**Word Search**

Write a program that will read a list of words from a disk data file called “words.txt”. This file should contain one word per line.

Create a class for this structure called "Puzzle" and utilize methods and variables appropriately in this class. NO PRINTING DIRECTLY FROM THE CLASS – the class methods should only return strings which the main program will utilize/print, etc.

This list of words will then need to be put into a grid (double array) whose size is 45 x 45 where each word is randomly inserted into the array in one of the following directions:

Horizontal, vertical, diagonal-right, diagonal-left

and their corresponding backwards equivalents. There would be a total of 8 possible directions for a word.

Once all the words are inserted into the array, then your program should fill in all the blank cells with random letters. Use ONLY letters that are contained in the Words from the word list.

Once your array is filled, print out the resulting puzzle along with the answer key, which would be the array without the random letters. The puzzle output must be created in a PDF file that contains 2 pages. First page is the puzzle with the words to find under the puzzle. The second page (of the same document) is the KEY. See example PDF.

**Algorithm**

Read the list of words and then sort them by the length of each word from largest to smallest.

This makes it easier to fit in the larger words first. You can use something like this to sort an array called “a”:

a.sort\_by{ |x| x.length }

Select a random row/col to start your placement of the word.

Choose a random “direction” to begin placement.

Start test-placing letters into the cells in the correct direction. You can only test-place a letter if the current cell is either empty or contains the correct letter in the correct position.

If you cannot continue for any reason, then mark the starting cell as “used” and start over with a new random direction. Don’t reuse a direction for the same word.

Once you can successfully place a word, place the word permanently and move on to the next word.

You can keep track of which random row/col that you have already “checked” for a word by using a parallel double array (45x45) to store if the cell is “used” or not. Create a method for getting a new start position. Be sure to reset this checking array for each word.**Example:**

Given a list of words in a file like:

Apple

Banana

Orange

Cantaloupe

Kiwi

Your output would be something like this. (Example is much smaller scale)

Puzzle

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

List of Words to Find:

Apple Cantaloupe

Banana Kiwi

Orange

-------------- New Page ---------------------

Answer Key

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | **I** |  |  |  |  |  |  |  |  |  |
|  | **W** |  | **B** |  | **E** |  |  |  |  |  |
|  | **I** |  |  | **A** | **G** |  |  |  |  |  |
|  | **K** |  |  | **N** |  |  |  |  |  |  |
|  |  |  | **A** | **A** |  |  |  |  |  |  |
|  |  | **R** |  | **N** |  |  |  |  |  |  |
|  | **O** |  |  | **A** | **P** | **P** | **L** | **E** |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **E** | **P** | **U** | **O** | **L** | **A** | **T** | **N** | **A** | **C** |  |

List of Words to Find:

Apple Cantaloupe

Banana Kiwi

Orange