Kenneth Tran –THE LVL: 100 UWU

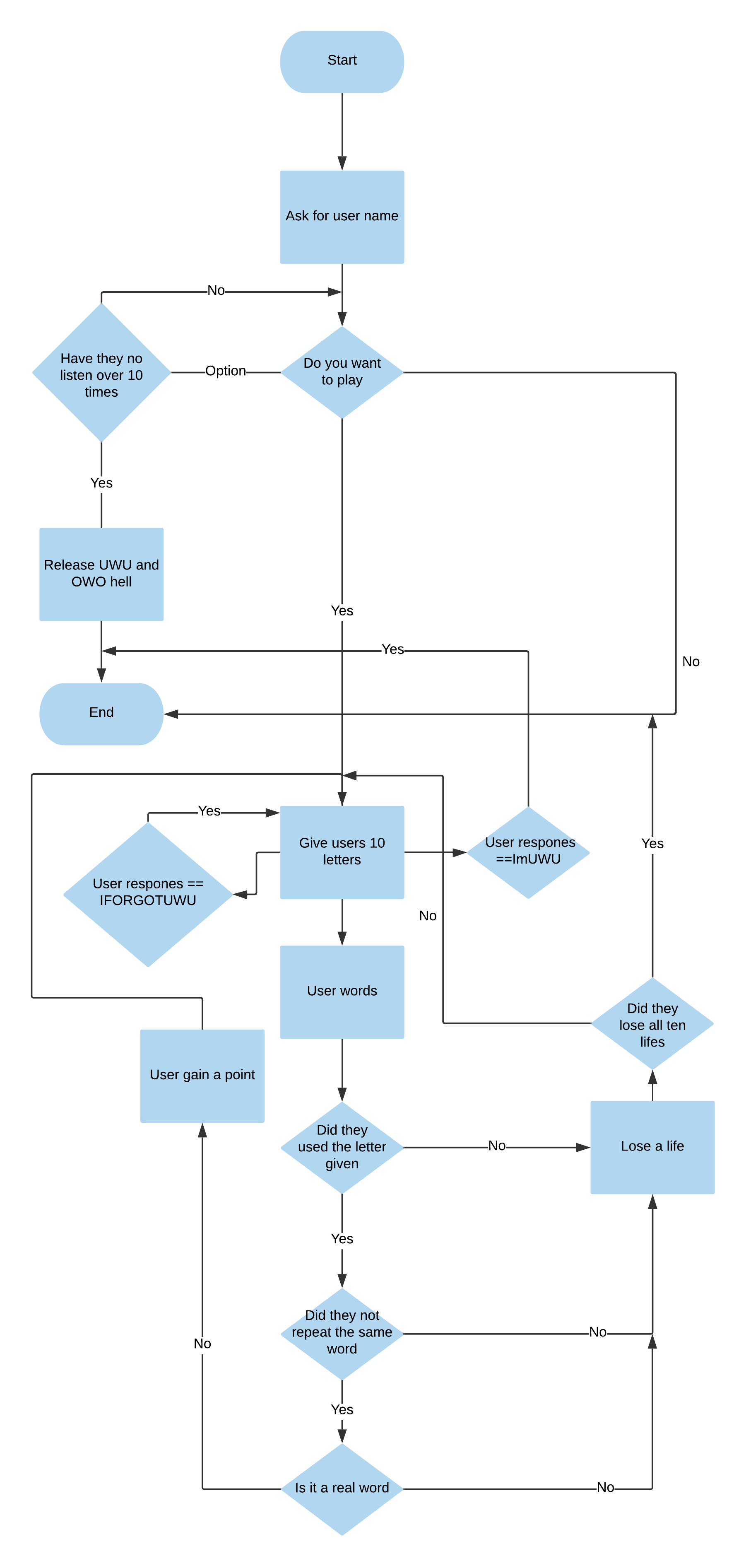
CPSC-39

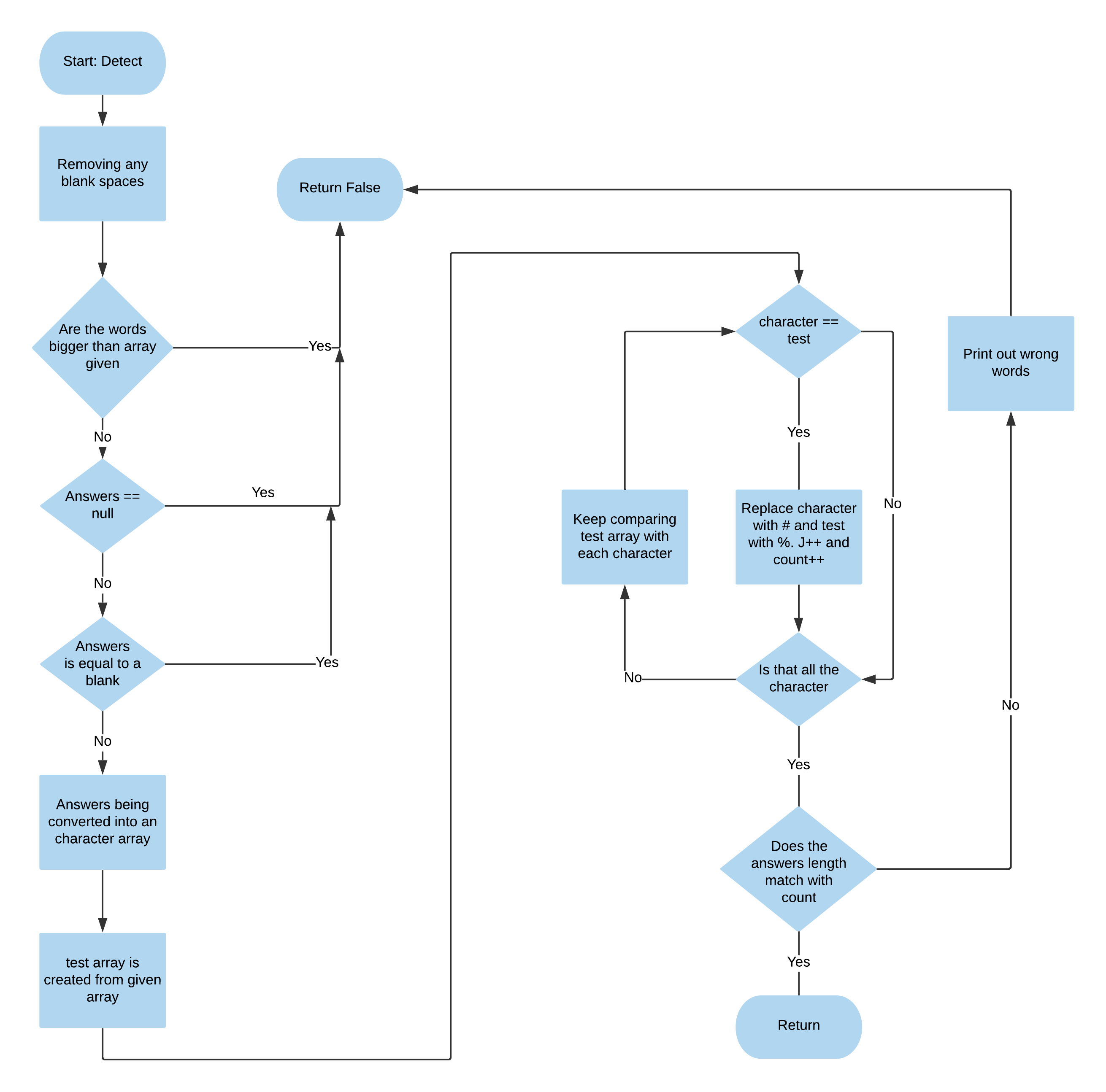
11/25/19

Final Project Report

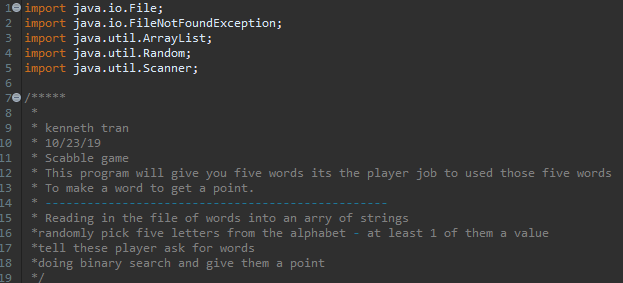
**Scrabble UWU Game**

The program I manifest from the pits of UWU gods. It’s a simple text game where it asks the user name. Sorry, no second chance until you restart the game. The scrabble UWU game will ask you would like to play yes or no. If user hit yes, then they will enter the game. If they hit no, then the game will end. If any other response, then the program will get angry at you. You have 10 tries to listen to the program with Yes or No. Until it spams UWU and OwO at you over a hundred times. The player is given about 10 random letter. They have 10 lives to try to make words out these given letter. The program will detect if you used letters out the array and if you used the same words. This will make you lose a life. The program will keep track of your points you accumulate. This code for this program is mostly original to Professor Kanemoto from my class assignment. But I’ve modified the code to ask user for response and reading input. I add a few new features to the code. Like adjusting the size of letter given, index for user words they input, checking if they are using the letter given. The program will loop the game until the user says he or she is done. Or until they lost all their lives.

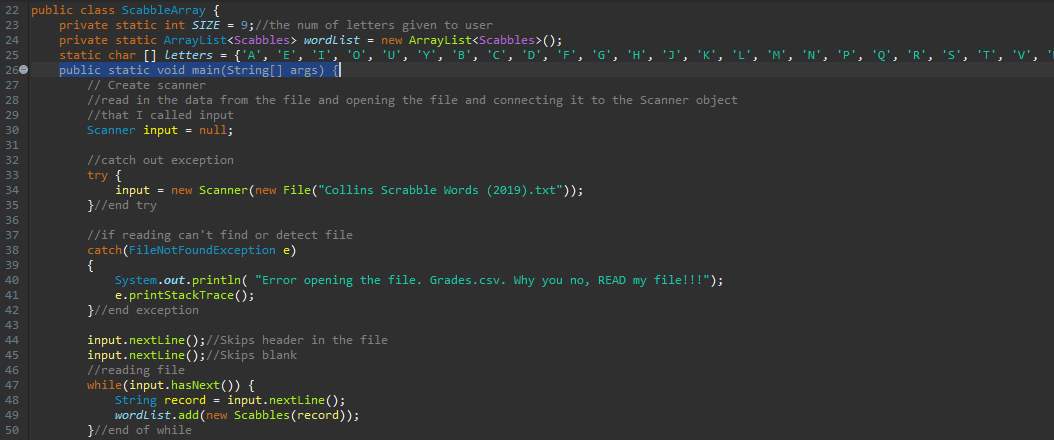




Library and Header

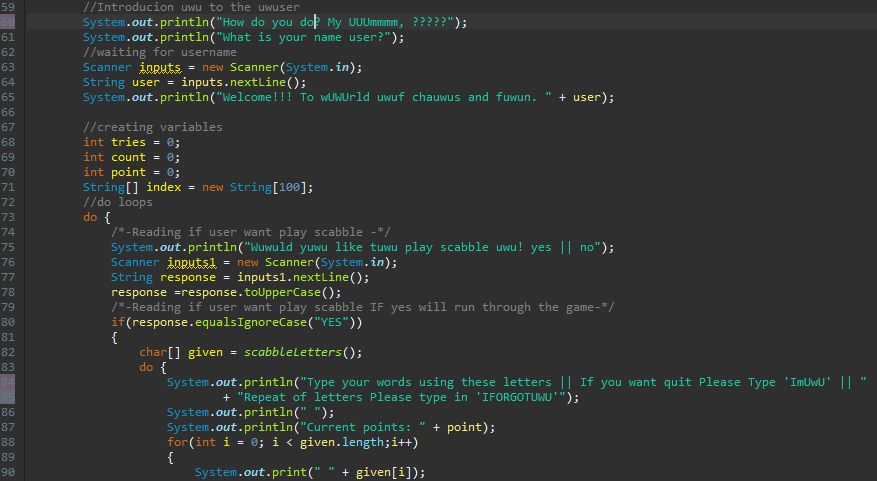


Main – Creating the array and variables && Reading the file for the dictionary of words.

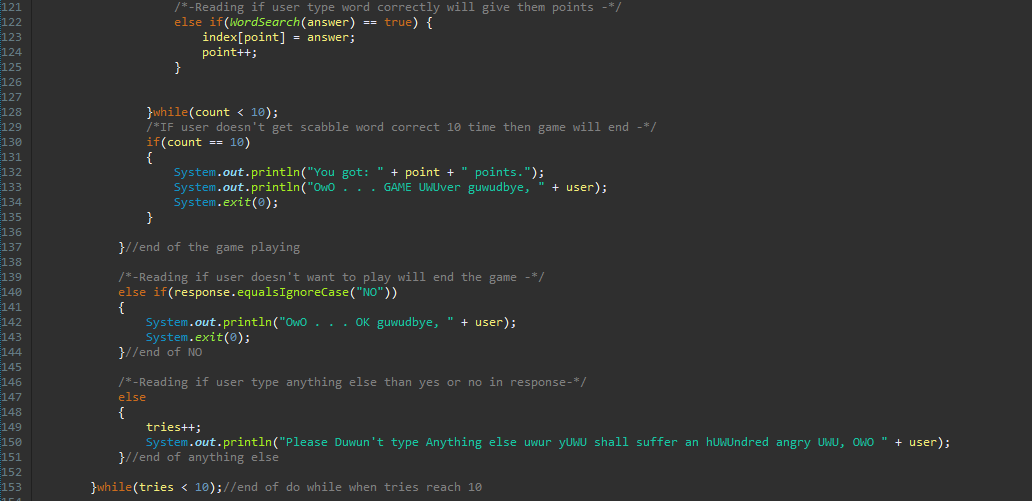


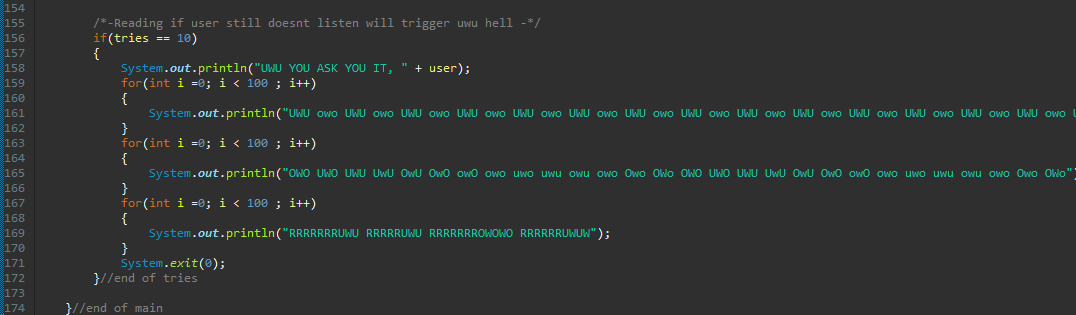
The basic for this programming is setting up the sizes for the number of letters given. Changing the size will make the letters give a N+1 array. The wordList array contains every word in the index to check if the user words are real. The letters array contains every letters in the alphabet. The file “*Collins Scrabble Words*”, will be read and store for the program to used.

Main – User introduction and full game with error detection







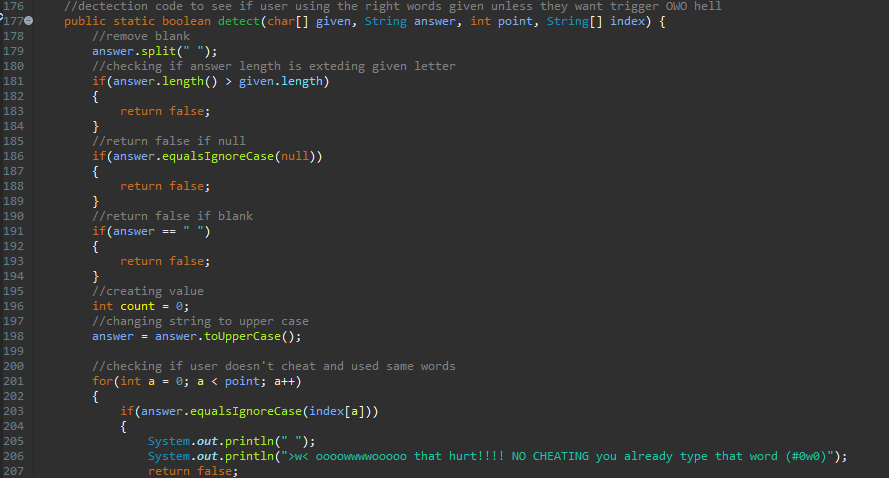


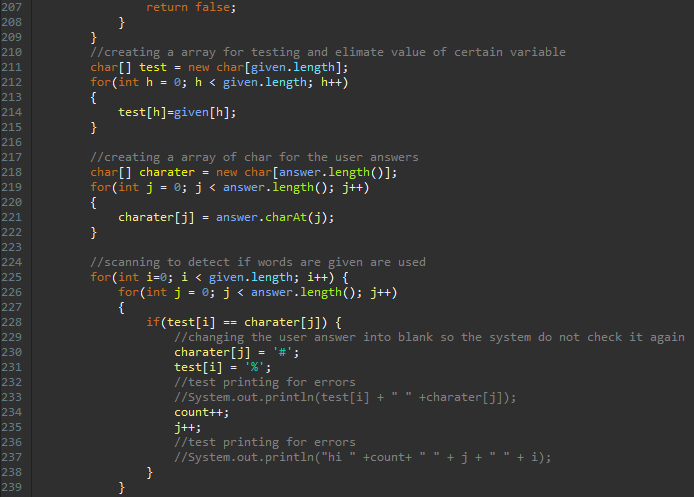
In this part my code will ask the user for his or her name or username. The program will now go deep into response and asking for user input. Like yes or no. If there anything else the code will not like that. And ask you to please read the question carefully. At like 147 of my code your tries will slowly demised. When it reaches 10 tries then the program will yell at you and end the with System.exit(0). Same with if you hit no but less yelling from my code.

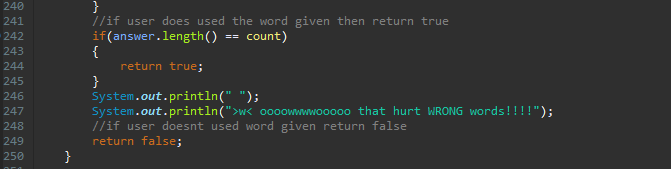
If you decide to hit yes, then the wonderful game will start. With a silly word including an excessive amount of UWU. The game will be on loop until you said “ImUwU” or you lost all 10 your lives in the game. If you hit “IFORGOTUWU” the program will repeat all the words that were given to you.

**Programmer notes -IMPROVEMENTS -:** I would like to add the code can be improved instead to restarting the game over and over. I could add a way for you to play again and shuffle the words if user doesn’t like what they are given. Also add the point system from the Scrabble game. Which every letters are given a certain value. Instead of just counting how many words you made.

Detect Methods-Checking if you send null || blank answers || Words out bounds the letters given. Checking if user hasn’t used the same words and the letters given.





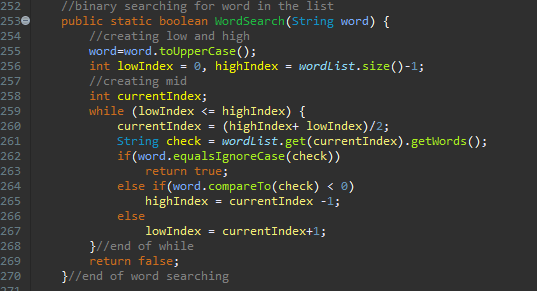


Detect method purpose is to check if user not trying to cheat and user words out of bounds or even used letters that were not given. It would check every word individual. Creating a Big 0 time of N^2 cause it going up and down the array checking every letter of the user word and comparing it to the array of letters. If the letters don’t match it will send a false. If they all matching, then it will send true. This might take a lot of time but doesn’t take that much space. The array is small for holding the letters.

**Programmer notes –IMPROVEMENT-:** The things I would like to improve method. Find another way to check if they are using the letters then forcing the array of letter and the word to change into # or %. I would love to used word replace but I learn that it replaces all the letters in the word and not one. Which lead to error. Another attempt I try to do is switching the for loops but lead to out bounds error. The size might have to fix. If the program is extended bigger with a huge array. Then doing the detect for checking if user didn’t cheat and use letter out of bounds of the array.

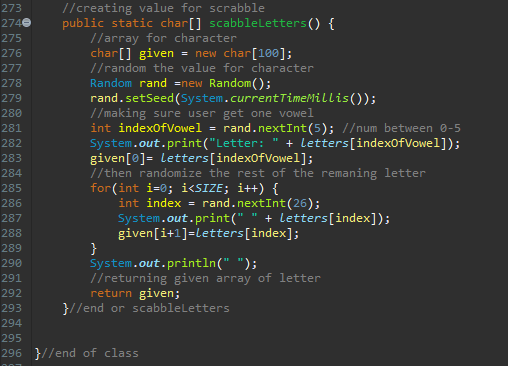
**Programmer notes –CHALLENGE-:** The Detection Method was the hardest method to work on. It was really buggy for me. Trying to make sure every letters are not used more than once. With double or triple made my program read it many times. It was difficult making a method to check the letters. After many trials and error. I came up with an idea to make an array of the letters then used a certain character that will be never used like the # hashtag and % percentage signs. Demonstrated in line 230 and 231.

Binary Search –Checking if user input is a real word in the index



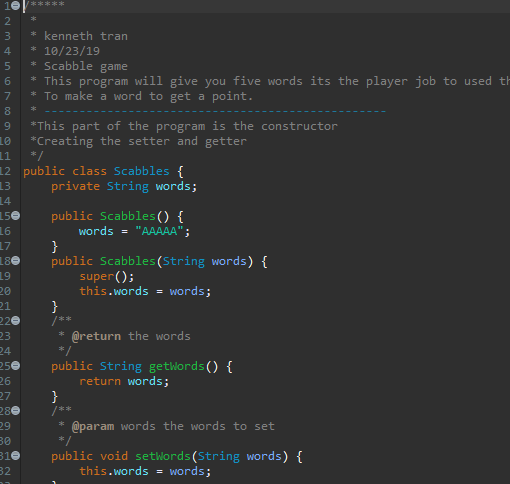
The binary search is the method to check the words that user used is a real word. By checking through the sorted index. The program checks at Big (0) time of Log(N) and space of 0(1). I see no changes in this part my code.

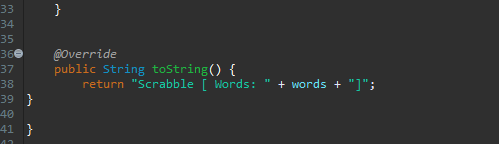
Letter generator – Method where it develops the 10 letters you’ll be using



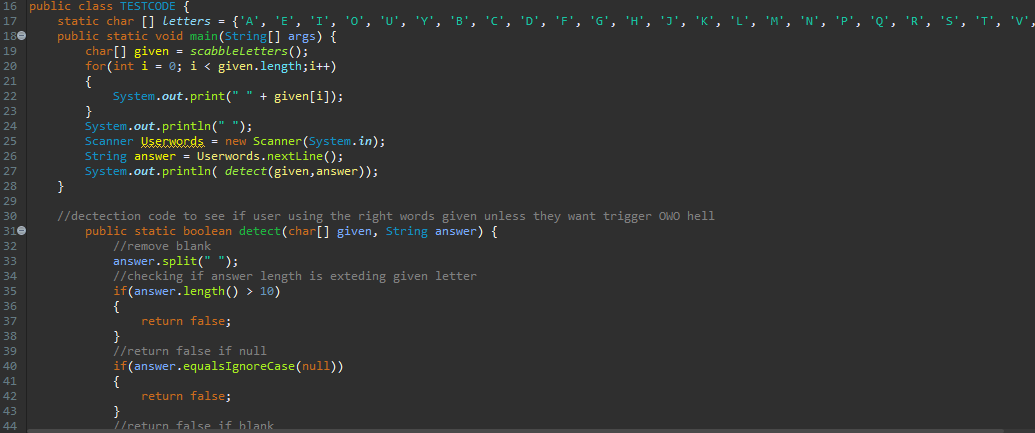
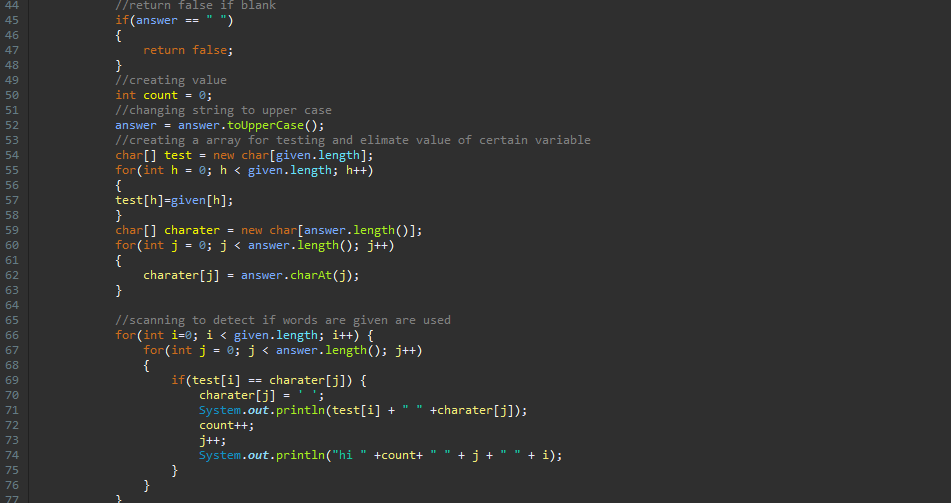
The letter generator is the final method in my program. Generating all random letters except for one. The one letter give will be one hundred percent be a vowel. Thus giving one random vowel and nine random letters.

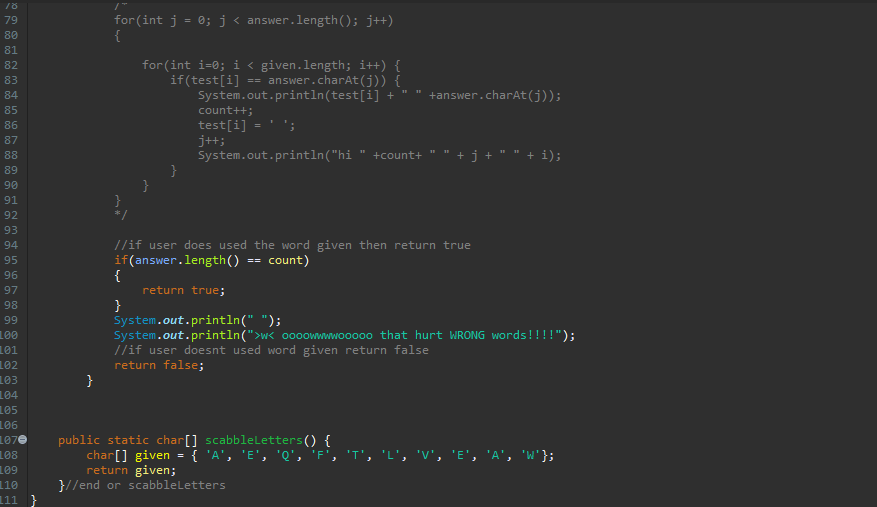
Method for creating constructor, Yes, this program used a constructor only for calling the words from the file.





Test Code



The propose of this other class is to test certain part my code. As I stated before it was my detect method that proven the most difficult. This help me focus on my detect method and try play around with the code until I gotten to work correctly as intended. Also to help me see any error in my code is creating by printing out the result in the loop.

Conclusion

**Algorithms**

The two algorithms that were used. Is in my Detect searching the letter array and ScabbleLetters looking for the word is the same as the list.

Detect-

The method was used to read the user words each one as a char. Using linear searching with two for loops. Then as it goes up each of the user words until it reaches the end. While it going through reading each word it has to go through another for loop where it checks if that char match another. If they are matching the word and letters are replaced in for a special character. The character I used is % and #. So that when it reads it again it will not make a mistake of reading the same thing.

WordSearch-

The method was used to check in a binary. The word that the user created or used. Is now being searched through the list. Thankfully the word in the file is in ABC order or I would have to create another method to organized the data. Using mid to compare the letters and words. If compare is less than zero, then High would be move down the middle minus one. Else than the low would move up middle plus one.

**Big O time**

Detect-

The big O time for my detect like I stated early is 0(n^2). This because of the two for loop checking linearly in my method to check every letter and word. To make sure that the user is not cheating of using the words out of bounds.

WordSearch-

The big O time for my ScabbleLetters is log(n). This because it used binary searching than linear searching. Making the time much more sorter and efficient. Checking every single word linearly would take my program big O time 0(n)

**Data Structures**

wordList: Array list**-** an ArrayList contain all the words in the index contain in the file “*Collins Scrabble Words*”, this is used in the **WordSearch method**

letters: Char array – an array of char contain the alphabet this is used in the **ScabbleLetters method**

tries: int – an integer contains the number of tries before the program yell at the user. Used in the **main method**

count: int– an integer contains the number of lives the user has before it’s game over. Used in the **main method**

points: int– an integer contain the number of points the user accumulated by getting the right word. Used in the **main method**

index: array String – contain an array of the words that user used and spell out. This used to check if they used doesn’t cheat and used the same words to stack up points. Used in the **main method**

character: array char - contain all the user words in an array. The purpose is to hold the user words as an array so I can change each char into a special character. Cause due to my error having trouble with manipulating with the string seems to advance for me. Used in **Detect method**

test: array char– contain all the array of the words in the list. This is a temporary array list. In the method purpose to hold a fake array list. The purpose is to make sure I don’t change the real array list outside the method. Used in **Detect method**

Given: array char- is an array of char containing the user given letters generated randomly in **ScabbleLetters method**

**An Opportunity**

It was an opportunity to work with error checking and user interface. By my programming asking yes and no question. Then using a message when the user type something else. It was fun working with word checking making sure every letters are not used more than one. Or even checking if the used words that were not given to them. Quite a headache trying to make my method work and making sure there were no errors in my code. I learn to used special character to help blank out the array so it doesn’t read it again.

I’ve made my code more humors and fun with when you type something then give. It will spam UwU and OwO at you. Also in the text when asking the user, it will say humors lines. With O or U replace with UWU. It’s quite silly but that the point of my program to make this a fun and interactive game using text.

**What I would change**

* I would like to improve my method and used hash maps to improve my space and time complexity of my code. Maybe improve detect algorithm to make it faster and less space used with so many array I’ve made in the method.
* Make a method for my response for the user instead doing it in main.
* Input a way for user to shuffle the words to given a desirable array instead of starting all over.
* User being able to add new words to the index to make the program more adaptive and user friendly