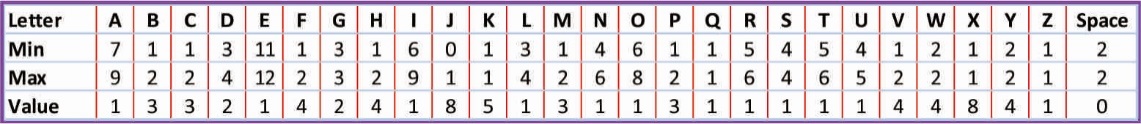
**Scrabble Documentation**

This is the game of construct the words. Where we play the brain to construct the letter for fit into the meaningful words. There are a board which contain 15x15 tiles, all 225 tiles have each value and own characters. There are several criteria and provision:

1. All tiles can be piled up with only one letter.
2. All tiles have their own characters.
   1. DL means Double Letter, which will multiplying the value of a word by 2 times which has placed into that tile.
   2. TL means Triple Letter, which will multiplying the value of a word by 3 times which has placed into that tile.
   3. DW means Double Word, which will be multiplying whole word total value by 2 times which crossed over the tile.
   4. TW means Triple Word, which will be multiplying whole word total value by 3 times which crossed over the tile.
3. Each new word will be check with these provisions:
   1. It can be pile into another letter, even it was an old letter or new letter.
   2. Each word that can be construct in all lines must have a meaning. When the words lied in horizontal direction, will be read from left to right. When the words lied in vertical direction, it will be read from top to bottom.
   3. All words will be check into AJAX URL, where will be processed on server-side.
   4. We need to remove all duplicated words at client and server-side.
   5. We check all the value at client-side, to show the total value of the words that can be construct after we place the new letters. It can be more than one words, as it explained on point b.
   6. If the letter is empty, the value will be 0, and cannot be changed
   7. But the empty cue, will be override by another letter, but the value still on 0. It will ask to player when it place on board.
   8. If a word only contains 1 letter, it will be dismissed and will not count as a word.
   9. A word needs to place besides the other old words.
   10. A first word needs to place cross the central of the board.
   11. The new word that cross over the old words will be count as previous statement, but the old letter value that crossed will not be multiplied by DL or TL.
4. At the start of the game, it will make a randomize letters, which have a criterion like this:  
     
   And it will be stored on a stockpile.
5. After check! Button pressed and submitted the letters, if the holder of new letters below than 7, It will take some letters from the stockpile.
   1. After the words left is only 7, we compare the total value of players.
   2. And show the winner.
6. AJAXURL Check word:
   1. Is a server-side programming, which hold the process for checking on python scramble library
   2. It uses python script for the language
   3. The input can be in array or a single letter (which can we see on twl.py)
   4. For the library I download from open source code (can see on twl06.py)
   5. After it was checked as meaningful words, it will be return as a JSON value which contain the words, value, and found.
   6. If all the combined words that submit is true, the program will submit the new word to old words. And will take some other letter to fulfil 7 letters at holder. It is random!
7. Stockpile will show its letter and total of each letter. And will be update each time the stockpile cues are taken.
8. If there is no word can be built, the program will not tell the player. But players are able to swap their letters to others from stockpile. Once again, it is given by random!
9. When there is no input program will show the highlight of the tile that can be filled. It shown as a green dimmed tile. (but still have an error)
10. Before the game start, program will ask the name of players, and show the player name on top. Besides on name it will show the total score that each player got.

**In this game,**

1. It built using latest Laravel 6 (which known as PHP framework).
2. Using small Python Script for English library.
3. Using Angularjs (v1.7) for client-side Javascript.
4. Using Sass pre-processor for converting scss files to css.
5. Using latest Bootstrap.
6. Using Fontawesome Pro v5 (paid license)
7. Using public Github for Submit the project
8. I build with partial loading, for some Javascript other than jQuery for make the loading faster.
9. I change the Fontawesome’s Javascript for svg, and delete some unused symbol. Its change the file-size from 5 MBs to 1 MB. Its help make the better performance. And SVG the icons is compatible to all browser and not to heavy to load when it was compared to image icon.
10. Using Godaddy.com to manage my hosting server. Its shared server on deluxe package. If I compare to Webmaster, Godaddy serve better speed and performance, but the price is up to 3 times than local webservers. In my opinion, Godaddy have a good UI/UX, CPanel server, and hierarchy, as same as Webmaster. (It was my first experience using a CPanel server on Godaddy)
11. Using pythonscript (twl06.py) to check the English words on server-side.

It is far from complete, but the backbone of the game processes all shown and can be played from browser. On my test, I use Google Chrome for browser. If there is some question, of this project, I feel free to be asked anytime and anyplace. I hope the result of is as expected to your project.

**Time of code:**

Saturday : 10 hours,  
Sunday : 10 hours,  
Monday : 3 hours,  
Tuesday : 2 hours,  
Wednesday : 2 hours,  
Thursday : 1 hour (host)  
Friday : 1 hour (host & git)

Total of code time is ±29 hours.   
(outside of creating the documentation)

All the red sections are not code yet.

Best regard,

Indra Saswita