# WORDLE PLAYER PROJECT

Bohan Feng (NUID 001564249)

Olasunkanmi Olayinka (NUID 001512266)

https://github.com/fengb3/INFO6205\_final\_project

## Table of Contents

* Introduction about the topic
* Aim of the project
* Steps to implement solution
* Build steps and requirements
* Complete project details and its functionalities
* Application images
* Analysis
* Unit test
* Conclusion

### **Introduction about the topic:**

Wordle is web-base word game in which the goal is to guess a five-letter word, and you are given 6 different chances to guess the word, each time you make a guess you get some information about how close the word you guessed is close to the actual word. The letters are color coded; grey means the letter doesn’t exist in the word, yellow mean the letter exist but it’s in the wrong position and green means the letter exist and is in the right position.

This project introduces a software that takes the word guess with each letter’s color code and calculate the next best possible guess word using information theory.

### **Aim of the project:**

Create a wordle player that will choose the best words for any given move, according to the principles of entropy, here by optimizing the number of chances it takes you to guess the correct word

### **Steps to implement solution:**

* Compare each pair of words to get a pattern matrix  
  We use a look up for every time we compare 2 words to speed up the process
* Calculate the expected amount of information we can get from each word in a word list
* Get the word frequency to know how likely a word is to be a common word
* Use two steps to get best beginner words

### **Build steps and requirements:**

You will need JDK 11 or greater and IntelliJ idea for build the application

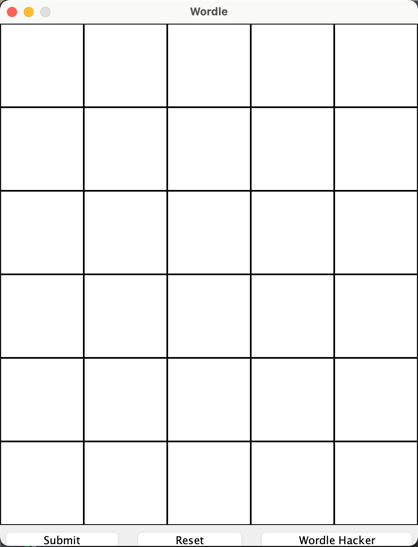
1. Download and unzip project into directory of choice
2. Setup project SDK in IntelliJ with Java 11 or greater
3. Run the Main.java file   
   Note: this will take time (15 – 40 mins) running the first time to create entropy\_map and pattern\_matrix, to reduce time you can add files [entropy\_map.txt and pattern\_matrix.txt](https://drive.google.com/drive/folders/1ffIKy1CKfYC06kVAohdaFUFc2TniVG4j?usp=sharing) (this couldn’t me pushed to GitHub because of file size limitation) to the data directory

### **Complete project details and its functionalities:**

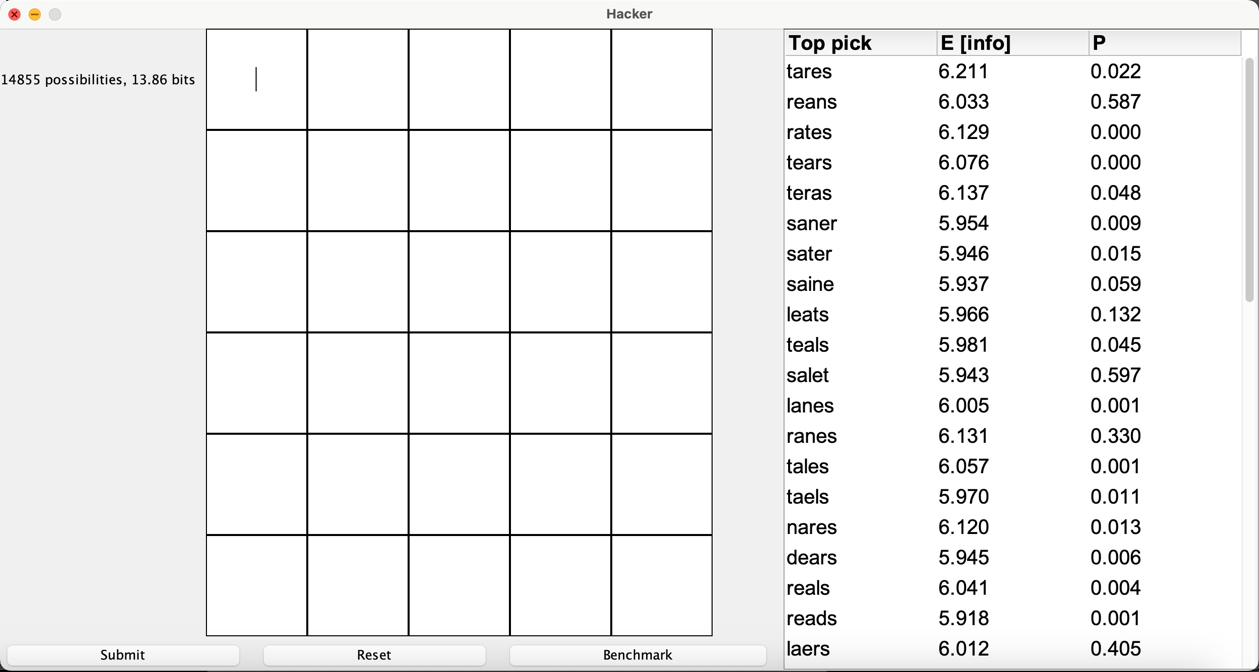
* Start the application by running Main.java  
  The Wordle application UI should pop up (you can play regular wordle with this)
* Click on Wordle Hacker button to open the Wordle Player (Hacker) UI  
  The Hacker UI should pop up showing Top pick, Entropy, and the Probability
* Play the Wordle application and the Hacker UI is auto populated showing next best possible words sorted with respect to probability.
* You can manually populate the Hacker app with your guess words and select the appropriate color, if you are playing wordle on a different platform or application
* You can run a Benchmark by clicking the benchmark button  
  This will run a test on 2309 guess words and returns the average number of chances it takes the Wordle Hacker to get the guess word. It logs how many tries it takes for each guess word on the console and writes the average number of chances into a file data/benchmark\_$dateTime.txt

### **Application Images:**

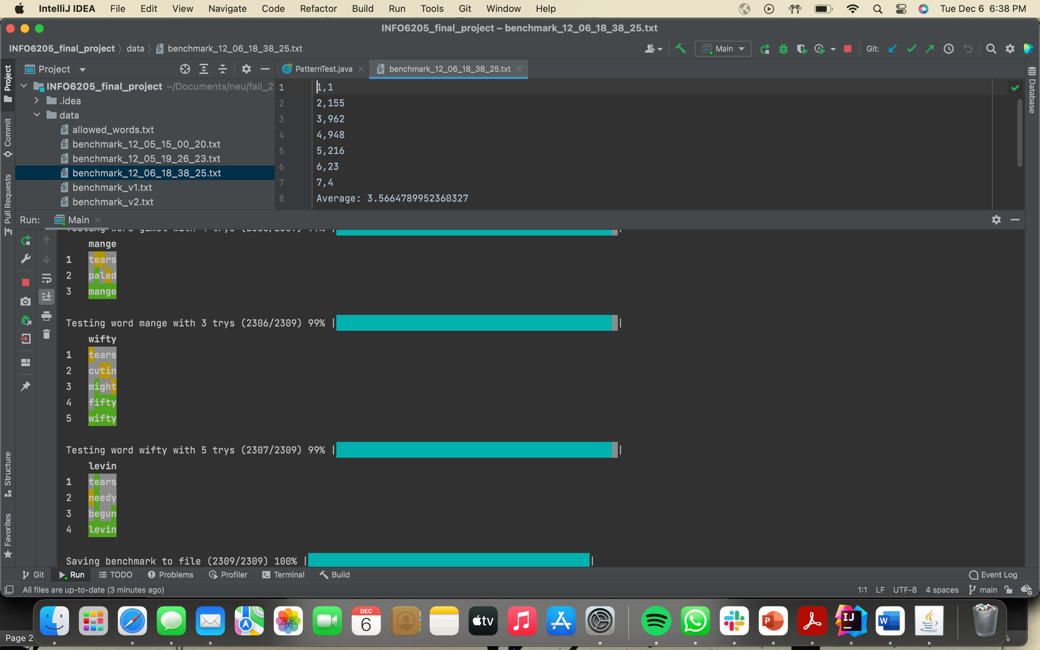
Wordles app:



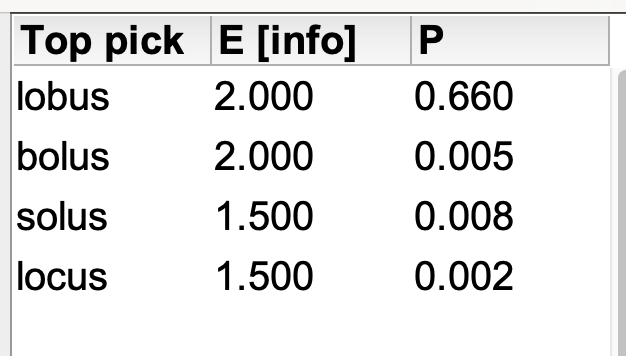
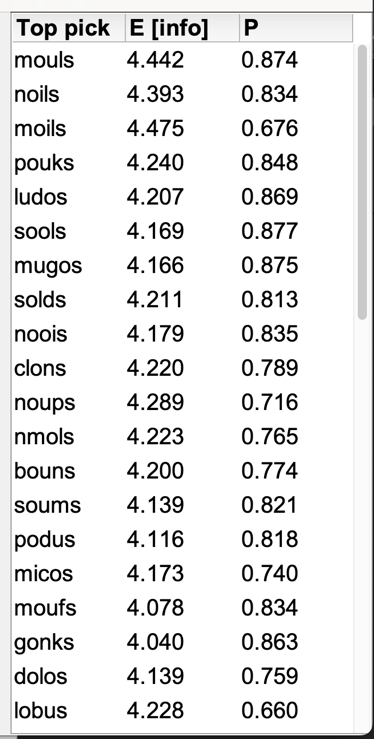
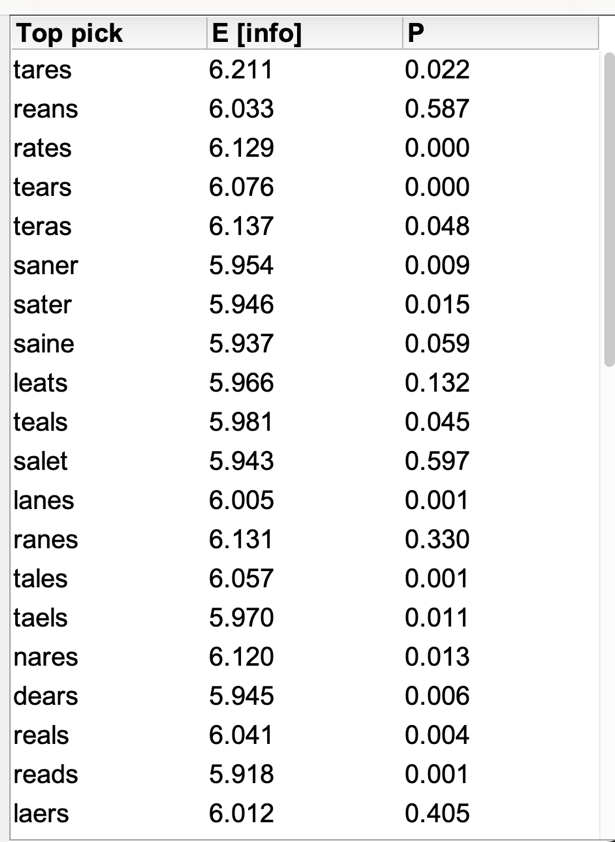
Hacker app:



Benchmarking:



Entropy table:



Table

Description automatically generated

Move logs:

Chart

Description automatically generated

### **Unit test:**

Graphical user interface, text

Description automatically generated

### **Analysis:**

### **Conclusion:**