University of Central Punjab

Faculty of Information Technology

PF Project # 2

Word Search Puzzle

Instructions

- Plagiarism will not be tolerated under ANY circumstances
- Make unit logical functions that are as small as possible
- Comment the code appropriately
- Do not use static memory or global variables
- Use good identifier names

Introduction

Word Search Puzzle is a word game that consists of the letters of words placed in a grid, which usually has a rectangular or square shape. The objective of this puzzle is to find a user defined word in the grid. The words may be placed horizontally, vertically, or diagonally. Words can be available straight forward in rows or columns, in reverse order of letters and maybe straight or reversed diagonally depending on the difficulty.

Main Features (Menu-Based Program)

- New Game
- Resume Game
- Levels (Easy , intermediate ,Hard)
  Note:
  - **Easy level** contains word in single row or single column.
  - **Intermediate Level** contains word in forward and backward direction as well as single row and single column.
  - **Hard level** contains word in forward, backward directions and in diagonals.
- High Record
- Exit

Basic Functionality

1. Read Board

   Read 2D character array from **"board.txt"** file that will be used as board which contains different characters.

   Example:
   board.txt

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

   NOTE: these example is for your understanding actual file don't have blank spaces. E.g: **JFSJRPDKXSERNFD**

2. Board Display

   Display the board on console

3. User record Input

   Take the name of the user in character array.

4. Get word as input:

   Ask player to enter a word.

5. Verify the word

   Verify that word from the **"dictionary.txt"** which contain multiple word.

6. Find word

It will find the user entered word in the board (2D character array). The word may be in a row, column, diagonal or reverse diagonal. The word may be is its order or in reverse order on board.

Example:

HAT (in order word)

TAH (reverse order word)

7. Pause Game

   It will save the current state of the board (2D character array) in a **"Pausedgame.txt"** file.

8. Load Game

   Read 2D character array from **"Pausedgame.txt"**.

9. Save high record

   Store first five highest record in a text file **"Highrecord.txt".**

10. Display high record

    Read record from **"Highrecord.txt"** and display it on console.

Examples of Word in File

Forward row-wise word:

APP

Start at(row 7, col 0)

End at(row 7,col 2)

Backward row-wise word:

GO

Start at(row 14, col 3)

End at(row 14,col 2)

Forward column-wise word:

TAG

Start at(row 14, col 6)

End at (row 14, col 8)

Backward column-wise word:

GATE

Start at (row 14, col 8)

End at (row 14, col 5)

Forward diagonal-wise word:

ILL

Start at (row 2, col 7)

End at (row 4, col 9)

Backward diagonal-wise word:

AM

Start at (row 8, col 4)

End at (row 7, col 3)

Forward reverse diagonal-wise word:

IT

Start at (row 4, col 5)

End at (row 3, col 6)

Backward reverse diagonal-wise word:

IT

Start at (row 7, col 2)

End at (row 8, col 3)

Game Description

This is a menu-based game:

Example:

Press n for new game
Press r for resume game
Press l for level selection
Press h for high score
Press e for exit

New game:

a. In new game first ask player name.
b. Then read board from a text file **"board.txt"**
c. Display the board on console.
d. Ask a word from a player which he wants to search.
e. There are two possibilities either player enters **p or P** for **pause the game** or a word which he wants to search.
    i. If the player enters **p or P** then write the 2D array of board into a text file **"Pausedgame.txt"**.
        1. Now show a new menu

           Press e for exit

           Press r for resume

        2. When player press e simply end the game
        3. When player press r then read 2D array from file **"Pausegame.txt"** and continue the game.
    ii. If the player enters a word perform the following steps:
        1. Verify the word from a text file **"dictionary.txt".**
        2. If the word is found now find that word in 2D array of board.
        3. If word is present in board in any of the above mentioned directions then increase the score by 10.
        4. If word is not present in board in any of the above mentioned directions then decrease the lives by one. **(Note: you have only three lives)**
        5. If your life is 0 then end the game and update the score in text file **"Highrecord.txt"** if required. **(Note: High record contains 5 top records only)**
        6. If the player successfully found all the words from 2D array board then end the game and update the score in text file **"Highrecord.txt"** if required. **(Note: High record contains 5 records only)**

Resume game:

When a player presses r then read 2D array from file **"Pausegame.txt"** and continue the game.

Level selection:

- **Press 1 for easy**
- **Press 2 for medium**
- **Press 3 for difficult**

High score:

Read record from **"Highrecord.txt"** and display it on console.

Exit:

Simple end the game.

Topics which will be used in this Project:

- Functions
- Filling
- Pointers
- 2D Arrays
- Dynamic Memory