Number Guessing Game Using Java

Project Report

Author: Sohel Datta

Date: 31-12-2024

Abstract

The Number Guessing Game is a simple yet engaging console-based application developed in Java. The game challenges players to guess a randomly chosen number between 1 and 100, providing a fun way to practice logical thinking and number recognition. The game is designed to be interactive, providing feedback on the player's guesses and encouraging continued attempts until the correct number is guessed.

Objectives:

- Develop an interactive console application that allows users to guess a number.
- Provide immediate feedback on each guess to guide the player.
- Implement a loop that continues until the player successfully guesses the correct number.
- Enhance user engagement through clear instructions and responses.

Game Mechanics

1. **Random Number Generation**:

   The computer randomly selects a number between 1 and 100 at the start of the game.

2. **User Input**:

   The player inputs their guess through the console.

3. **Feedback Mechanism**:

   - If the guessed number is greater than the computer's number, the game responds with: "The Number you have guessed is Larger than My number."
   - If the guessed number is smaller, it responds with: "The Number you have guessed is Smaller than My number."
   - If the guessed number matches the computer's number, it congratulates the player with: "Yeah!! You have guessed the Correct Number."

4. **Replay Option**:

   After guessing correctly, players can choose to play again or exit the game.

Technical Requirements

- Programming Language: Java
- Development Environment: Any IDE that supports Java (e.g., IntelliJ IDEA, Eclipse).
- Input/Output: Use of console for user interaction.

Implementation Steps

1. **Set Up Project**:

   Create a new Java project in your chosen IDE.

2. **Random Number Generation**:

   Use java.util.Random to generate a random number between 1 and 100.

3. **User Interaction**:
   - Prompt the user to enter their guess.
   - Capture user input using Scanner.

4. **Game Logic**:
   - Implement conditional statements to compare user guesses with the generated number.
   - Provide feedback based on user input.

5. **Looping Mechanism**: Use a loop to allow continuous guessing until the correct number is found.
6. **End Game Conditions**: Provide an option for users to replay or exit after guessing correctly.

Code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // Mini Project

        Scanner sc = new Scanner(System.in);
        int myNumber = (int)(Math.random()*100);
        int UserNumber = 0;

        do{

            System.out.println("Guess My Number(1-100): ");
            UserNumber = sc.nextInt();

            if (UserNumber == myNumber){
                System.out.println("Yeah!! U hv guessed the
Correct Number");
                break;
            }
            else if (UserNumber > myNumber){
```

```
            System.out.println("The Number u hv guessed is
Larger than My number");
          }
          else {
            System.out.println("The Number u hv guessed is
Smaller than My number");
          }
       }while (UserNumber >=0 );

       System.out.println("My Number was: ");
       System.out.println(myNumber);
   }
}
```

Conclusion

The Number Guessing Game project serves as an excellent introduction to Java programming concepts such as loops, conditionals, and user input handling. It provides an engaging way for users to interact with basic programming logic while enhancing their problem-solving skills. This project can be easily expanded with additional features such as difficulty levels or score tracking, making it a versatile foundation for future enhancements.