

Software Requirement Specification

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

→ **INTRODUCTION:**

A word mania is a word puzzle that normally takes the form of a square or rectangular and random words are been placed. A Word mania is a vocabulary based puzzle game. This is word game which is played on 15*15 grid where words are arranged horizontally or vertically,etc. The aim of the game is to identify all the meaningful words from the grid given.

→ **PURPOSE:**

1. To create a fully functional Wordsearch game with a pleasing interface and good usability.
2. Secondary purpose of project was to gain some knowledge of OOPS concepts, data file handling in C++, and to sharpen our programing skills.
3. 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.

➤ **General description of Project**

PERSPECTIVE VIEW OF THE PROJECT

The project consists of C++ based game “WordMania!”. User can find different types of words placed in the grid using arrow keys. It is an addictive game where you have to find as many word as possible and develop brain skills.

OVERVIEW

Our project is to display the grid in which words are randomly arranged and allow the end user to solve it. Words may be arranged in any manner, goal is to find as many words as possible. Words are too arranged with varying level of difficulties.

BASIC STRUCTURE OF PROJECT

- I. Displays main menu window of the game which contains 'Start Game', 'Level description', 'How to play', 'Highscores' and 'Exit Game'.
- II. User is asked to enter name. On pressing enter, user can select difficulty level he wants in the game. Modes of difficulty are 'Easy', 'Medium', 'Hard'.
- III. Based on the mode user has selected now a window is displayed which has grid containing words. There will be 10 words that user has to find out in the grid as per difficulty level user has chosen and the words will be displayed near it.
- IV. A particular time is allotted to user to find the words

In the grid and points are allotted as per no of letters clicked by the user correctly of that word. And points are deducted for choosing the wrong letter of that word or by choosing another letter that was not in the word.

For every second taken to solve the word search, one point is deducted.
- V. The way in which user can select the word is by clicking the letters that are in that word. If user finds all the words and there is time left, there is option of 'SUBMIT SOLUTION'. The game ends when time runs out or when this button is clicked.

- VI. User can also return to the 'Main menu' in between the game so that user can choose the difficulty level as per required or also can quit the game.
- VII. 'Word game' can be closed by clicking on the 'Exit'.

- **Hardware requirements for playing the game**

Mouse with left click function, keyboard.

- ***SPECIFIC REQUIREMENTS***

Codeblocks13.12 version IDE with simplecpp integrated version.

Windows 7/8 OS to support it.

RAM 512 MB

Functions used:

1. **string* easywords(string words[], int a[]);**

This function contains an array of 50 easy words and returns 10 of them from the indices as specified in the integer array passed as parameter.

2. **string* medwords(string words[], int a[]);**

This function contains an array of 50 medium words and returns 10 of them from the indices as specified in the integer array passed as parameter.

3. **string* hardwords(string words[], int a[]);**

This function contains an array of 50 hard words and returns 10 of them from the indices as specified in the integer array passed as parameter.

4. **int* randomarray(int a[10],int x);**

This function returns an array with 10 randomly generated numbers between 0 and x.

5. **int* fixdup(int a[10], int x);**

This function fixes duplicate values present in the array passed as parameters.

6. **int checkdup(int a[10]);**

This function checks for duplicate values in the array passed as parameters.

7. **string* gottenwords(string words[10], int x);**

This function returns an string of 10 randomly chosen words of the chosen level.

8. void WordsIntoGrid(char grid[][15],int align, int align2, string word, int x, int y);

This function is called by the generategrid() function, each time a word needs to be hidden in the grid. According to the various parameters passes to it, it ensures the placement of the word into the grid smoothly.

9. int generategrid(int level, char grid[][15], string words[]);

This function coordinates the generation of the solution grid. The 10 words to be hidden are passed as parameters and this function ensures that they are randomly hidden in the grid. It does this by first checking the level, and calling the WordsIntoGrid() function with appropriate parameters.

10. void hashtorandom(char gridsolution[][15]);

This function converts the solution grid into the problem grid. All '#' are replaced by a random letter and hence the problem grid is generated.

11. void design();

This function serves a purely aesthetic purpose. It is responsible for the animation seen in the beginning of the game.

12. void howtoplay();

This function is called from the main function when the How to Play button is clicked. It displays instructions to the user and a button by which he can return to the main menu.

13. void getgrid(int level);

This function is the analogue of the main function in the non graphical program. It coordinates the generation of the list of ten hidden words, the solution grid and the problem grid, using all the relevant functions.

14. void highscores();

This function uses file management resources of c++ to display the high scores of the game read from an external file. It is glitchy and not fully functional.

15. void showrules();

This function is called from the main function when the Description of Levels button is clicked. It displays the description of each level and the points scheme. There is also a button by which he can return to the main menu.

16. int chooselevel();

This function is called from the playgame() function. It provides the functionality of 3 buttons which enables the user to choose the level of difficulty of the game. It returns an integer value corresponding to the difficulty level.

17. void playgame();

This is the largest function of the program and the heart and soul of the main functionality of the wordsearch grid. It performs the following tasks:

1. It first facilitates the input of the users name. This is done by checking for a keyPressEvent.
2. It then facilitates the choosing of the level, by calling the chooselevel() function.
3. It generates the grid by calling the getgrid() function, according to the level chosen by the user.
4. It displays the wordsearch and the words to be found in the next window, along with buttons for exiting, going back to main menu and for submitting solution.
5. It then checks for mouseButtonPressEvent. According to where the user clicked, it acts appropriately.

6. It also maintains a time count from when the user started playing the game, and informs the user with appropriate messages.
7. It provides functionality to the Back to Main Menu and Exit Buttons.
8. On clicking on a letter in the wordsearch, it marks it as a part of the users solution grid and displays a rectangle around it. This rectangle is removed on clicking a selected letter again. This is achieved thorough dynamic memory allocation for the new rectangles.
9. When the time runs out, the functionality of the grid stops and the user is asked to press the submit solution button.
10. On clicking the submit solution button, the users solution grid is cross checked with the actual solution grid. The grids in the wordsearch are appropriately marked showing the user the letter he got right, wrong, and the ones he missed.
11. The users score is calculated and the parameters used to calculate the score are also displayed to the user.
12. The users score is compared to the scores in an external Highscores file and the highscores file is updated as necessary. This part of the code is glitchy and is not functioning as expected.
13. The game ends here. The user can either exit or go back to the main menu by clicking on the appropriate button.

18. int main();

The main function of the program does the following tasks:

1. Initializes the canvas, ensures animation and display of main menu.

2. Coordinates the function of each button present in the main menu by hiding the buttons and calling the appropriate function.
3. The structure of this function ensure the smooth functionality of the Back to Main Menu button.

- **Graphics**

The graphics has been implemented using simplecpp.

The Opening menu has an animation in the beginning. Then buttons appear and the user can proceed as indicated. Some of the windows ecounteed are displayed below in the order of their appearnce.



Start Game
Level Description
How to play
View Highscores
Exit Game



Choose level:

Easy
Medium
Hard



N G B B Q X R P E A V E N T B
Q X W L K A E Z C E V K L U S
E E Q Q K Y D W N D Y C O U Q
Q C E R N H N S E S U M Y U N
T N G L R O U X U T R E E U D
R A I A U P O Z L E C P E X C
O T T D O O F H F L L O S J C
P S V I J Q U C N L P L Q S N
M N I R D G V P I A K E E R O
I I C B A G F Y Y P A V M U I
A T J B E K L I M Z K N K O P
U O J X V R U A L P T E T M M
S W R X Z A W T U P D A L U A
W J O K Y P Q Z J F S I J R H
A L J N M K L M T X V O F U C

120 seconds left!

PALLET
CHAMPION
RUMOUR
ENVELOPE
IMPORT
BRIDAL
FOUNDER
INSTANCE
ADJOURN
INFLUENCE

Submit Solution

Back to main menu	Exit
-------------------	------

N G B B Q X R P E A V E N T B
 Q X W L K A E Z C E V K L U S
 E E Q Q K Y D W N D Y C O U Q
 Q C E R N H N S E S U M Y U N
 T N G L R O U X U T R E E U D
 R A I A U P O Z L E C P E X C
 O T T D O O F H F L L O S J C
 P S V I J Q U C N L P L Q S N
 M N I R D G V P I A K E E R O
 I I C B A G F Y Y P A V M U I
 A T J B E K L I M Z K N K O P
 U O J X V R U A L P T E T M M
 S W R X Z A W T U P D A L U A
 W J O K Y P Q Z J F S I J R H
 A L J N M K L M T X V O F U C

50 seconds left!



N G B B Q X R P E A V E N T B
 Q X W L K A E Z C E V K L U S
 E E Q Q K Y D W N D Y C O U Q
 Q C E R N H N S E S U M Y U N
 T N G L R O U X U T R E E U D
 R A I A U P O Z L E C P E X C
 O T T D O O F H F L L O S J C
 P S V I J Q U C N L P L Q S N
 M N I R D G V P I A K E E R O
 I I C B A G F Y Y P A V M U I
 A T J B E K L I M Z K N K O P
 U O J X V R U A L P T E T M M
 S W R X Z A W T U P D A L U A
 W J O K Y P Q Z J F S I J R H
 A L J N M K L M T X V O F U C

PALLET
 CHAMPION
 RUMOUR
 ENVELOPE
 IMPORT
 BRIDAL
 FOUNDER
 INSTANCE
 ADJOURN
 INFLUENCE

Time taken: 99

Correct letters chosen: 18 Wrong letters chosen: 38

Number of correct letters not chosen: 53

Back to main menu	Exit
-------------------	------

You have scored -139 points

➔ REFERENCES

- 1) Lecture Slides of Prof. Supratik Chakraborty and Prof. Deepak Phatak .
- 2) An introduction to programming and object-oriented design by James P. Cohoon, Davidson
- 3) 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