GUESS THE NUMBER GAME

C Programming Mini Project Report

Student Name: K .Ganesh

Register Number: 921724102046

Department:B.E.CSE

College Name:S.I.T

Course: Programming in C

Trainer:Mr M.Pugazhmara

Submission Date: 25\02\26

Abstract

This project is a simple number guessing game developed using the C programming language. The program generates a random number between 1 and 10, and the player must guess the number within five attempts. After each guess, hints are provided and the score is maintained across rounds.

Introduction:

This project demonstrates an interactive game that enhances logical thinking and introduces random number generation and user interaction using the C programming language.

Objectives:

1 Understand structured programming concepts
2 Use loops and conditional statements
3 Implement random number generation
4 Improve logical thinking and problem-solving skills
5 Develop user-interactive programs

Tools & Technology:

1 Programming Language: C
2 Compiler: GCC / CodeBlocks / Turbo C
3 Platform: Windows / Linux

System Requirements:

1 Hardware: Computer with minimum 4GB RAM
2 Software: C Compiler and any IDE/Text Editor

Algorithm:

1 Start the program
2 Generate a random number between 1 and 10
3 Allow five attempts to guess the number
4 Compare the guess with the generated number
5 Display hints (Too High / Too Low)
6 Update score if correct
7 Ask the user to play again
8 Display the final score
9 End the program

Flowchart:

Start → Generate Number → User Guess → Compare → Hint / Correct → Score Update → Play Again? → End

Program Code:

```
#include  <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {

int number, guess;
int attempts;
    int score = 0;
    char playAgain;

    srand(time(0));

    do {
        attempts = 0;
        number = rand() % 10 + 1;

        printf("\n=== Guess the Number Game (1-10) ===\n");
        printf("You have 5 chances to guess the number.\n");

        while(attempts < 5) {
            printf("Enter your guess: ");
            scanf("%d", &guess);
            attempts++;

            if(guess > number)
                printf("Too High!\n");
            else if(guess < number)
                printf("Too Low!\n");
            else {
                printf("Correct! You guessed in %d attempts.\n", attempts);
                score++;
                break;
            }
        }

        if(guess != number)
            printf("Sorry! The correct number was %d\n", number);

        printf("Current Score: %d\n", score);

        printf("Do you want to play again? (y/n): ");
        scanf(" %c", &playAgain);

    } while(playAgain == 'y' || playAgain == 'Y');

    printf("\nFinal Score: %d\n", score);
    printf("Thank you for playing!\n");
```

```
    return 0;
}
```

Sample Output:

```
=== Guess the Number Game (1-10) ===
You have 5 chances to guess the number.
Enter your guess: 5
Too Low!
Enter your guess: 9
Too High!
Enter your guess: 7
Correct! You guessed in 3 attempts.
Current Score: 1

Do you want to play again? (y/n): n

Final Score: 1
Thank you for playing!
```

Result:

The program executed successfully. The game correctly generated random numbers, provided hints, and maintained the score.

Applications:

1. Educational tool for beginners learning C programming
2. Demonstrates loops, conditions, and random number usage
3. Improves logical thinking and problem-solving skills
4. Can be extended into graphical or mobile games
5. Useful for understanding basic game development concepts

Conclusion:

This project demonstrates fundamental C programming concepts including loops, conditions, random number generation, and user interaction.

Future Enhancements:

1. Increase difficulty levels
2. Add graphical interface
3. Store high scores
4. Add timer-based challenges
5. Expand number range

References:

C Programming textbooks
Online tutorials
Classroom notes