Fall 2022 INFO6205 Project

Wordle Player

Team:

Kalyan Patnaik – 00296650
Pranav Rajavelu Sivakumar – 002922546
Mital Dudhat – 002983786

Github Link: https://github.com/Kalyan96/INFO6250_Project_Wordle_player/tree/master

Introduction:
Wordle is a word game that has some pretty straightforward rules. The goal is to guess a 5-letter word that is provided by the computer in 6 chances. As the player, we attempt to guess the 5-letter word without hints to start. Through each guess, we are given some pertinent information. In each attempt, as you make a guess, the system will inform you which one of your chosen letters is there in the final target word and whether they are in the correct place. And you keep making guesses to complete the puzzle. If the letter is in its correct position it will be marked as green, if the letter exists in the wrong position then it will be marked yellow and if it does not exist, then it will be marked as gray.

Aim:
Our aim is to create a bot that will choose the best words by using the principle of entropy for each move, hence optimizing the number of chances it takes to guess the correct word.

Steps to implement Wordle:

- Import the set of words from word list and Possible_answers list
    - the word list has the frequency of the words
- The first word for every game is guessed based on the word that provides the highest information of the words in the possible_answers and the word list. Hence as long as the word lists remain the same, the first word that is guessed for every game remains the same
- After we get the information from each step, we reduce the solution space(of possible solutions)

- For every guess we calculate the entropy that is generated using the previous guess, the list of possible answers remaining after every guess, and frequency of the words.
    - Calculation of the Entropy :
        - Entropy is the amount of information provided by each pattern
        - We use the following formula to find the entropy :

$$H(X) = -\sum_{i=1}^{N} P(X = x_i) \log_2 P(X = x_i)$$

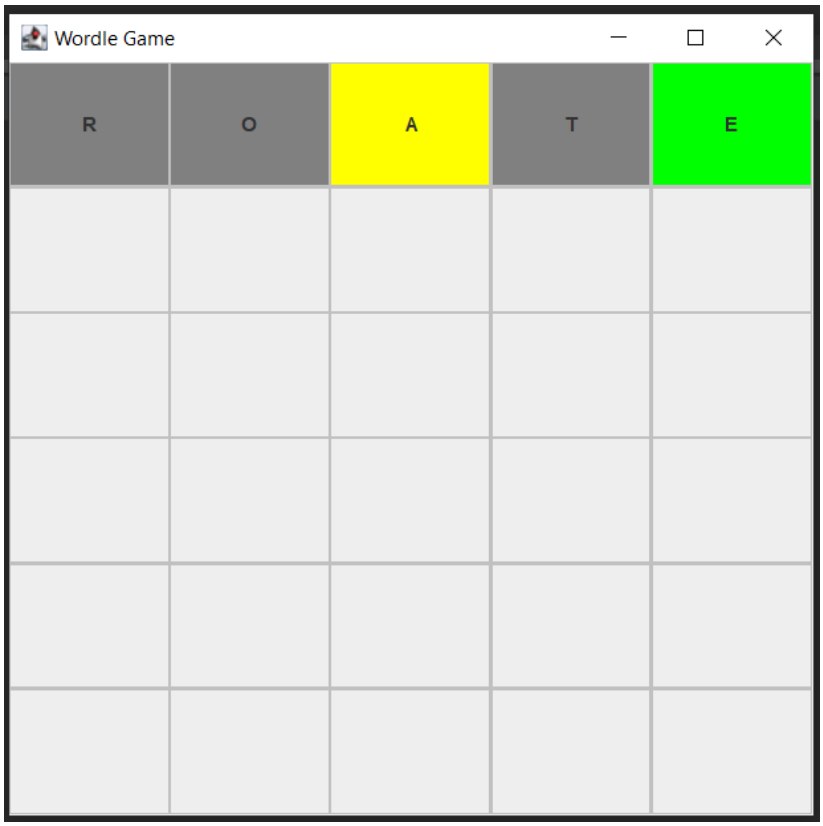
    - Once the entropy provided by each word is known, we will get the word that gives us the maximum information
    - Repeating this eventually reduces the solution space to a small number(usually 2 or 1) and gets us the correct answer

Steps to build and run the project:

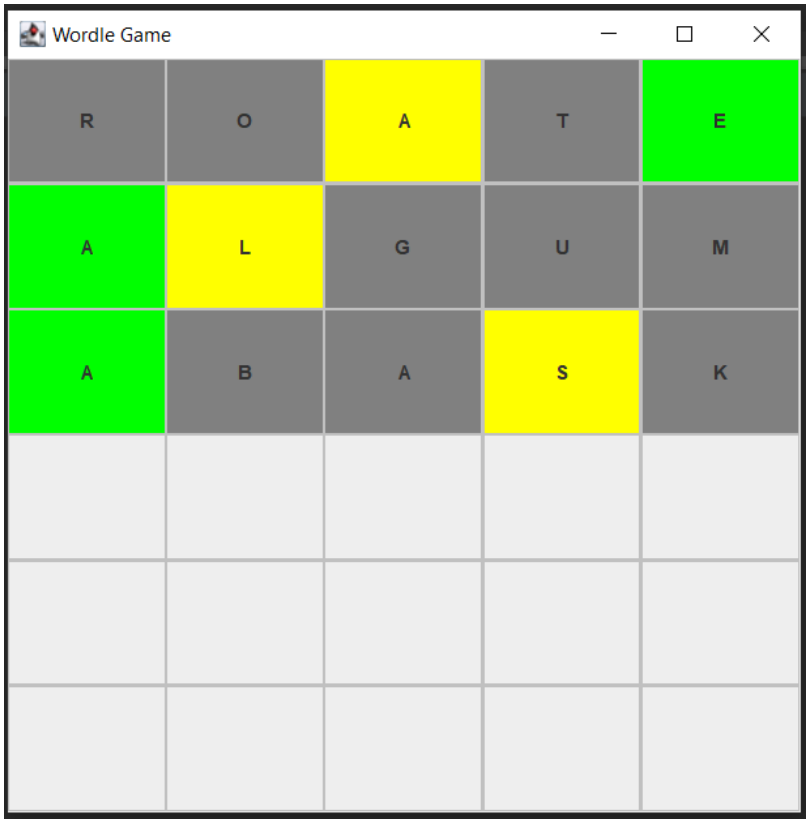
1. Install IntelliJ IDEA
2. Download the project from git (**https://github.com/Kalyan96/INFO6250_Project_Wordle_player/tree/master**)
3. Run the Main.java File.
4. Initially, the program will take approximately 15-30 mins time to run the project to create a pattern matrix and entropy map.
5. Once you start the application UI will pop up on the screen and it will start running in auto mode.
6. To automate test cases, we have provided the results of each run in the terminal.
7. To run unit test cases, go to the testing folder and run the WordleGameTest class.

Working:

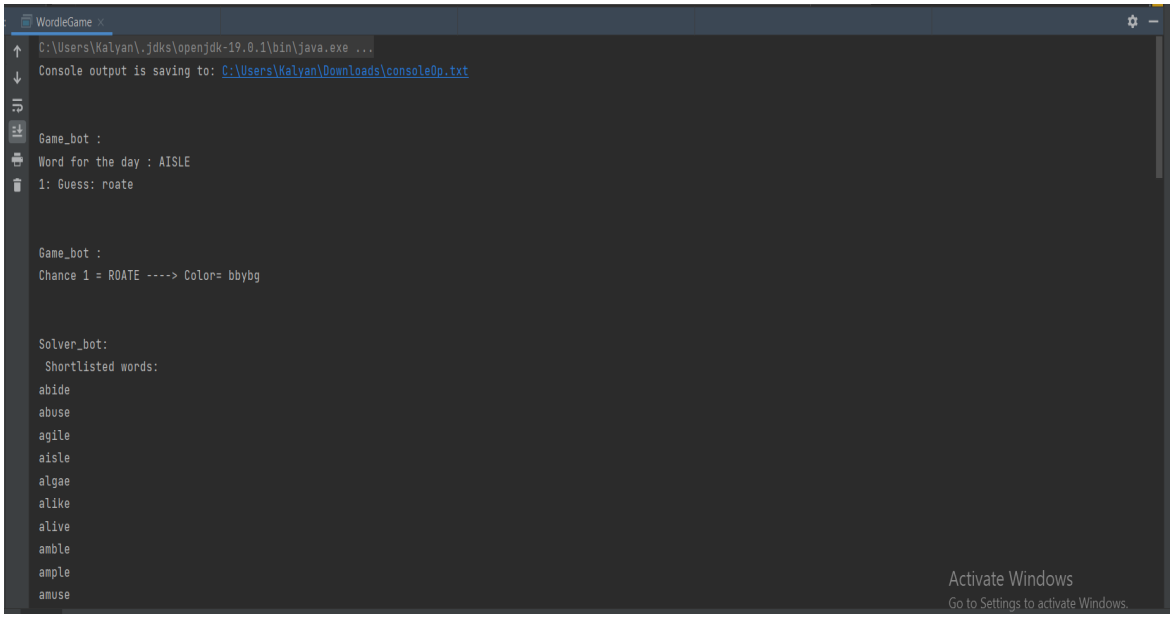
Once you run WordleGame.java, it will display a window with a 5*6 grid and algorithm will automatically select the first 5 letter word based on the entropy. Grey color represents a letter not present in the solution, green color represents the letter in the correct position, yellow represents the letter in incorrect position. Based on the color of all the letters from the previous guesses, the algorithm will generate the next guess using pattern matching and entropy. It will provide the correct answer in 3-4 chances and output will be displayed in the terminal.


Wordle Game
R
O
A
T
E

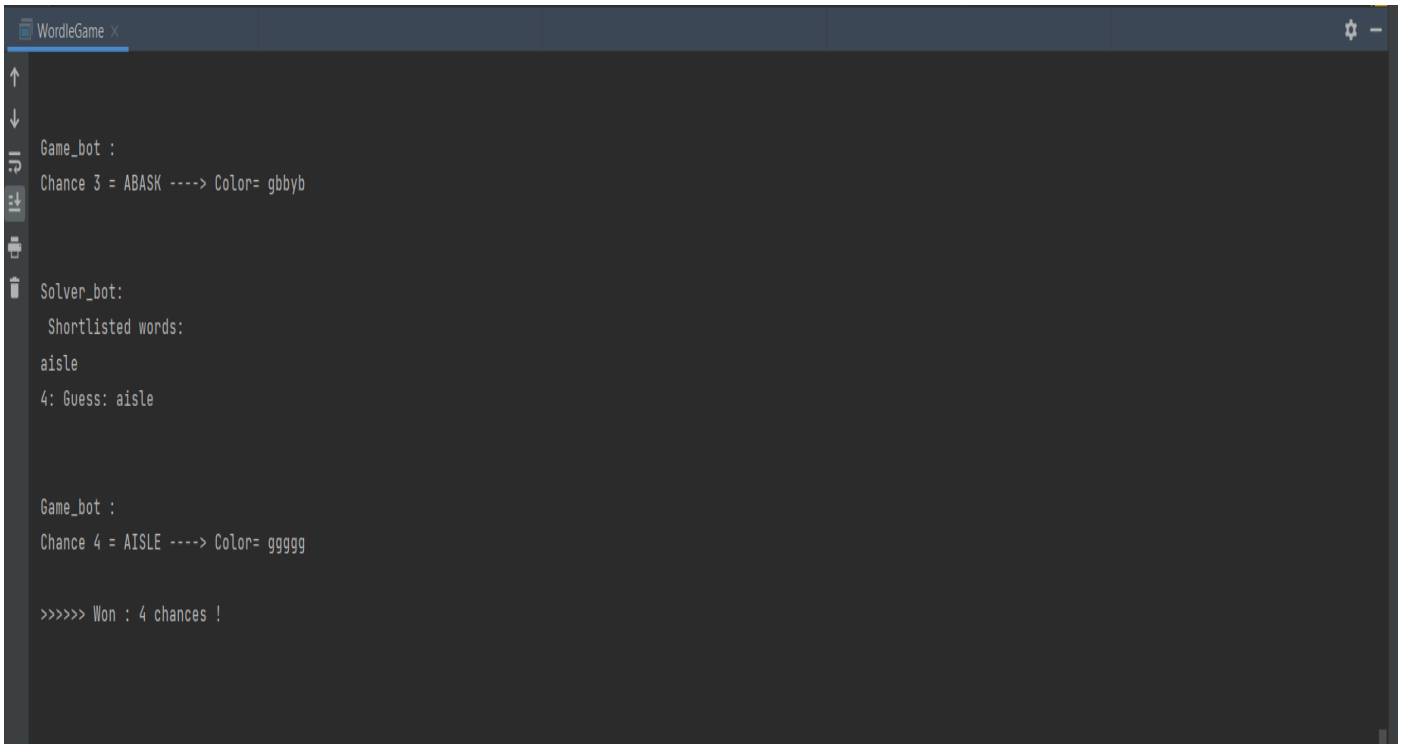

Wordle Game
R O A T E
A L G U M


Wordle Game
R O A T E
A L G U M
A B A S K


Wordle Game
R O A T E
A L G U M
A B A S K
A I S L E
Congrats
You Win in 4chances!!!
OK

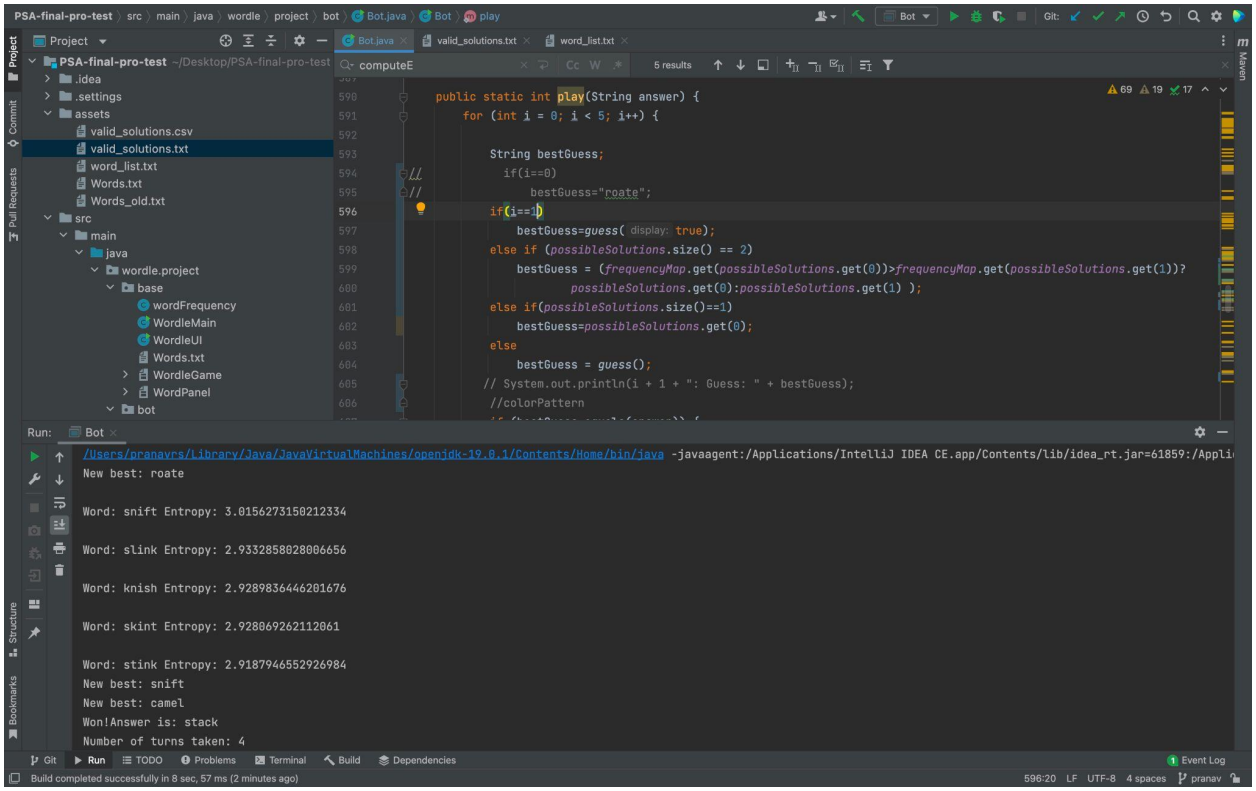

WordleGame
C:\Users\Kalyan\.jdks\openjdk-19.0.1\bin\java.exe ...
Console output is saving to: C:\Users\Kalyan\Downloads\consoleOp.txt
Game_bot :
Word for the day : AISLE
1: Guess: roate
Game_bot :
Chance 1 = ROATE ----> Color= bbybg
Solver_bot:
Shortlisted words:
abide
abuse
agile
aisle
algae
alike
alive
amble
ample
amuse
Activate Windows
Go to Settings to activate Windows.

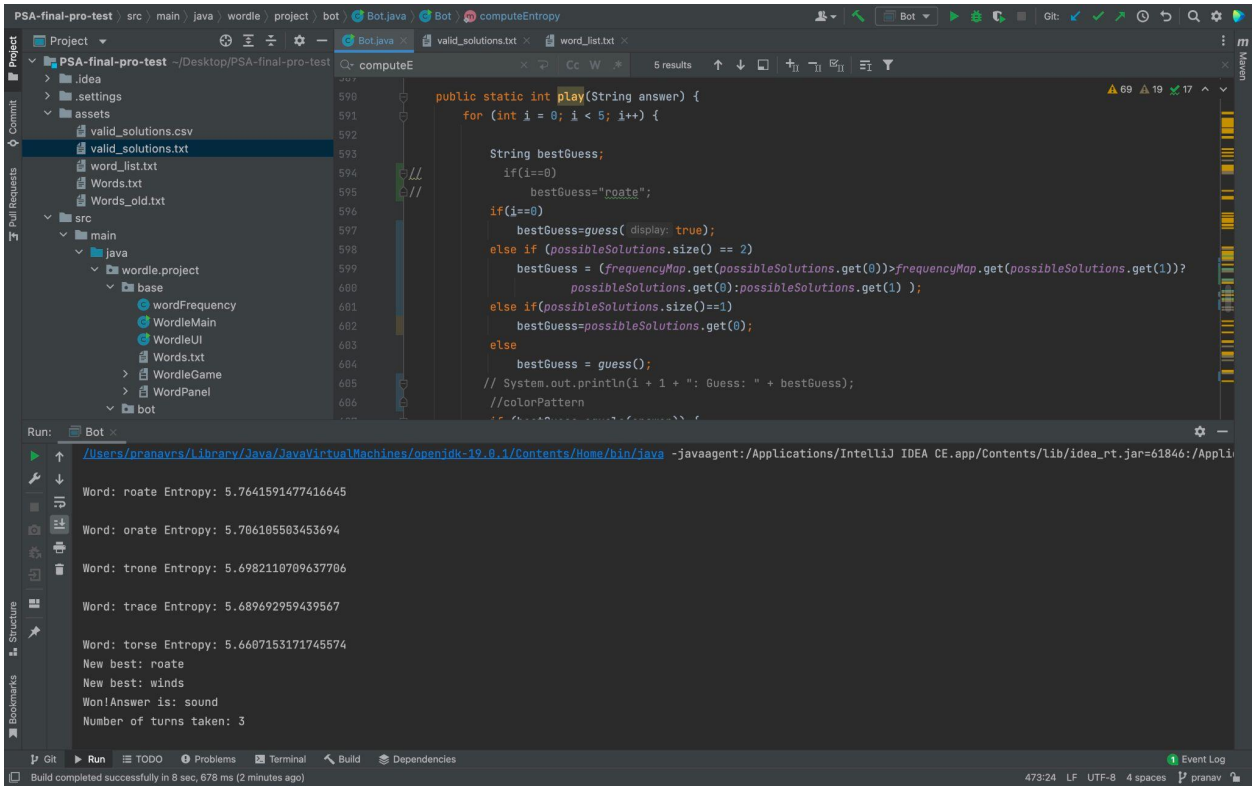

naive
pause
payee
salve
sauce
vague
value
valve
waive
New best: algum
2: Guess: algum
Game_bot :
Chance 2 = ALGUM ----> Color= gybbb
Solver_bot:
Shortlisted words:
aisle
ankle
apple
New best: abask
3: Guess: abask
Activate Windows
Go to Settings to activate Windows.

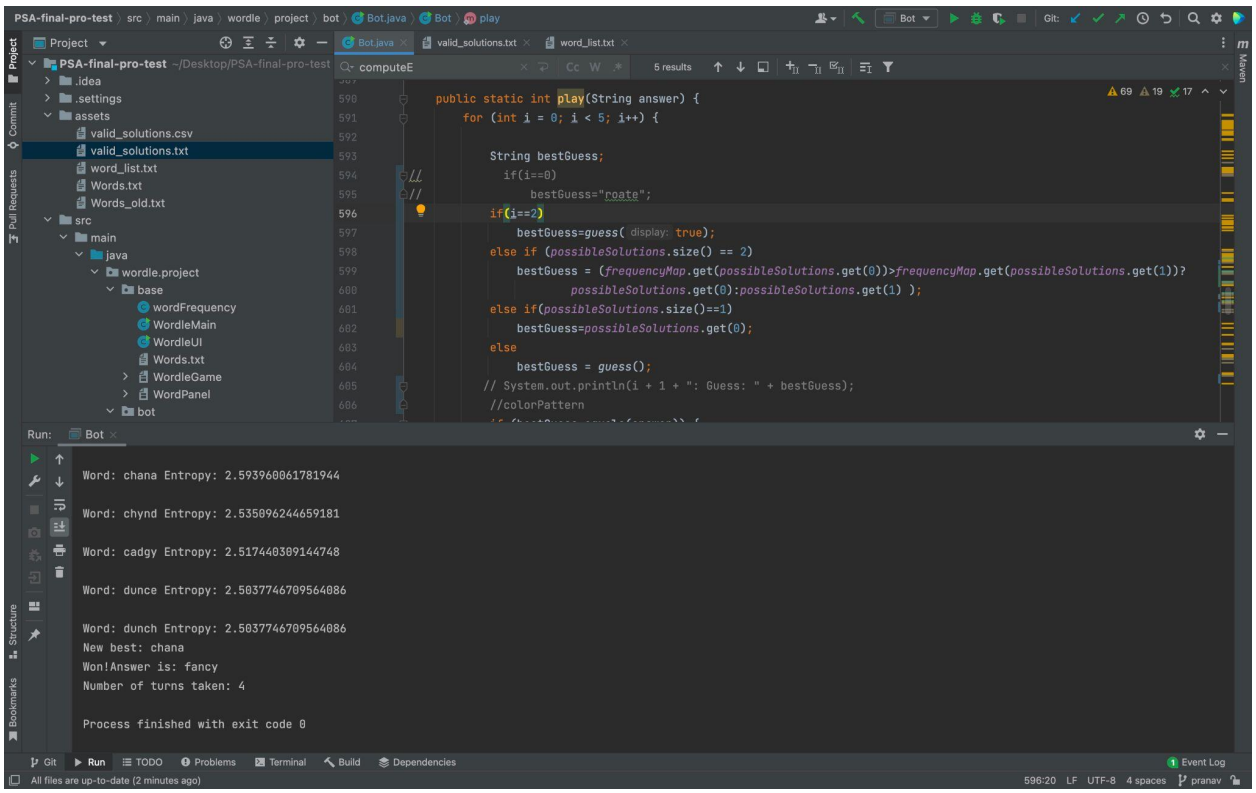

WordleGame
Game_bot :
Chance 3 = ABASK ----> Color= gbbyb
Solver_bot:
Shortlisted words:
aisle
4: Guess: aisle
Game_bot :
Chance 4 = AISLE ----> Color= ggggg
>>>>>> Won : 4 chances !

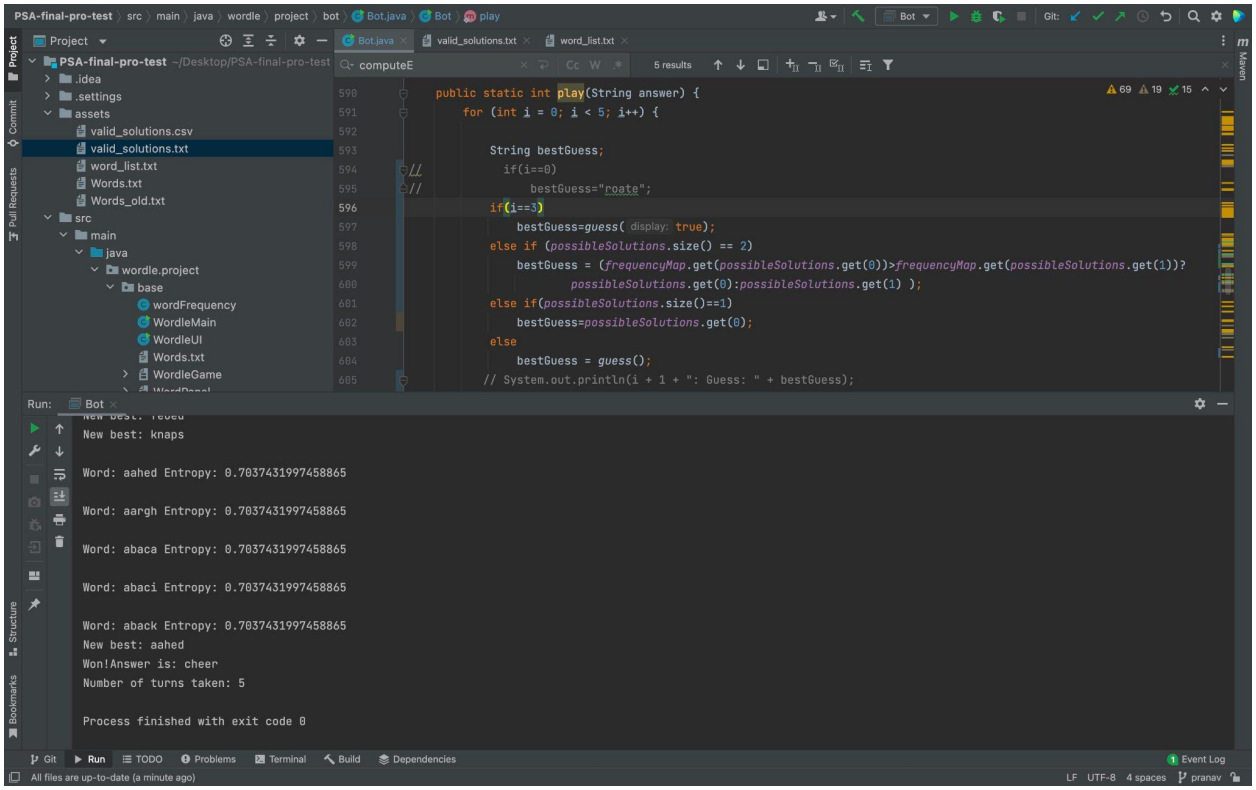
Result:

Screenshots for the result of entropy of a word after each guess.

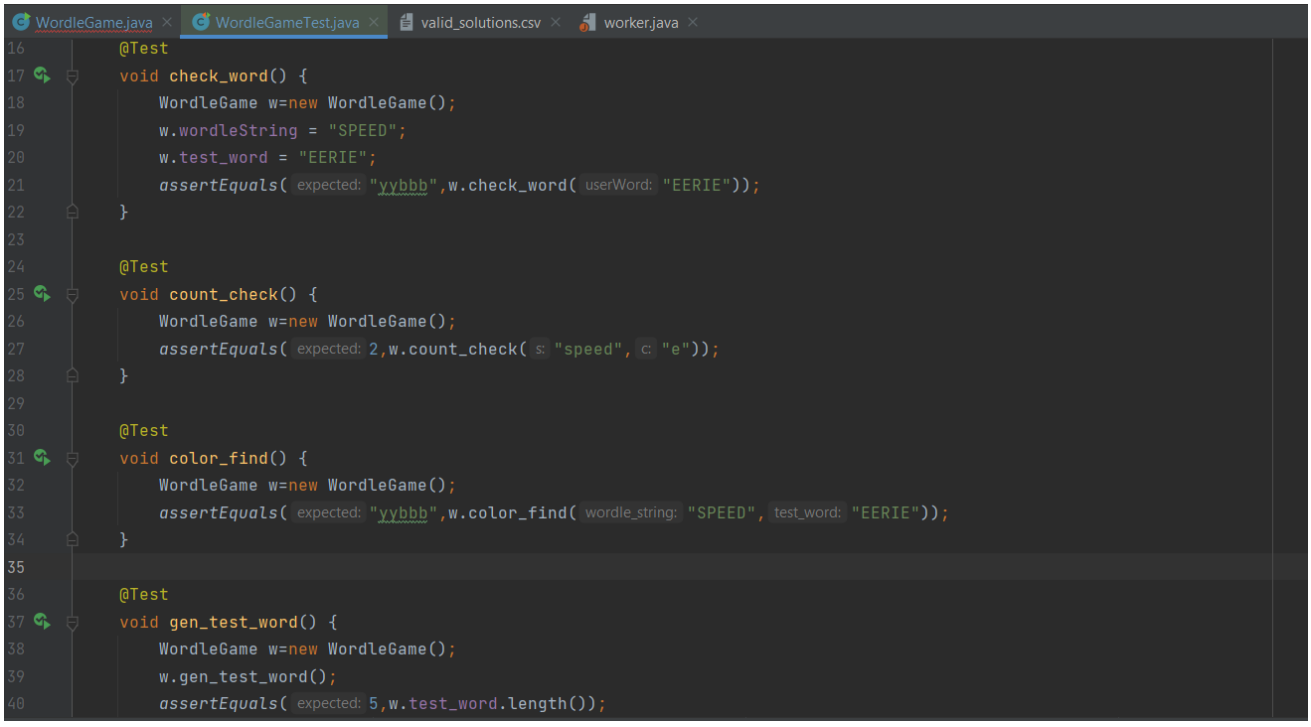


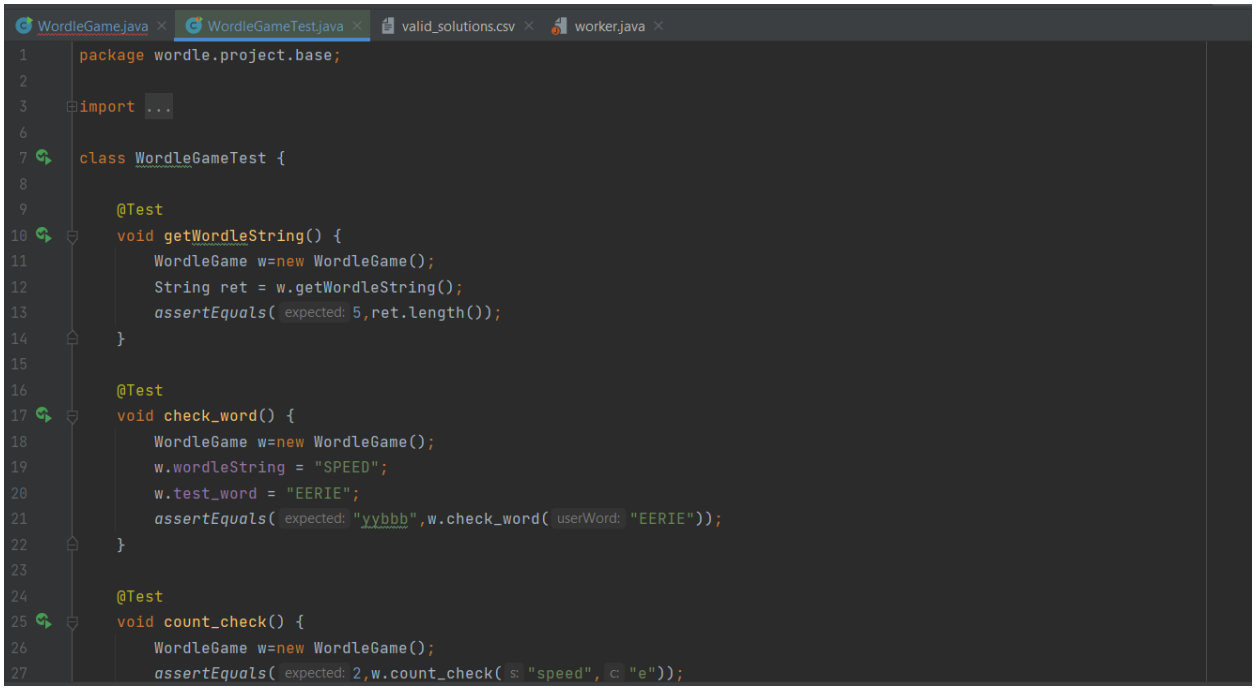

PSA-final-pro-test › src › main › java › wordle › project › bot › Bot.java › Bot › computeEntropy
Project
PSA-final-pro-test ~/Desktop/PSA-final-pro-test
.idea
.settings
assets
valid_solutions.csv
valid_solutions.txt
word_list.txt
Words.txt
Words_old.txt
src
main
java
wordle.project
base
wordFrequency
WordleMain
WordleUI
Words.txt
WordleGame
WordPanel
bot
Bot.java
valid_solutions.txt
word_list.txt
computeE
5 results
public static int play(String answer) {
for (int i = 0; i < 5; i++) {
String bestGuess;
if(i==0)
bestGuess="roate";
if(i==0)
bestGuess=guess(display: true);
else if (possibleSolutions.size() == 2)
bestGuess = (frequencyMap.get(possibleSolutions.get(0))>frequencyMap.get(possibleSolutions.get(1))?
possibleSolutions.get(0):possibleSolutions.get(1));
else if(possibleSolutions.size()==1)
bestGuess=possibleSolutions.get(0);
else
bestGuess = guess();
// System.out.println(i + 1 + ": Guess: " + bestGuess);
//colorPattern
Run: Bot
/Users/pranavcs/Library/Java/JavaVirtualMachines/openjdk-19.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=61846:/Appli
Word: roate Entropy: 5.7641591477416645
Word: orate Entropy: 5.706105503453694
Word: trone Entropy: 5.6982110709637706
Word: trace Entropy: 5.689692959439567
Word: torse Entropy: 5.6607153171745574
New best: roate
New best: winds
Won!Answer is: sound
Number of turns taken: 3
Git
Run
TODO
Problems
Terminal
Build
Dependencies
Event Log
Build completed successfully in 8 sec, 678 ms (2 minutes ago)
473:24 LF UTF-8 4 spaces pranav


PSA-final-pro-test › src › main › java › wordle › project › bot › Bot.java › Bot › play
Project
PSA-final-pro-test ~/Desktop/PSA-final-pro-test
.idea
.settings
assets
valid_solutions.csv
valid_solutions.txt
word_list.txt
Words.txt
Words_old.txt
src
main
java
wordle.project
base
wordFrequency
WordleMain
WordleUI
Words.txt
WordleGame
WordPanel
bot
Bot.java
valid_solutions.txt
word_list.txt
computeE
5 results
public static int play(String answer) {
for (int i = 0; i < 5; i++) {
String bestGuess;
if(i==0)
bestGuess="roate";
if(i==2)
bestGuess=guess(display: true);
else if (possibleSolutions.size() == 2)
bestGuess = (frequencyMap.get(possibleSolutions.get(0))>frequencyMap.get(possibleSolutions.get(1))?
possibleSolutions.get(0):possibleSolutions.get(1));
else if(possibleSolutions.size()==1)
bestGuess=possibleSolutions.get(0);
else
bestGuess = guess();
// System.out.println(i + 1 + ": Guess: " + bestGuess);
//colorPattern
Run: Bot
Word: chana Entropy: 2.593960061781944
Word: chynd Entropy: 2.535096244659181
Word: cadgy Entropy: 2.517440389144748
Word: dunce Entropy: 2.5037746709564086
Word: dunch Entropy: 2.5037746709564086
New best: chana
Won!Answer is: fancy
Number of turns taken: 4
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Build
Dependencies
Event Log
All files are up-to-date (2 minutes ago)
596:20 LF UTF-8 4 spaces pranav


PSA-final-pro-test
Project
.idea
.settings
assets
valid_solutions.csv
valid_solutions.txt
word_list.txt
Words.txt
Words_old.txt
src
main
java
wordle.project
base
wordFrequency
WordleMain
WordleUI
Words.txt
WordleGame
Bot.java
valid_solutions.txt
word_list.txt
computeE
5 results
public static int play(String answer) {
for (int i = 0; i < 5; i++) {
String bestGuess;
if(i==0)
bestGuess="roate";
if(i==3)
bestGuess=guess(display: true);
else if (possibleSolutions.size() == 2)
bestGuess = (frequencyMap.get(possibleSolutions.get(0))>frequencyMap.get(possibleSolutions.get(1))?
possibleSolutions.get(0):possibleSolutions.get(1));
else if(possibleSolutions.size()==1)
bestGuess=possibleSolutions.get(0);
else
bestGuess = guess();
// System.out.println(i + 1 + ": Guess: " + bestGuess);
Run:
Bot
New best: knaps
Word: aahed Entropy: 0.7037431997458865
Word: aargh Entropy: 0.7037431997458865
Word: abaca Entropy: 0.7037431997458865
Word: abaci Entropy: 0.7037431997458865
Word: aback Entropy: 0.7037431997458865
New best: aahed
Won!Answer is: cheer
Number of turns taken: 5
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Build
Dependencies
Event Log
All files are up-to-date (a minute ago)
LF
UTF-8
4 spaces
pranav

Unit Tests:


WordleGame.java
WordleGameTest.java
valid_solutions.csv
worker.java
@Test
void check_word() {
WordleGame w=new WordleGame();
w.wordleString = "SPEED";
w.test_word = "EERIE";
assertEquals(expected: "yybbb",w.check_word(userWord: "EERIE"));
}
@Test
void count_check() {
WordleGame w=new WordleGame();
assertEquals(expected: 2,w.count_check(s: "speed", c: "e"));
}
@Test
void color_find() {
WordleGame w=new WordleGame();
assertEquals(expected: "yybbb",w.color_find(wordle_string: "SPEED", test_word: "EERIE"));
}
@Test
void gen_test_word() {
WordleGame w=new WordleGame();
w.gen_test_word();
assertEquals(expected: 5,w.test_word.length());



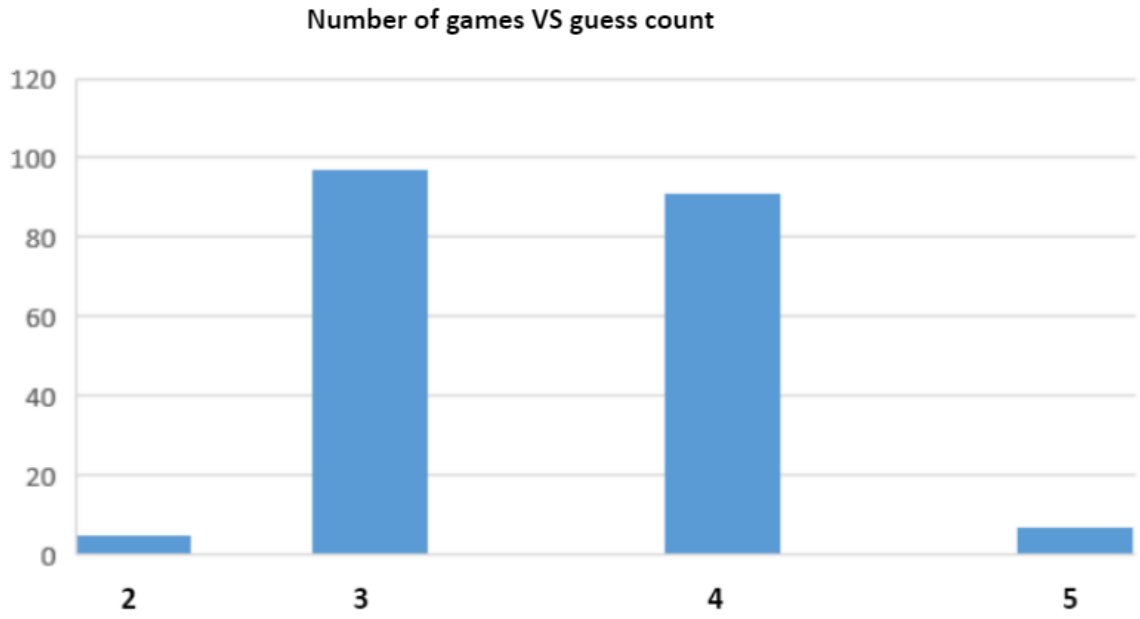
Analysis :

To calculate the time complexity for a word's entropy we define n as the words in the word list and k as the number of letters in each word. Each letter in a pattern has one of three possible statuses. In total, there are 3^k distinct patterns.
In order to get the color pattern for a given answer and guess, the program runs in linear time for the best case and quadratic time in the worst case. So time complexity of obtaining the entropy of a word is O(3^k * k^2 * n).

However, this complexity decreases greatly because we have the word,"roate", as the first guess. The average information we gain from this word is 5.76 bits, which leaves us with 7.9 bits of uncertainty. This is equivalent to shortening our possible solutions list to 239 words with equal frequency of occurrence.

Conclusion:

The below image provides the average number of attempts made by the algorithm to find the correct answer.



References:

https://github.com/3b1b/videos/blob/master/_2022/wordle/data/freq_map.json