**Name: Sarthak Bhagat**

**Course name: GUI I**

**Instructor- Dr.Wenjin Zhou**

**Assignment#8 - Creating Scrabble Game Using Drag and Drop**

**ReadMe- Implemented Features and their Status**

**File type: .docx file**

***Implemented Features and Their Status***

The purpose of this assignment is to implement a bit of the game of Scrabble using drag and drop. The idea is to display one line of the Scrabble board to the user along with seven letter tiles on a tile rack. The user can drag and drop the tiles to the board to make a word and calculate the player score based on dropped tiles and taking in consideration of bonus squares and keeping track of horizontal and vertical inputs.

The first thing I implemented is the **index.html** file where I did the layout for this assignment. In the body of the html file, **Carousel** was used just to make things look nice and attractive. Carousel of a scrabble board was implemented using **Bootstrap.** Then the tracker for the vertical tiles, horizontal tiles and player score were shown in a horizontal row at the top of the scrabble board just below the carousel. Square tiles were implemented using the rows and columns of a table. Images were put in for the double points/bonus squares at array index 2 and 5. **Scrabble\_Tile\_Blank** implemented at other indexes. At the bottom of the table, 3 buttons were implemented. Restart Game Submit and Replace Word. Class **GameBoard** has all the tiles which are jumbled well because of the **RandomNumber()** function and showed on the screen each time the page is refreshed.

The **scrabble.css** file has all the stylings for the buttons, images, square blocks and most importantly the background of the scrabble board named **scrabble.jpg**

Majority of the work was done in **scrabble.js** where the behavior of the page was shown on how it changed on certain clicks and inputs. The js file started with the pieces of the alphabets which was given by the instructor in the **graphics\_data.zip.** The file had information about the letter pieces, their values and amounts. Vertical array and horizontal array of length of 7 was made.

The function **checkWord()** checks if the word does exist in the array matches the word in dictionary. If it does, the prints the success message. If not, it will tell about does not exist alert message. However, I was not able to implement this feature correctly because my dictionary.js was not read correctly in the function. The checkWord() also uses a function named **IsFound()** which checks if the word exists in dictionary.js. **Dictionary.js** has all the words from the English dictionary was copy pasted from the **/usr/share/dict/words** from the **cs.uml.edu** server.

The function **wordTracker()** keeps track of the word in the vertical and horizontal array.

The hardest part of this assignment was to do the **DragDrop()** The journey UI was used to add the drag and drop functionality for the board . The score board is implemented where the score can be kept track of also it has a feature where the use can see the word he’s trying to make on both the vertical and horizontal line. Snap Tolerance was used widely for drag and drop which checks on which tiles the word is placed and then **WordTracker()** is being used to know the vertical and horizontal inputs. In my opinion, this was the hardest part to implement in this assignment. It was very time consuming and initially confusing.

Another important function is the **scoreTracker()** which keeps track of player points on the scrabble board. At array index 2 and 5 the points are doubled at that specific value of the tile rack. On other tiles, the score is same as the one described on the tile rack. Also, the index of the array connecting horizontal and vertical inputs is scored double. In other words, the tile connecting **the L.**

**Play()** function has an variable named alphabetArray which has 26 letters. I randomly choose 13 tiles which were generated by the function **Math.random()** which choose 13 letters from the 26 letters. **Next\_Search()** was implemented to change the word which the user want to replace. However, it replaces all word, so it is a flaw in this assignment.

**Things Implemented Well:**

Drag and Drop of the Tiles

Scoreboard with multiple features

Scrabble board with at least two double/bonus scoring features

**Restart()** Function

**Things which did not work or could have been handled well:**

**Submit()** feature can’t track words if right or wrong as it is not able to detect words from other js file.

**Replace\_Word()** feature was implemented just to change the specific letter on the board if the user gets stuck. However, it changes all the tiles and kind of works like restart function which was not the purpose of this feature.

**CheckWord()** function does not display the right alert message because it is not able to read words from the dictionary.

**isFound()** function checks if the word is in the English dictionary however it is not able to read/compare the word from the dictionary.js file