Scrabble Word Helper

Nguyen Nation

(Kevin Nguyen, Kevin A. Nguyen, and Ian Nguyen)

Team Members

Project Member	Responsibilities
Kevin Nguyen (Better)	Coding/Documentation
Kevin A. Nguyen (Professor)	Coding/Documentation
Ian Nguyen	Coding

Project Requirements

1. **Work with a small group: no fewer than 2, no more than 3.**

Our group (Nguyen Nation) consists of Kevin Nguyen, Ian Nguyen, and Kevin A. Nguyen.

2. **Pick a project that all group members agree to work on.**

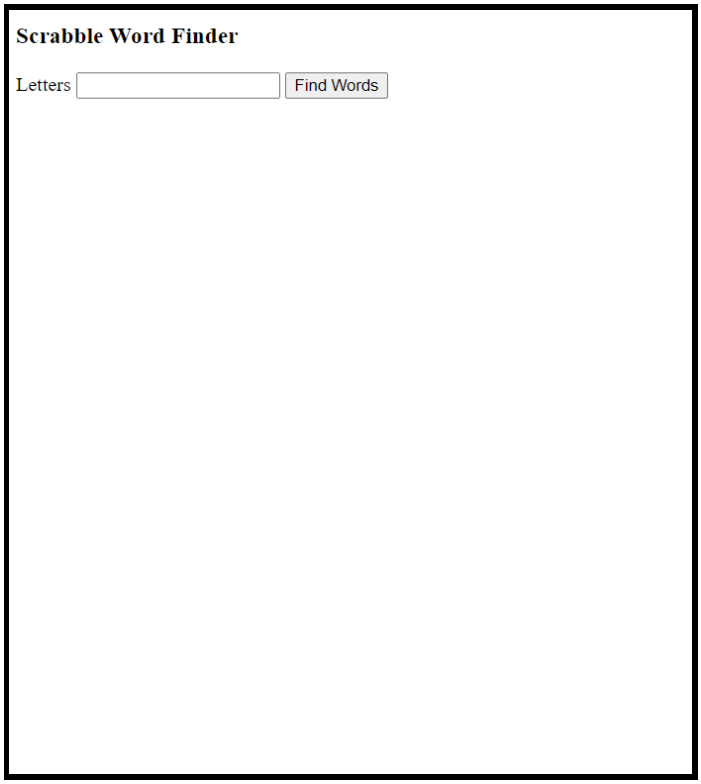
After much debate, we landed on working with a scrabble helper. This scrabble word helper will generate words. Simply type in the letter you have and our software will search nearly 80,000 words and generate you a few words.

3. **Must include existing software, available as open source**

Our project is based on an open source project we found on Github. The original creator is DTing. We used his basic framework of his version of Scrabble Word Finder. DTing project was very useful because it included a database of roughly 80,000 words that we can parse though.

We also used BootStrap. BootStrap is an open source toolkit that helps style your website. It is responsive and has a lot of plugins for JavaScript. We used Bootstrap because it is responsive. This means, no matter the screen size, it will fit onto that screen. In the future, since we are using Bootstrap we can make it a mobile. The grid system on bootstrap will allow us to place things along the grid.

Below is a screenshot of the original Scrabble Word Finder by DTing we based our project off.



4. Must include original code written by your group

```
var letterInput = document.getElementById('letters');
var isDouble = document.getElementById("double"); //2x button
var isTriple = document.getElementById('triple'); //3x button
var foundWords = document.getElementById('words');

var findWords = function() {
  foundWords.innerHTML = ScrabbleWordFinder.find(letterInput.value.toLowerCase(), isDouble.value, isTriple.value).join('\n');
};

```

```
//validwords pushes words into the result array
//Valid words now pushes points into the result array as well.
 var validWords = function(node, letters, word = '', results = []) {
   if (node.isWord) {
     score = points(word);
     results.push('Word: ' + word + ' | ' + 'Points: ' + score);
   }
   var seen = new Set();
   for (let ch of letters) {
     if (!seen.has(ch)) {
       seen.add(ch);
       if (node.children[ch]) {
         validWords(node.children[ch], letters.replace(ch, ''), word + ch, results);
       }
     }
   }
   return results;
 };

 //Calculates the points, takes in a word and if it is using point modifier
 //Replaced the old condition with getting element by id
 //Now multiplies by 2 or 3 or both depending on whether the checkbox is checked or not.
 var points = function(word) {
   let score = ScrabbleWordList[word];
   if(isDouble.checked)
   {
     score *= 2;
   }
   if(isTriple.checked)
   {
     score *= 3;
   }
   return score;
 }
```

```
//The function used to iterate through ScrabbleWordList and destroy my computer
//Basically creates a new object and writes the words + the total point value of it into a text file
//Delete after we get new database
var dict = new Object();
var dict = {}


const fs = require('fs')

//nested for loop
//Doesn't work properly but is unneeded at this point.
//new database has been made so we can delete this code.
for (var i = 0; i < words.length; i++) {
  //Do the math here
  //word connects to the words[i]
  var word = words[i];
  for (var ii = 0; ii < word.length; ii++) {
    var letter = str.charAt(ii);
    //compare to the points
    var points = points[x]
    var score = 0
    score += points[x]
    let data = dict[words[i]] + points;
    fs.writeFile('Output.txt', data);
  }
}
console.log(dict);
```

```
//Letter Scores
//Unused in the overall scheme of code
//Only purpose was to iterate through scrabble Word List and tally up the
//base score
//Delete after it's been used
var ScrabbleLetterScore = {
  "a" : 1,
  "b" : 3,
  "c" : 3,
  "d" : 2,
  "e" : 1,
  "f" : 4,
  "g" : 2,
  "h" : 4,
  "i" : 1,
  "j" : 8,
  "k" : 5,
  "l" : 1,
  "m" : 3,
  "n" : 1,
  "o" : 1,
  "p" : 3,
  "q" : 10,
  "r" : 1,
  "s" : 1,
  "t" : 1,
  "u" : 1,
  "v" : 4,
  "w" : 4,
  "x" : 8,
  "y" : 4,
  "z" : 10
}
```

```
<!-- <div class="page-content"> -->
<div class="jumbotron text-center">
  <h1>Scrabble Word Finder</h1>
  <p>Are you stuck? Enter in your letters and we can help you. </p>
  <div class="wrapper">
    <label for="letters">Letters:</label>
    <input type="text" id="letters"/>

    <input type="checkbox" id="double"/>
    <label for="double">2x</label>

    <input type="checkbox" id="triple"/>
    <label for="triple">3x</label>

    <button onclick="findWords()">Find Words</button>

    <pre id="words"></pre>
    </div>
  </div>
</div>

<script src="lib/ScrabbleWordList.js"></script>
<script src="src/ScrabbleWordFinder.js"></script>
<script src="src/index.js"></script>
</body>

</html>
```

5. Must run on a PC or Mac on Linux or Free BSD

Our project will be hosted locally and can run on PC.

6. Must use Version Control Software

We are using Git as our Version Control Software. Git allows us to collaborate while working on this project. It allows us to store our current in the repository. When one of our team members wants to update the code they have, they just pull from the updated repository stored on Git. When someone wants to push their updated code to the repository, they can easily do that on Git.

7. Must use CASE tools

For CASE tools for our project, Github was our main CASE tool. Github allowed us to have quick development phases as we can push or pull from our repository.

Another CASE tool we used was LucidChart. LucidChart is a real time collaboration tool for diagrams. This helps use diagrammatic and graphical representations of the data and system processes.

We also used Discord to discuss and collaborate ideas. We met on the voice channel and text channels to bounce around recommendations.

8. Improvements compared to original project and issues

From the original database, the database was just all the words. We wanted to improve upon it by adding in the base total points associated to each word. So we created an object called dict which allows for key : values. Create an object that holds the scores associated to each letter in scrabble rules. Iterated through the old word list, separated each word into individual characters, added up points and put the word and points into the dict object and write it to a file.

The text file after it was done was really ugly, so we used https://codebeautify.org/jsviewer to beautify the database instead of trying to do it manually (we're not pressing enter 80000 times).

Because we used a new database we had to change what the original code did and read it off an object instead. We changed the results array to add words and points. The next improvement compared to the original project is multipliers. We couldn't get the letter score multiplier to work so we scrapped that idea, but since scrabble has a double or triple word score multiplier we implemented that into the system.

The second issue was that the system kept multiplying the points by six so after googling some things about javascript and looking at the condition for the code. We added in the getElementById('double') and ('triple). And since javascript has a boolean value for checkboxes we replaced the condition with isDouble.checked/isTriple.checked to prevent the score from being multiplied unless the user clicks on the checkboxes to do so.

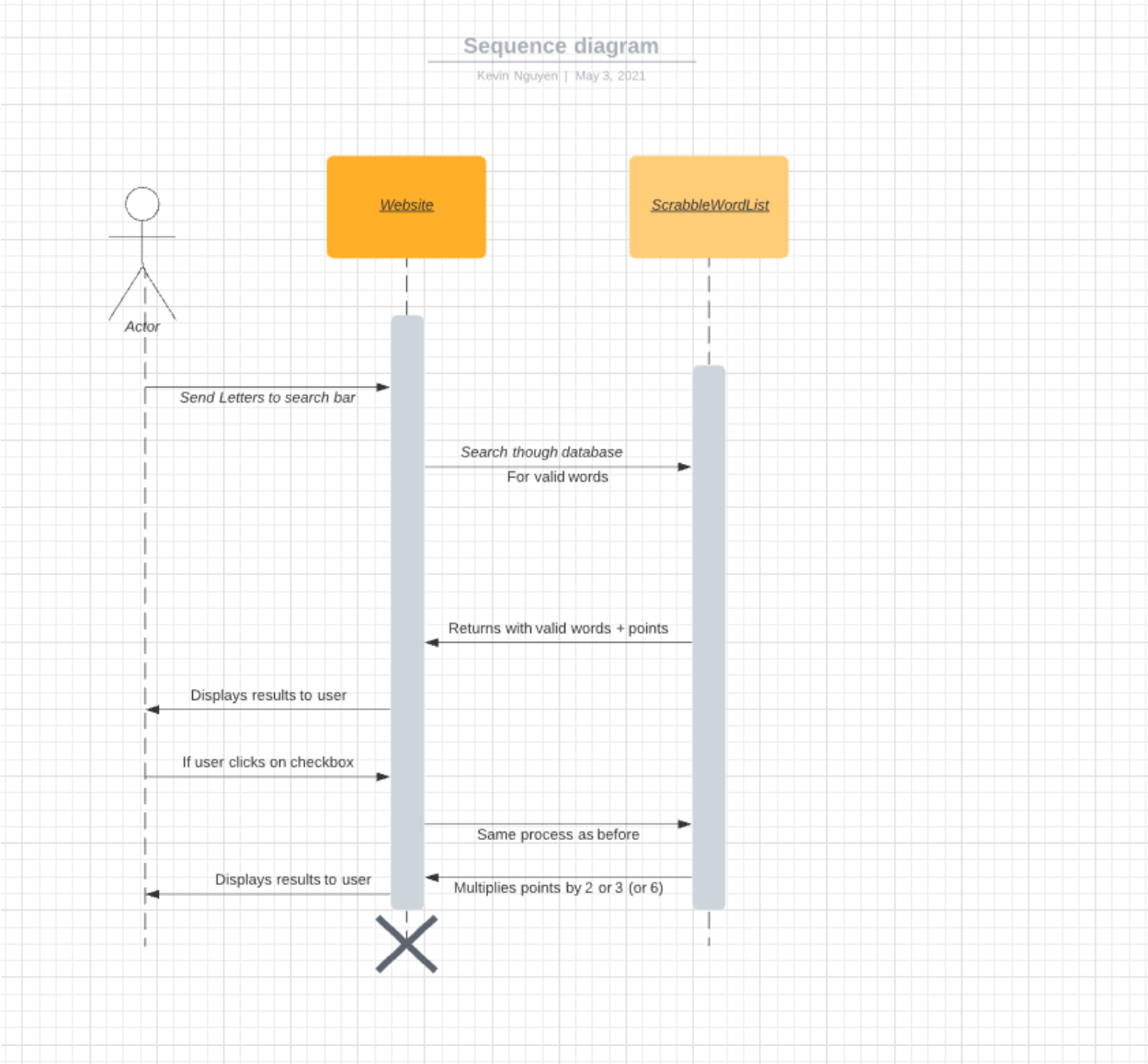
We used bootstrap to improve on the front-end because the front-end of the original project was horrendous.

Results of writing it all to a text file.

```
//Improved on original code by adding points to each word
//Never do this on your computer because it will absolutely murder your processor for half a second
var ScrabbleWordList = {
  "aa": 2,
  "aah": 6,
  "aahed": 9,
  "aahing": 10,
  "aahs": 7,
  "aal": 3,
  "aalii": 5,
  "aaliis": 6,
  "aals": 4,
  "aardvark": 16,
  "aardwolf": 15,
  "aargh": 9,
  "aarrgh": 10,
  "aarrghh": 14,
  "aas": 3,
  "aasvogel": 12,
  "aba": 5,
  "abaca": 9,
  "abacas": 10,
  "abaci": 9,
  "aback": 13,
  "abacus": 10,
  "abacuses": 12,
  "abaft": 10,
  "abaka": 11,
  "abakas": 12,
  "abalone": 9,
  "abalones": 10,
  "abamp": 11,
  "abampere": 14,
  "abamps": 12,
  "abandon": 10,
  "abandons": 11,
  "abapical": 14,
  "abas": 6,
  "abase": 7,
  "abased": 9,
  "abasedly": 14,
  "abaser": 8,
  "abasers": 9,
  "abases": 8,
  "abash": 10,
  "abashed": 13,
  "abashes": 12,
  "abashing": 14,
  "abasia": 8,
  "abasias": 9,
  "abasing": 10,
  "abatable": 12,
```

After beautifying it

Diagrams


Sequence diagram
Kevin Nguyen | May 3, 2021
Website
ScrabbleWordList
Actor
Send Letters to search bar
Search though database
For valid words
Returns with valid words + points
Displays results to user
If user clicks on checkbox
Same process as before
Displays results to user
Multiplies points by 2 or 3 (or 6)