level
else:
    print("Unmatched difficulty level!")
    return # Exit if difficulty level not matched

while not guess_correctly and attempt < max_attempts:
    guess_number = int(input(f"Attempt {attempt + 1}/{max_attempts}: Guess the number: "))
    attempt += 1

    if guess_number < number:
        print("The number you guessed is too low.")
    elif guess_number > number:
        print("The number you guessed is too high.")
    elif guess_number == number:
        print(f"Congratulations! You guessed the correct number {number} in {attempt} attempts.")
        guess_correctly = True

```

```
if not guess_correctly:

    print(f"Oops! You've used all your {max_attempts} attempts. The correct number was
{number}.")

# Repeat the game

guess()
```

Result/Output

Example Output for Easy Level:

Welcome to the 'Guessing a number' Game

Press S to Start the game: S

Choose Your Difficulty Level:

1) Easy (Press E)

2) Medium (Press M)

3) Hard (Press H)

E

The Number is now between 0-50.

Attempt 1/3: Guess the number: 25

The number you guessed is too low.

Attempt 2/3: Guess the number: 40

The number you guessed is too high.

Attempt 3/3: Guess the number: 33

Congratulations! You guessed the correct number 33 in 3 attempts.

Conclusion

The Number Guessing Game is a simple yet engaging Python project that demonstrates the use of conditional statements, loops, and random number generation. Through this project, the importance of user interaction, feedback, and efficient code structure was highlighted. The game provides a fun and interactive experience while also strengthening Python programming concepts.

