GAME CODE

B32VB-Praxis Programming

Larry Tumaini Laiser

H00481465

Professor: Girish Balasubramanian

INTRODUCTION

This project presents the number guessing game which was made using C programming language. This game's aim to let a player guess a random from a specific range of numbers. Which contains different difficulties for each level it has. The primary main of this project is to obtain knowledge on some fundamental C components including input/output functions, loops, arrays, conditional statements, functions and many more. It also helps in making real time decision making which gives a more creative ways on doing things.

METHODOLOGY

This section describes the actions taken to carry out the project. It contains two subsections: the research methodology and the development methodology.

Research Methodology

Initially, research was conducted to enhance the knowledge of approaches to developing a number guessing game in the C programming language. Various online tutorial guides and resources were researched, especially those focusing on basic concepts such as loops, conditional statements, random number generation, and processing user input. Websites like GeeksforGeeks, Stack Overflow, and official C programming tutorials had insightful information and real-world examples.

In addition, an investigation was carried out on existing instances of guessing games written in C. This research enabled the envisioning of open game structures, flexibility in levels of difficulty, and effective means of player interaction. Examining these instances permitted the orderly arrangement of features that could make the game more engaging for players, such as multiple levels of difficulty and extensive user feedback.

Development Methodology

After the research, the game was created iteratively, testing frequently to check functionality. The basic structure of the game was first created, with four levels of difficulty decided upon: easy, medium, hard, and impossible. Each level would have a different range of numbers to guess from, progressively making the game harder. In order to ensure that the gameplay was unpredictable, the rand() function in the C standard library was utilized for generating random numbers and was seeded using srand(time(0)) with the current system time. Next was implementing user input and interaction. Input from the player was taken using scanf() and stored as strings, which made it convenient to manage odd commands such as enabling the player to quit by typing 'q' or 'Q'. Then, the input was converted to integers and matched against the target number that was chosen randomly.

The main game loop gave the user immediate feedback on each guess, telling them if it was too high, too low, or right, and also how many lives they have remaining. If the player guesses right, they can move on to the next level or quit the game. Running out of lives prompted an option to restart the game or quit. Throughout development, frequent testing was performed to identify and resolve any bugs or logical errors.

Finally, the code was refined by adding comments and improving readability to ensure it was easy to understand and maintain for future use.

FLOW CHART

```
+---------------------------+
| Start Game                |
+---------------------------+
            |
            v
+---------------------------+
| Initialize Level & Lives  |
+---------------------------+
            |
            v
+---------------------------+
| Generate Random Number    |
+---------------------------+
            |
            v
+---------------------------+
| Prompt User for Input     |
+---------------------------+
            |

            v

+----------------------------+

| Input "q"?                 |-- Yes --> Quit Game

+----------------------------+

            |

            v No

+----------------------------+

| Convert Input to Integer   |

+----------------------------+

            |

            v

+----------------------------+

| Compare Guess              |

+----------------------------+

   |        |        |

   v        v        v

Too Low   Too High  Correct!

   |        |        |
   v         v         v
Lives--   Lives--    Win Flag
   |         |         |
   v         v         v
+---------------------------+
| Lives Left?               |
| No --> Game Over          |
+---------------------------+
             |
             v
+---------------------------+
| Restart or Quit           |
```

CONCLUSION

This project has really helped in setting foundation on later applications of C programming language. It also helped to understand/learn some key language features like random number generations, level based game play and arrays. This knowledge can be useful in later studies on different programming languages.

REFERENCES

ABCya (2024) *Guess the Number Game.* Available at: https://www.abcya.com/games/guess_the_number .

Kernighan, B. W. and Ritchie, D. M. (1988) *The C Programming Language.* 2nd edn. Englewood Cliffs: Prentice-Hall.

TutorialsPoint (2024) *C - Functions and Recursion.* Available at: https://www.tutorialspoint.com/cprogramming/c_functions.htm .

GeeksforGeeks (2024) *rand() and srand() in C/C++.* Available at: https://www.geeksforgeeks.org/rand-and-srand-in-ccpp/ .