Report On

Number Guessing Game

Submitted in partial fulfillment of the requirements of the Course project in
Semester IV of Second Year Artificial Intelligence and Data Science

By
Pratik Avhad(Roll No. 1)
Priyanka Bhandari (Roll No. 2)
Charmi Jani (Roll No. 17)

Supervisor
Prof. Bhavika Gharat

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(2023-24)

CERTIFICATE

This is to certify that the project entitled “Number Guessing Game” is a bonafide work of Pratik Avhad (Roll No.),Priyanka Bhandari (Roll No. 2), Charmi Jani (Roll No. 17), submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester IV of Second Year Artificial Intelligence and Data Science engineering.

Prof. Bhavika Gharat	**Dr. Tatwadarshi P. N.**
Supervisor	**Head of Department**

Table of Contents

Abstract ii

Acknowledgement iii

1 Introduction

1.1 Introduction

1.2 Problem Statement & Objectives

1.3 Scope

1.4 Mini Project Contribution

2 Proposed System

2.1 Introduction

2.2 Architecture/ Framework/Block diagram

2.3 Algorithm and Process Design

2.4 Details of Hardware & Software

2.5 Experiment & Result for Verification & Validation

2.6 Conclusion and Future work.

2.7 References

1 Introduction

1.1 Introduction :-

In the realm of recreational programming and gaming, one classic activity is the guessing game. Typically, a player attempts to guess a randomly selected number within a given range

. This engagement provides both entertainment and a challenge. In this context, we introduce a simple Python implementation of such a game.

1.2 Problem Statement and Objective :-

The problem statement revolves around creating an interactive guessing game where the player tries to guess a randomly chosen number within a predefined range. The primary objective is to develop a program that offers an enjoyable and engaging experience for users while enhancing their logical thinking and decision-making skills.

1.3 Scope :-

The scope of this project encompasses the development of a basic guessing game using Python. It includes generating random numbers within a specified range, accepting user input, providing feedback on the guess (whether it is too high, too low, or correct), and looping until the correct guess is made. Additionally, the project involves error handling to ensure smooth user interaction.

1.4. Course Project Contribution :-

This project contributes to understanding fundamental concepts of programming, particularly in Python. It involves aspects such as using random number generation, user input handling, conditional statements, loops, and basic error handling techniques. It also provides insights into problem-solving and algorithmic thinking.

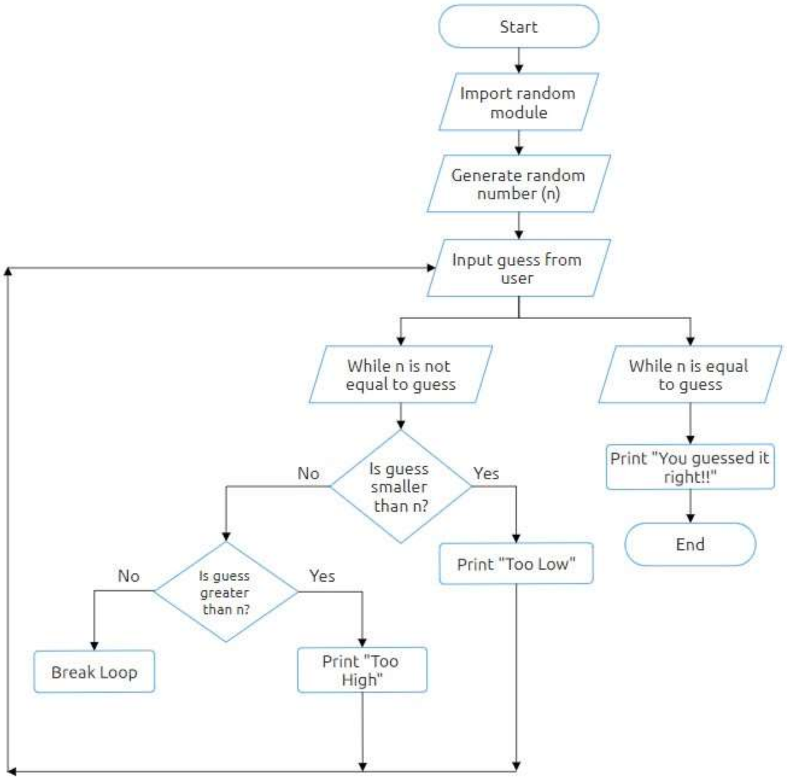
2 Proposed System :-

The proposed system is a Python-based guessing game where the user interacts with the program through the command line interface. The system generates a random number within a specified range and prompts the user to guess the number. Based on the user’s input, the system provides feedback until the correct number is guessed. Upon guessing correctly, the system congratulates the user and terminates.

Existing System (Brief Overview) :-

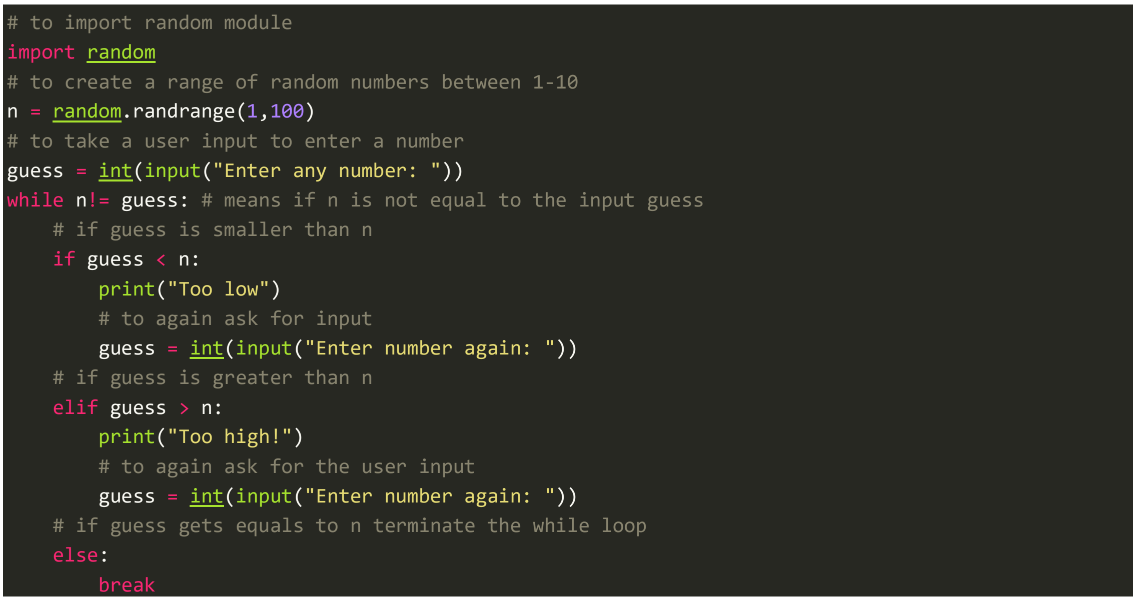
Prior to this project, various implementations of guessing games existed. These ranged from simple console-based programs to more elaborate graphical user interface (GUI) versions. However, this project focuses on a minimalistic console-based approach to introduce beginners to programming concepts without overwhelming them with complexities. The existing system may lack features found in more advanced versions but serves as a solid foundation for learning and understanding basic programming principles.

2.1 Basic Data Flow Daigram :



2.2 Algorithm and Process Design

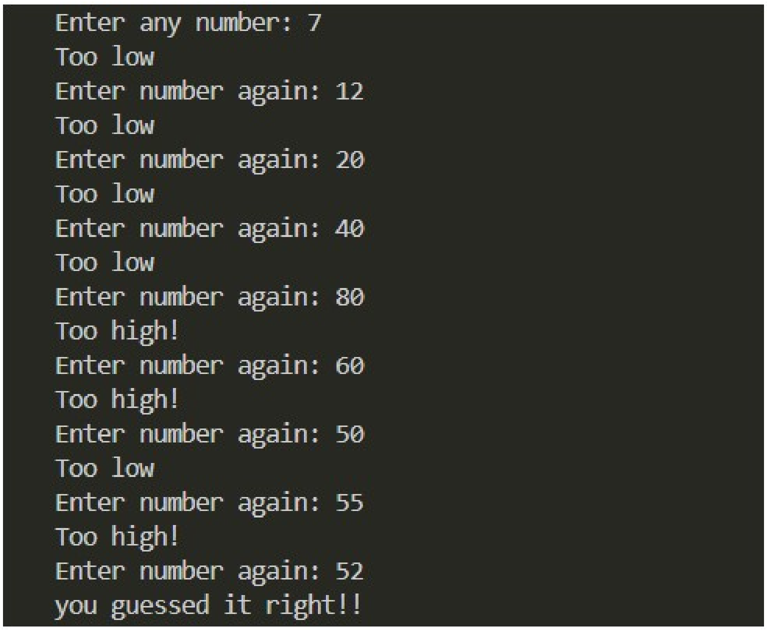
#Code



```
# to import random module
import random
# to create a range of random numbers between 1-10
n = random.randrange(1,100)
# to take a user input to enter a number
guess = int(input("Enter any number: "))
while n!= guess: # means if n is not equal to the input guess
    # if guess is smaller than n
    if guess < n:
        print("Too low")
        # to again ask for input
        guess = int(input("Enter number again: "))
    # if guess is greater than n
    elif guess > n:
        print("Too high!")
        # to again ask for the user input
        guess = int(input("Enter number again: "))
    # if guess gets equals to n terminate the while loop
    else:
        break
```

```
print("you guessed it right!!")
```

Output



```
Enter any number: 7
Too low
Enter number again: 12
Too low
Enter number again: 20
Too low
Enter number again: 40
Too low
Enter number again: 80
Too high!
Enter number again: 60
Too high!
Enter number again: 50
Too low
Enter number again: 55
Too high!
Enter number again: 52
you guessed it right!!
```

Hardware & Software

Hardware :-

1. Edition: Windows 11 Home Single Language.

2. Processor: 12th Gen Intel(R) Core (TM) i5-1240P 1.70 GHz

3. Memory: Sufficient RAM for quick data access(8-16 GB).

4. Storage: Adequate storage space for image and attendance data (256 SSD).

5. Network Connectivity: Internet or intranet access for data sharing (if needed).

6. System type : 64-bit operating system, x64-based processor.

Software :-

1. Vs code

Conclusion :-In conclusion, the number guessing system implemented in Python provides an entertaining and interactive experience for users to guess a randomly generated number within a specified range. Through the utilization of conditional statements, loops, and user input handling, the program effectively guides the user through the guessing process while providing feedback on each attempt. . Additionally, incorporating features such as limiting the number of guesses and providing hints enhances the user experience and adds depth to the gameplay. Overall, this project demonstrates the versatility and practicality of Python in creating engaging and interactive applications.

Future Scope :-

In the future, there's potential to expand the project in several ways. One direction could be to enhance the user interface by incorporating graphics or building a GUI version using libraries like Tkinter or Pygame. Adding difficulty levels, such as adjusting the range of numbers dynamically or introducing time constraints, could make the game more challenging. Overall, there are numerous avenues for growth and improvement to make the guessing game more engaging and versatile.

Refrences :-

1) https://www.shecodes.io/athena/39028-how-to-make-a-simple-guessing-game-in-python
2) https://www.reddit.com/r/math/comments/iyofs6/best_strategy_for_number_guessing_game/?rdt=41944
3) http://opensask.ca/VisualProgrammingEnv/NumberGuessingGame.html