10.48047/jocaaa.2024.32.02.76

Number Guessing Game Using Java Programming Language

Rakesh Kumar Giri

Associate Professor,
Department of Computer Science & Engineering,
Saisha Institutions,Chennai, India
Email: rakeshgiriap@gmail.com

Received: 05.12.2023 **Revised: 15.01.2024** **Accepted: 05.02.2024**

ABSTRACT: All technologies in the current world have achieved their pinnacle of viewpoint, particularly computer technology. These days, everyone's life is significantly impacted by game applications. The gaming industry and gaming applications are growing more and more popular every day. The purpose of this article is to design, develop, and create a computer game called Numeric Guess, a guessing game to test the mind that should be enjoyable to play during leisure and times of relaxation. A desktop computer game is called The Numeric Guess. A guessing game is one in which an object has the ability to locate or uncover information. In a similar vein, The Numeric predict is a guessing game in which a number within a predetermined range is fixed at random. The player's goal is to predict the number in a restricted number of tries. The player can select the user chosen range or the default range to play this game.

Keywords: Java, Password, Generator, Cyber Security, Random, Interface

I.INTRODUCTION

The gaming industry is a unique industry that follows the development of computer technology. We know that there are many devices for playing games, but the computer system is the one that always comes first, whatever devices become more fashionable. Game is the best anti-stress remedy for everyone. There are lots of type guessing games available in the gaming industry such as Charades, 30 seconds, Battleship, Botticelli, and more . In Charades, a player can act a sentence or a word in front of another group of players or another single player, he or they might be able to guess the sentence or word of the player who acts. Our game has the same concept as guessing games, but the special concept we are using here is to guess an exact number. For this, we have developed NUMERIC GUESS using Java and Html and CSS for front end.

The system generates a random number from a given range which is not visible to users.The user is prompted to enter their given number in a displayed dialogue box.The computer then tells if the entered number matches if it matches then the user will win. If the number doesn't match then it displays whether the guessed number is higher/lower than the generated number. The game continues under the user guessing the number. User guesses a randomly generated number within a range.
This game generates a random number, and the player attempts to guess it within a set range (e.g., 1–100). After each guess, the program tells the player if their guess is too high, too low, or correct.

Input: User's guess of a number within a range (e.g., 1–100).
Output: Indication of whether the guess is too high, too low, or correct.

A number-guessing game in Java is a simple program where the computer randomly selects a number, and the user has to guess it within a limited number of attempts. The program provides feedback on whether the guessed number is too high or too low, guiding the user toward the correct answer.

This project is an excellent way for beginners to practice loops, conditionals, user input handling, and random number generation in Java.

2.RULES OF THE GAME

10.48047/jocaaa.2024.32.02.76

- If the guessed number is bigger than the actual number, the program will respond with the message that the guessed number is higher than the actual number.
- If the guessed number is smaller than the actual number, the program will respond with the message that the guessed number is lower than the actual number.
- If the guessed number is equal to the actual number or if the **K** trials are exhausted, the program will end with a suitable message.

3. IMPLEMENTATION DETAILS

- The program generates a random number between a predefined range (e.g., 1 to 100).
- The user has limited attempts (K tries) to guess the number.
- At each guess, the program provides a hint:
    - If the guessed number is higher, it tells the user to guess lower.
    - If the guessed number is lower, it tells the user to guess higher.
- If the user guesses correctly, they win.
- If all attempts are exhausted, the game reveals the correct number.

4. APPROACH

To implement the game, we need to follow the steps listed below:

- Generate a random number using the Math.random() method.
- Take user input using Scanner.
- Compare the guess with the random number and provide feedback.
- Limit the number of attempts using a loop.
- Handle edge cases (e.g., invalid inputs).
-

Implementation:
The below Java program demonstrates the number-guessing game:

```
// Java Program to demonstrates the Number guessing game
import java.util.Scanner;

public class NumberGame {

    public static void guessingNumberGame()
    {
        Scanner sc = new Scanner(System.in);

        // Generate a random number between 1 and 100
        int number = 1 + (int)(100 * Math.random());

        // Number of attempts
        int K = 5;

        System.out.println(
            "A number is chosen between 1 and 100.");
        System.out.println(
            "You have " + K
            + " attempts to guess the correct number.");

        // Loop for K attempts
        for (int i = 0; i < K; i++) {
            System.out.print("Enter your guess: ");
            int guess = sc.nextInt();

            // Check conditions
```

10.48047/jocaaa.2024.32.02.76

```
            if (guess == number) {
                System.out.println(
                    " Congratulations! You guessed the correct number.");
                sc.close();

                // Exit function if guessed correctly
                return;
            }
            else if (guess < number) {
                System.out.println(
                    " The number is greater than " + guess);
            }
            else {
                System.out.println(
                    " The number is less than " + guess);
            }
        }

        // If the user runs out of attempts
        System.out.println(
            "You've exhausted all attempts. The correct number was: "
            + number);
        sc.close();
    }

    public static void main(String[] args)
    {
        guessingNumberGame();
    }
}
```

Output:

```
"C:\Program Files\Java\jdk-18\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024
A number is chosen between 1 and 100.
You have 5 attempts to guess the correct number.
Enter your guess: 50
 The number is greater than 50
Enter your guess: 89
 The number is greater than 89
Enter your guess: 74
 The number is greater than 74
Enter your guess: 79
 The number is greater than 79
Enter your guess: 81
 The number is greater than 81
You've exhausted all attempts. The correct number was: 95
```

Note: We can also use **Random** class from **java.util** instead of **Math.random()** to generate random numbers in Java.

```
Random random = new Random();
int number = random.nextInt(100) + 1;
```

10.48047/jocaaa.2024.32.02.76

Number Guessing Game with Unlimited Rounds and Score Tracking

Approach:

- After Exhausting K attempts, the user is given another chance to continue the game.
- The game tracks the score (number of attempts taken to guess correctly).
- The game continues until the user guesses the correct number.

Implementation:

The below Java program demonstrates the number-guessing game with ***Unlimited Rounds and Score Tracking.***

```
// Number guessing game with
// Unlimited Rounds and Score Tracking
import java.util.Scanner;

public class NumberGuess {
    public static void guessingNumberGame()
    {
        Scanner sc = new Scanner(System.in);

        // Generate a random number between 1 and 100
        int number = 1 + (int)(100 * Math.random());

        // Track the number of attempts
        int attempts = 0;

        // Maximum attempts per round
        int K = 5;
        boolean guessedCorrectly = false;

        System.out.println(
            "A number is chosen between 1 and 100.");
        System.out.println(
            "You have " + K+ " attempts per round to guess the correct number.");

        while (!guessedCorrectly) {

            // Give the user K attempts per round
            for (int i = 0; i < K; i++) {
                System.out.print("Enter your guess: ");
                int guess = sc.nextInt();
                attempts++; // Increment attempt count

                if (guess == number) {
                    System.out.println(
                        "Congratulations! You guessed the correct number in "
                        + attempts + " attempts.");
                    guessedCorrectly = true;
                    break;
                }
                else if (guess < number) {
                    System.out.println(
                        "The number is greater than "
                        + guess);
                }
                else {
                    System.out.println(
                        "The number is less than " + guess);
```

10.48047/jocaaa.2024.32.02.76

```
                }
            }

            if (!guessedCorrectly) {

                // Ask the user if they want to continue
                // after exhausting K attempts
                System.out.println("You have used all " + K
                                   + " attempts.");
                System.out.print(
                    "Do you want to continue guessing? (yes/no): ");
                String response = sc.next();

                if (!response.equalsIgnoreCase("yes")) {
                    System.out.println(
                        "Game Over! The correct number was: "
                        + number);
                    break;
                }
            }
        }

        sc.close();
    }

    public static void main(String[] args)
    {
        guessingNumberGame();
    }
}
```

Output:

```
"C:\Program Files\Java\jdk-18\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024
A number is chosen between 1 and 100.
You have 5 attempts per round to guess the correct number.
Enter your guess: 77
The number is less than 77
Enter your guess: 50
The number is greater than 50
Enter your guess: 55
The number is greater than 55
Enter your guess: 65
The number is greater than 65
Enter your guess: 68
The number is greater than 68
You have used all 5 attempts.
Do you want to continue guessing? (yes/no): yes
Enter your guess: 75
The number is less than 75
Enter your guess: 70
The number is less than 70
Enter your guess: 69
Congratulations! You guessed the correct number in 8 attempts.

Process finished with exit code 0
```

10.48047/jocaaa.2024.32.02.76

5. CONCLUSION

Based on the evaluation and analysis, we can determine the following: The game's application design is intended to provide entertainment for the player during their free time. It created and executed an easy-to-use environment, and the game interface is more appealing. This number guessing game can be enjoyed by anyone, regardless of age. In the Java language, students can gain a clear understanding of how to develop a project using Java on Eclipse. The Numeric Guess Game is an engaging, challenging, and straightforward game as well.

6. REFERENCES

1. Shouthry p tigers and Goats A Board Game for Androids, International journal of Scientific and Research publications 2017;7(10);59-64.

2. Geer David. Eclipse becomes the dominant Java IDE. IEEE Computer. 2005;38:16-18.

3. “Guessing”. Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. date last updated; 2019.

4. . Dong Ying. Design and evaluation of Java game programming environment; 2020

5.Johnston J (1972): Econometrics methods. New York: McGraw-Hill.

6.Kalakota, R., A.B. Whinston (1996). Frontiers of Electronic Commerce, Addison-Wesley, Reading, MA.

7.Keeney, Ralph L. (1999). The Value of Internet Commerce to the Consumer. Management Science 45(4) 533-542.

8.Green, P.E. and V. Srinivasan (1978). Conjoint Analysis in Consumer Research: Issues and Outlook. The Journal of Consumer Research 5(2) 103-123.

9. Jarvenpaa, S. and P. Todd (1997). Consumers’ reactions to electronic shopping on the World Wide Web. International Journal of Electronic Commerce 1(2) 59-88.

10.Harris Interactive (2001). Consumer privacy attitudes and behaviors survey wave II, The Privacy Leadership Initiative. http://www.understandingprivacy.org/content/library/ harris2-execsum.pdf.

11.Hair, J.F. (1992). Multivariate Data Analysis with Readings. Third ed. New York, NY, USA: Macmillan Publishing Company.

12.Hair, J.F., Jr, R.E. Anderson, R.L. Tatham and W.C. Black. (1995). Multivariate Data Analysis with Readings. Upper Saddle River, NJ: Prentice Hall.