

Project Report

WordMania!

CS 101 Project, Autumn Semester

Slot 11, Group 13, Group TA: Prakhar Gupta

Project Team Members:

Goutham Ramakrishnan (Team Leader): 140020039

Omkar Kajrolkar: 140020014

Rajat Yadav:140020059

Chetan Thakkar:140020004

ACKNOWLEDGEMENT:

We would like to thank Prof. Deepak Phatak and Prof. Supratik Chakraborty for giving us the opportunity and freedom to do the CS101 project on a topic of our choice, giving us a platform to use the programming skills we have learnt over the duration of this course.

INTRODUCTION TO PROJECT:

The topic chosen for the project is to create a wordsearch game, using the C++ programming language. We have called it "WordMania!". A wordsearch consists of a number of words hidden in a grid of letters. The objective of the game is to find all the hidden words as soon as possible.

Wordmania! is a word puzzle that normally takes the form of a square grid in which random words have been hidden. It is a vocabulary based puzzle game, which tests the users quick thinking. This is word game which is played on 15*15 grid where words are arranged horizontally and vertically. The aim of the game is to identify the hidden words from the grid given, as fast as possible. It develops and sharpens the users' thinking ability. Word Mania is a program that will undoubtedly appeal to a wide range of audiences.

➔ PURPOSE:

- a> To create a fully functional Wordsearch game with a pleasing interface and good usability.
- b> Secondary purpose of project was to gain some knowledge of OOPS concepts, data file handling in C++, and to sharpen our programing skills.
- c> Purpose of the project is to develop mind skills, sharpen thinking ability and much more.

SCOPE

It has the scope of choosing 10 words randomly from a pre existing database of words, and arranging them into a 15x15 grid as per the specifications of the level. The program can interact with the user, as the user can choose letters from the grid shown, indicating it as a part of their solution grid. After the user finishes choosing his solution or when the time runs out, the ideal solution is displayed. The program then calculates the points scored by the user and updates the highscore file that is maintained externally.

PROJECT DESCRIPTION

1. MAIN INTERFACE

As the program runs “WELCOME TO WORDMANIA” is displayed on the screen. Main menu window will be displayed which comprises of ‘START GAME’, ‘LEVEL DESCRIPTION’, ‘HOW TO PLAY’, and ‘EXIT GAME’ options. On clicking ‘START GAME’, the user is asked to enter his name. After this, difficulty levels such as ‘EASY’, ‘MEDIUM’, ‘HARD’ will be displayed on the screen and user can select the level he wants to play.

2. GRID GENERATION

On choosing the difficulty level, 15*15 grids will be displayed in which words will be randomly placed. User has to find the words which will be displayed in the box. A total of 10 words have to be found out by the user. Difficulty level of words gets increased as difficulty level increases. The words are chosen randomly from a pre existing database.

3. LEVEL DESCRIPTION

There are three levels in the game namely 'EASY', 'MEDIUM', 'HARD'. In each level, words are hidden in a 15*15 grid. The words to be found are provided to the player. In each level, the arrangements of hidden words are different. The alignment of words in each level is unknown initially. The possible four alignments are A. Left to right, B. Right to Left, C. Up to Down, D. Down to up.

1) EASY LEVEL:

All words in either alignment A or in alignment C (only one of two)

2) MEDIUM LEVEL:

All words in either alignment B or D (Only one of the two)

3) HARD LEVEL:

All words are arranged either horizontally or vertically.(i.e. A and B together or C or D together)

The first task is to find the alignment of the words. Once the word is found, the alignment of other words is automatically obtained as above.

4. TIMER

Word game has to be solved in 120 seconds. And as per the number of letters in the words displayed in the box that has been found out by the user, score will be awarded to him. And if user fails in finding all the letters of those words in a particular time, points will deducted as per the time taken by the user. And score will be displayed on the screen. The time left displayed below the word search is updated every 10 seconds.

5. SELECTION OF WORDS

The way in which user can select letters is to give input sequentially by clicking on the letter of the given word provided. Points will be awarded as per number of correct letters selected. As the letter is selected, it is marked in black.

6. RETURNING TO MAIN MENU AND EXITING GAME

If in between the game user wants to return to main menu for some reason, there is an option called 'Back to main menu'. Player can exit the game in between by clicking on the option 'EXIT'.

7. HOW TO PLAY

Player is been given maximum of 120 seconds to find the ten hidden words. Clicking on a letter in the word search selects it as a part of your solution. Each correctly selected letter awards you 10 points. Each wrongly selected letter deducts 3 points from your score. For each letter that was not selected by you, but was a part of the correct solution, 2 points are deducted. For every second taken to solve the word search, 1 point is deducted. Time left displayed below the word search is updated every 10 seconds. On clicking the 'SUBMIT SOLUTION' button, all correctly selected letters become green. All wrongly selected letters become red. Letters which were part of the solution but not selected are marked in black.

The score is been displayed on the screen, along with the relevant parameters that were used to calculate it.

8. QUIT

User can quit the game on clicking on 'EXIT GAME', and can leave the game.

FUNCTIONS USED:

Listed below are the prototypes of the functions used in the project. Their purpose has been described in the SRS documentation.

1. `string* easywords(string words[], int a[]);`
2. `string* medwords(string words[], int a[]);`
3. `string* hardwords(string words[], int a[]);`
4. `int* randomarray(int a[10],int x);`
5. `int* fixdup(int a[10], int x);`
6. `int checkdup(int a[10]);`
7. `string* gottenwords(string words[10], int x);`
8. `void WordsIntoGrid(char grid[][15],int align, int align2, string word, int x, int y);`
9. `int generategrid(int level, char grid[][15], string words[]);`
10. `void hashtorandom(char gridsolution[][15]);`
11. `void design();`
12. `void howtoplay();`
13. `void getgrid(int level);`
14. `void highscores();`
15. `void showrules();`
16. `int chooselevel();`
17. `void playgame();`
18. `int main();`

WORK DISTRIBUTION

- 1) **GOUTHAM RAMAKRISHNAN:** Teach/Solve doubts about concepts to team members. Incharge of all algorithms and programming. Built entire program from scratch. Also did much of the documentation.
- 2) **OMKAR KAJROLKAR:** Maintaining diary, Writing parts SRS and Project Report.
- 3) **CHETAN THAKKAR:** Was interested in working but was unable to due to lack of proficiency in C++. Worked mainly on documentation
- 4) **RAJAT YADAV:** Did practically no work for the project.

HARDWARE REQUIREMENTS FOR PLAYING THE GAME:

Mouse with left click function, keyboard.

SPECIFIC REQUIREMENTS:

Codeblocks13.12 version IDE with simplecpp integrated version.

RAM 512 MB

REFERENCES

- I. Lecture Slides of Prof. Supratik Chakraborty and Prof. Deepak Phatak .
- II. An introduction to programming and object-oriented design by James P. Cohoon, Davidson
- III. An Introduction to Programming through C++, Prof. Abhiram Ranade

Websites:

1. www.wikipedia.org (various articles)
2. www.cplusplus.com (various forums and links on this site)
eg. <http://www.cplusplus.com/reference/ctime/ctime/>
<http://www.cplusplus.com/forum/beginner/99196/>
3. www.stackoverflow.com (various threads)
4. <http://cse.iitb.ac.in/~cs101/index.html>
5. www.studytonight.com
6. en.cppreference.com
7. www.tutorialspoint.com
8. www.programmingsimplified.com
9. www.hscripts.com
10. www.dreamincode.com
11. www.daniweb.com
12. www.cppforschool.com

Search engine used: www.google.com