



# National University

Of Computer and Emerging Sciences



---

Department: BS (SE)

Object Oriented Programming

Assignment 1

Cross Word Match



You are required to develop a game that simulates a 20\*20 crossword puzzle and the user matches the words on the puzzle to earn points. When all the right matches are done the game ends.

For your better understanding here is a picture.

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

### **Problem Description:**

You have to develop a game that has three levels.

1. Easy (10\*10 matrix)
2. Medium (15\*15 matrix)
3. Hard (20\*20 matrix)

In all the three levels a board appears in front of the user and the user has to look for words of English and match them. The dimensions of all the three boards are different. The matches can be three ways.

1. Match right →
2. Match Downwards ↓
3. Match Diagonally ↘

Also, with the board there appears the words that are to be found in the board. For every legitimate match the user is awarded points depending upon the length of words matched. There is a timer running for each game. After all the matches are done the game is won, the board is removed and the total time is displayed. The scoring is done taking in account the score of all the words matched and the time taken to do so. (You have to develop a formula which takes into account the score and time to calculate the total score.)

### **Problem Solution:**

You have various text files in your project that contain set of words of English of different lengths. You are required to do the following.

Load the text file into your project and randomly extract words from it. The words in the files can be in groups depending upon their lengths. For example, your word file might contain words in the following order.

(3 lettered words)

RED

END

FOX

(4 lettered words)

BLUE

GLUE

TRUE

(5 lettered words)

WHITE

RIGHT

SIGHT

And so on....

The words can also be jumbled and not sequenced according to length. For example,

WHITE

RED

BLUE

And so on...

You are required to handle both the scenarios.

**NOTE:** You can use delimiters of your own choice or keep spaces to separate words in your text files.

Once the file is included, you have to read the file and place words in a 2-Dimensional matrix in the eight ways as discussed above. The rest of the spaces in the matrix are filled with random English letters. Once the board is loaded the users can navigate on it using the navigation keys and lock the letters with some key specified. The locking sequence can be done in a way where users hovers on the words, lock the starting character of a word and ending character of the word. Once the word is locked the system matched the word with the ones present in text file included and if accurately matched change the color of the word found and award appropriate points accordingly. If not, the system displays appropriate error message. You can also use any other navigation, locking and matching mechanism for example, mouse. If the word is found, the color of the word on the board is changed and the color of the word in the list of words also changes to grey.

## **Features:**

Your game should use the following features as they contain marks in evaluation.

1. **Menu:** The game should be menu driven and can contain the following options.
  - Game Play
  - Instructions
  - Credits
  - Leaderboard
  - Exit
2. **Timer:** There should be a timer running adjacent to the board which will display the time passed since the game has been executing.
3. **Scoring:** The game should include a scoring mechanism as defined earlier. The scoring is done taking in account the score of all the words matched and the time taken to do so. (You have to develop a formula which takes into account the score and time to calculate the total score.)

**NOTE:** You can also include your own scoring mechanism but you will have to defend it during the evaluation.

4. **Music:** Let us not make it boring for a player to play this game. You have to include two kinds of sound effect. Firstly, there should be a sound track that plays at the back of the game in a loop. Secondly, there should be sound effects for each action namely, navigating on the keyboard, locking/selecting words, successful combinations and unsuccessful combinations etc.
5. **Leaderboard:** The leaderboard should display the scores of top five players from each difficulty levels.
6. **Hints:** The game should provide three-word hints to the user. The user will only highlight the first letter of the potential word that can be in the game.
7. **Bonus:** For the bonus task the user can divide the input to the games into a set of genres for example the user selects a genre say, Drama or cooking, and the only words that appear in the game are related to the genre selected. If the user selects cooking for example, the following words appear.
  - a. Stove
  - b. Kettle
  - c. Shell
  - d. Bake
  - e. Grate
  - f. Fridge
  - g. Pasta

**Note:** This game almost covers the whole course you have studied in the Programming Fundamentals course. You are expected to use the following concepts comprehensively in the game.

- If/else/switch
- Loops
- Functions
- Arrays (1-D and 2-D)
- File Handling.

Also, the game is expected in “sfml” for which tutorials are being arranged. You are expected to develop the game in sfml. It not a bonus but is mandatory to develop the game in sfml.

For a demo, you can explore the following url.

<https://www.mindgames.com/game/Find+Words>

Best Wishes 😊