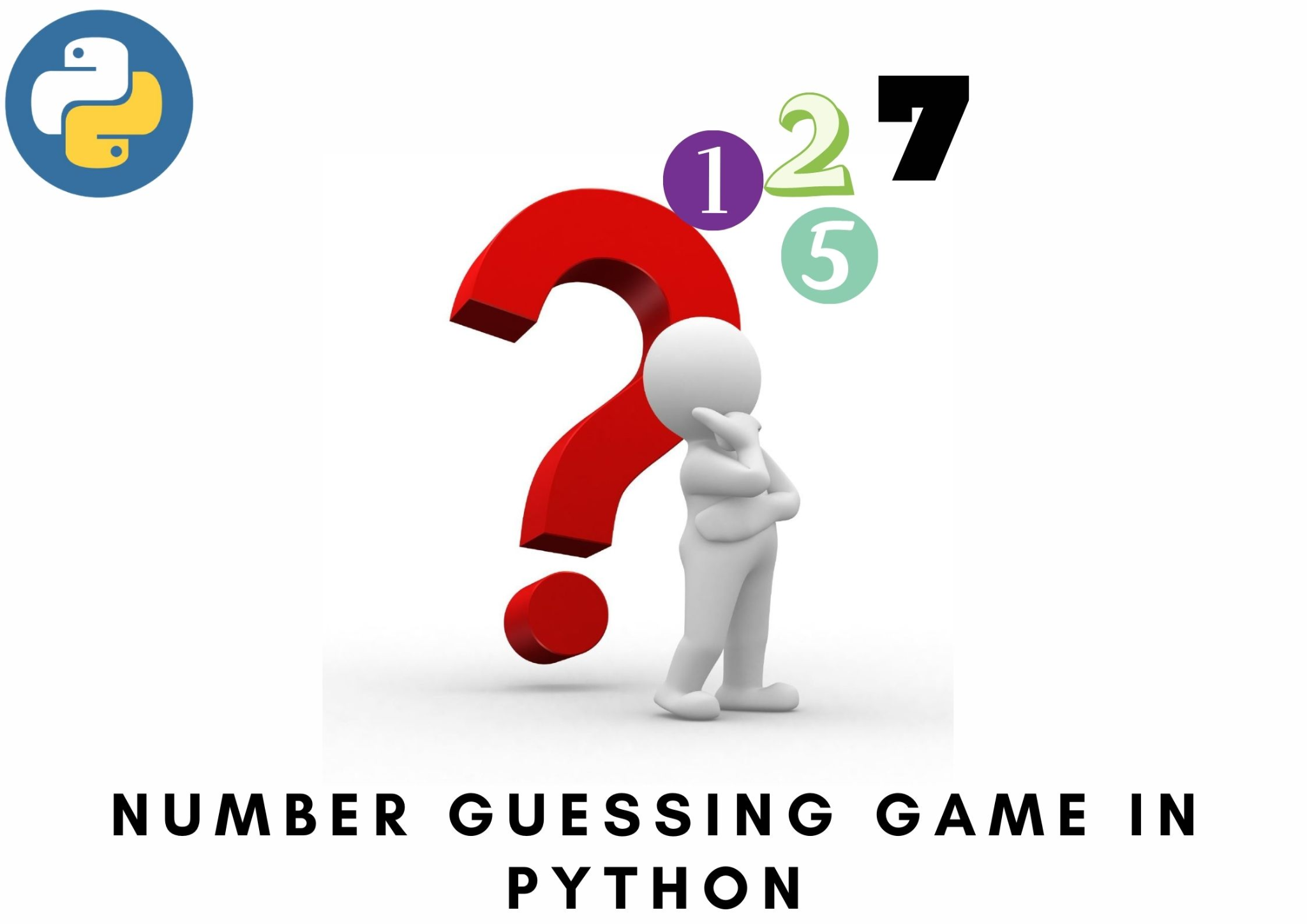
**Mini-Project-3: Number Guessing Game**

****



**Introduction**

Welcome to the Python Number Guessing Game! This interactive program presents a stimulating challenge, inviting you to engage your logical prowess. Within a range of 1 to 100, a number awaits your deduction. Delve into this game by selecting your preferred difficulty level – easy, medium, or hard – each offering a distinctive experience. With a limited number of attempts at your disposal, embark on a strategic journey to ascertain the concealed number. Prepare yourself for a cerebral exercise in this intriguing Python gaming endeavor.

**Problem Statement:**

Develop a Python-based Number Guessing Game that challenges players to deduce a randomly generated number within a given range.

In our number guessing game, there are several major functions that handle different aspects of the game:

1. **check\_answer(guess, answer, turns):** This function compares the user's guess with the actual answer. It deducts a turn if the guess is either too high or too low and provides feedback to the player. If the guess matches the answer, it notifies the player about the correct guess.
2. **set\_difficulty():** This function prompts the user to select a difficulty level – easy, medium, or hard – and sets the number of available attempts (turns) accordingly. It returns the number of turns based on the selected difficulty level.
3. **game():** This function orchestrates the entire game. It initializes the game by welcoming the player and generating a random number within the specified range (1 to 100). It then sets the difficulty level and starts the guessing loop. Within this loop, it takes user input for guesses, checks the guess against the answer, and updates the number of turns accordingly. It continues until the player either guesses the correct number or exhausts all attempts.

These functions work in tandem to provide a structured and engaging gameplay experience. The **check\_answer()** function evaluates the guesses, **set\_difficulty()** configures the game's difficulty, and **game()** ties everything together, managing the game flow from start to finish.

**Algorithm for the Number Guessing Game:**

1. Display Welcome Message
   1. Display a welcome message to greet the player and introduce the game.
2. Generate a Random Number
   1. Generate a random number between 1 and 100 to be guessed by the player.
3. Define Difficulty Levels
   1. Assign a set number of attempts (turns) for each difficulty level (easy, medium, hard).
4. Prompt User for Difficulty Level Selection
   1. Ask the player to choose a difficulty level (easy, medium, hard).
   2. Set the number of attempts based on the chosen difficulty level.
5. Initiate Game Loop
   1. Start a loop to manage the game flow until the player guesses the correct number or runs out of attempts.
6. Display Remaining Attempts and Get User Input
   1. Display the number of remaining attempts to the player.
   2. Prompt the player to enter a guess.
7. Check the User's Guess
   1. Compare the player's guess with the randomly generated number.
   2. If the guess is too high, provide feedback to the player ("Too high").
   3. If the guess is too low, provide feedback to the player ("Too low").
   4. If the guess matches the number, congratulate the player for guessing correctly and end the game.
8. Decrement Attempt Count
   1. Decrease the remaining attempts count after each guess.
9. End Loop
   1. If the player guessed the number correctly, end the game with a congratulatory message.
   2. If the player exhausts all attempts, end the game with a message revealing the correct number.
10. Game Over
    1. Display a game-over message, revealing the correct number if it wasn’t guessed.

This algorithm provides a step-by-step guide outlining the flow of the number guessing game from the initialization phase to the game's conclusion. Adjustments and additional features can be incorporated based on specific requirements or enhancements desired for the game.

**Project Abstract:**

This Python-based Number Guessing Game offers an engaging challenge, prompting players to decipher a randomly generated number within the range of 1 to 100. With adjustable difficulty levels – easy, medium, and hard – and a limited number of attempts, the game fosters an interactive experience for players of varying skill levels. Through a systematic approach of guessing, receiving clues, and strategic deduction, players aim to unveil the concealed number within the allocated attempts, providing an intellectually stimulating and entertaining gameplay encounter.

This abstract highlights the core elements of the project, emphasizing its gameplay dynamics and adaptability to different difficulty levels.

**Source Code:**

The source code of this Python project is shown as follows:

import random

Logo="""

────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

─██████████████─██████──██████─██████████████─██████████████─██████████████────██████████████─██████──██████─██████████████─

─██░░░░░░░░░░██─██░░██──██░░██─██░░░░░░░░░░██─██░░░░░░░░░░██─██░░░░░░░░░░██────██░░░░░░░░░░██─██░░██──██░░██─██░░░░░░░░░░██─

─██░░██████████─██░░██──██░░██─██░░██████████─██░░██████████─██░░██████████────██████░░██████─██░░██──██░░██─██░░██████████─

─██░░██─────────██░░██──██░░██─██░░██─────────██░░██─────────██░░██────────────────██░░██─────██░░██──██░░██─██░░██─────────

─██░░██─────────██░░██──██░░██─██░░██████████─██░░██████████─██░░██████████────────██░░██─────██░░██████░░██─██░░██████████─

─██░░██──██████─██░░██──██░░██─██░░░░░░░░░░██─██░░░░░░░░░░██─██░░░░░░░░░░██────────██░░██─────██░░░░░░░░░░██─██░░░░░░░░░░██─

─██░░██──██░░██─██░░██──██░░██─██░░██████████─██████████░░██─██████████░░██────────██░░██─────██░░██████░░██─██░░██████████─

─██░░██──██░░██─██░░██──██░░██─██░░██─────────────────██░░██─────────██░░██────────██░░██─────██░░██──██░░██─██░░██─────────

─██░░██████░░██─██░░██████░░██─██░░██████████─██████████░░██─██████████░░██────────██░░██─────██░░██──██░░██─██░░██████████─

─██░░░░░░░░░░██─██░░░░░░░░░░██─██░░░░░░░░░░██─██░░░░░░░░░░██─██░░░░░░░░░░██────────██░░██─────██░░██──██░░██─██░░░░░░░░░░██─

─██████████████─██████████████─██████████████─██████████████─██████████████────────██████─────██████──██████─██████████████─

────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

─────────────────────────────────────────────────────────────────────────────────────────────────────────────────

─██████──────────██████─██████──██████─██████──────────██████─██████████████───██████████████─████████████████───

─██░░██████████──██░░██─██░░██──██░░██─██░░██████████████░░██─██░░░░░░░░░░██───██░░░░░░░░░░██─██░░░░░░░░░░░░██───

─██░░░░░░░░░░██──██░░██─██░░██──██░░██─██░░░░░░░░░░░░░░░░░░██─██░░██████░░██───██░░██████████─██░░████████░░██───

─██░░██████░░██──██░░██─██░░██──██░░██─██░░██████░░██████░░██─██░░██──██░░██───██░░██─────────██░░██────██░░██───

─██░░██──██░░██──██░░██─██░░██──██░░██─██░░██──██░░██──██░░██─██░░██████░░████─██░░██████████─██░░████████░░██───

─██░░██──██░░██──██░░██─██░░██──██░░██─██░░██──██░░██──██░░██─██░░░░░░░░░░░░██─██░░░░░░░░░░██─██░░░░░░░░░░░░██───

─██░░██──██░░██──██░░██─██░░██──██░░██─██░░██──██████──██░░██─██░░████████░░██─██░░██████████─██░░██████░░████───

─██░░██──██░░██████░░██─██░░██──██░░██─██░░██──────────██░░██─██░░██────██░░██─██░░██─────────██░░██──██░░██─────

─██░░██──██░░░░░░░░░░██─██░░██████░░██─██░░██──────────██░░██─██░░████████░░██─██░░██████████─██░░██──██░░██████─

─██░░██──██████████░░██─██░░░░░░░░░░██─██░░██──────────██░░██─██░░░░░░░░░░░░██─██░░░░░░░░░░██─██░░██──██░░░░░░██─

─██████──────────██████─██████████████─██████──────────██████─████████████████─██████████████─██████──██████████─

─────────────────────────────────────────────────────────────────────────────────────────────────────────────────

"""

print(Logo)

Easy\_Level\_Turns=15

Medium\_Level\_Turns=10

Hard\_Level\_Turns=5

def check\_answer(guess, answer, turns):

"""checks answer against guess, returns the number of turns remaining"""

while turns>0:

if guess>answer:

print("Too high")

return turns-1

elif guess<answer:

print("Too low")

return turns-1

elif guess==answer:

print(f"Correct! The number I guessed was {answer}")

break

def set\_difficulty():

level=input("Choose your difficulty level. Type 'easy', 'medium' or 'hard':").lower()

global turns

if level=="easy":

return Easy\_Level\_Turns

elif level == "medium":

return Medium\_Level\_Turns

elif level=="hard":

return Hard\_Level\_Turns

def game():

print("Welcome to the number guessing game!")

print("I am thinking of a number between 1 and 100.")

answer=random.randint(1,100)

turns=set\_difficulty()

guess=0

while guess!=answer and turns!=0:

print(f"You have {turns} attempts remaining to guess the number.")

guess=int(input("Enter your guess number:"))

turns=check\_answer(guess, answer, turns)

if turns==0:

print(f"You've lost! The number I guessed was {answer}.")

game()

**Output and Screenshots:**

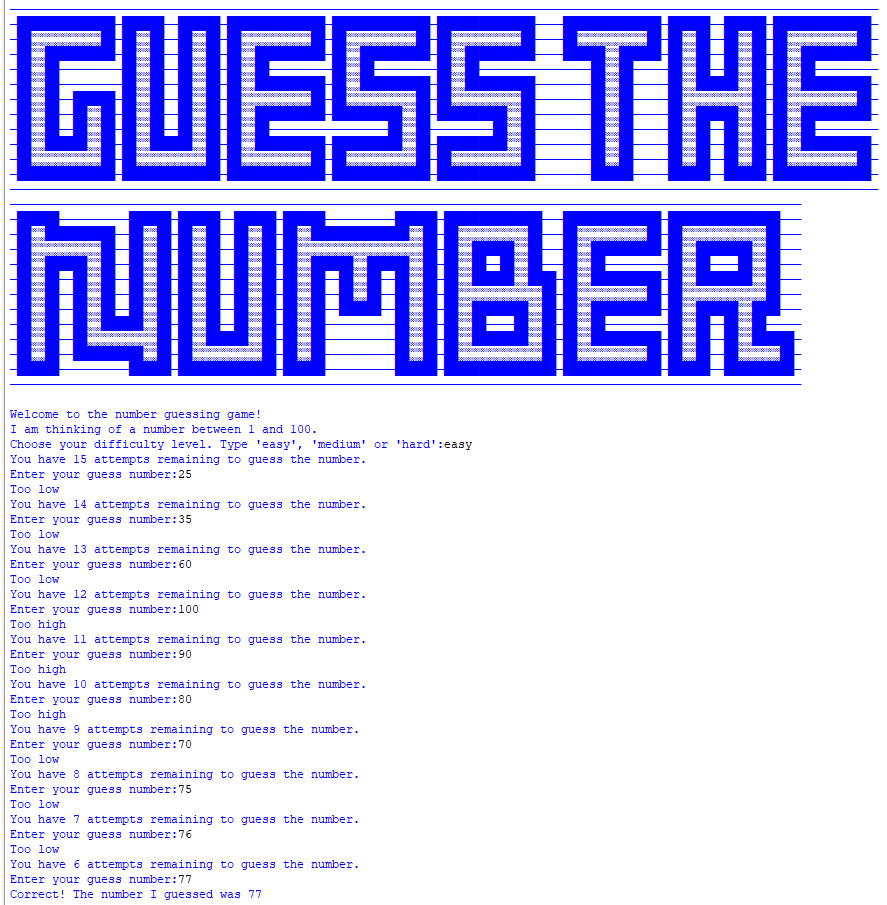
**Sample Output-1:**



**Sample Output-2:**



**Sample Output-3:**



**Sample Output-4:**



**Sample Output-5:**



**Conclusion**

In conclusion, this Number Guessing Game not only entertains players but also serves as a practical introduction to utilizing Python's random module. Through this project, users gain hands-on experience in harnessing the power of randomization within Python. The generation of random numbers for the game's core mechanism provides a tangible example of implementing the random module's functionalities. Players engaging with this game learn firsthand how to employ the random.randint() function to generate random integers within a specified range, fostering a foundational understanding of leveraging Python's randomization capabilities in real-world applications. This immersive experience not only entertains but also equips users with a practical skill set in utilizing the random module for diverse programming needs.

**References**

Following were the references of this Python Project:

1. Udemy.com
2. https://en.wikipedia.org/wiki/Turtle\_graphics
3. Google.com