CSE115 Section 02 Group 10
Fariba Ahmed, Tawsif Al Imam Ayon, Toufik Omar, Safwat Sadiq, Nakib Uddin Azadi.

Word Search Generator

Abstract

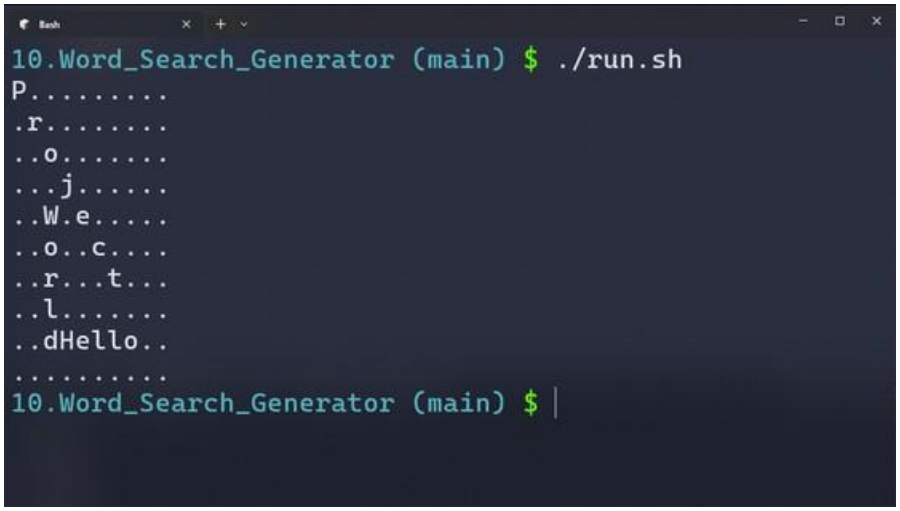
This project implements a word search puzzle generator in C language of computer programming. In which a fixed 10×10 grid is created and filled with dots before placing words horizontally, vertically, and diagonally. The generator serves as an educational tool. As the project is written in C program, basic operations and functions of C program have used in this program. However, the project is still in the development

Index Terms

Word Search Generator, C programming, grid generation, algorithm design, educational tool, array manipulation, string handling.

Introduction

The project's objective is to develop a program that generates word search puzzles from a given list of words. The current implementation uses a fixed grid and supports word placements in three orientation. Horizontally, vertically and diagonally.



Current Status

1.Functionality: The program creates a 10 x 10 grid data structure and then populates it with placeholder characters(in this case that is dot “.”) and after that it places predefined words in that grid vertically, horizontally and diagonally

2.Testing: The program has been compiled and tested; words appear correctly when placement conditions are met.

3.Limitations: The grid size is fixed; overlapping words are not currently managed; and the placement directions are limited to forward placements only.

Target Work Output

a 20x20 grid with 10 predefined words randomly generated in different orientation and then placed in the grid. and the rest of the grid will be filled with random capital alphabets. A fancy ascii art that says

Word Search Generator and will be placed in top of the grid. at the bottom of the grid there will be a printed text saying Enter a word and also beside that there will be a counter for the words that the used guessed. Input: User can go thought the grid and if they find any words. They can simply type it in any format(Capital letter, small letter or combination of both) If they guessed it correctly the program will say Word Found and then remove the word from the grid. If they guessed a word that is not in the grid then the program will say Word not found. And the program repeats itself until the user guess all 10 words

Observation

During testing, the generator successfully populated the grid with dot character and placed words according to predetermined rules. However, manual stepping in was sometimes required to avoid overlaps, especially when multiple words was placed in similar regions of the grid.

Performance Evaluation

- The current implementation runs in negligible time on a 10×10 grid.
- Memory usage is minimal.
- The linear search for placement positions works well on the current scale.

Discussions & Conclusion

The code includes standard input/output and string handling libraries and defines a 10x10 grid which initializes by filling with '.' characters to represent empty spaces. The code prints the grid row by row, allowing visualization of word placements. The code prints the words horizontally, vertically and diagonally within the grid. This project provides an efficient, modular approach for word placement within a structured grid. The current version achieves its basic objectives as the project is still in development.

Citation : 0

References

For knowledge :

1.https://www.w3schools.com/c/index.php was used to understand the basic syntax of certain C keywords and functions.

2. https://www.geeksforgeeks.org/c-programming-language/ was used to gain more knowledge about arrays and pointers.

3. https://www.learn-c.org/en/Pointers to learn better implementation of pointers in the program.

For writing structure (Scientific Journals):

1. Markov Logic Networks by Matthew Richardson and Pedro Domingos.

2. Extreme Learning Machine by Guang Huan,Qin Yu Zhu,Chee Kheong Siew.

Writing Correction: ChatGPT