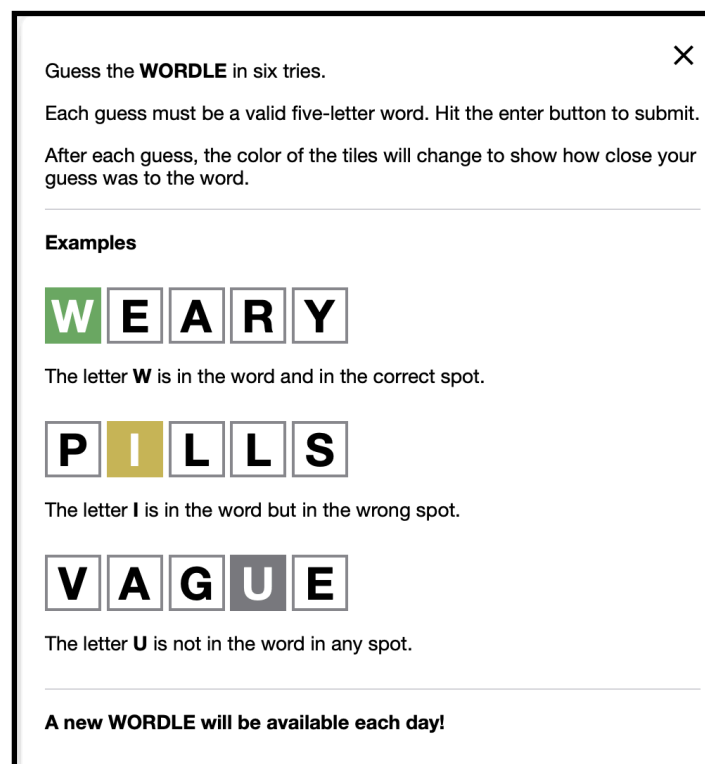


Extended Java Final Assignment: JWord

In this assignment you will implement your own version of the Internet game Wordle. If you aren't familiar with Wordle you can play it here:

<https://www.nytimes.com/games/wordle/index.html>

The idea of the game is to guess the correct 5 letter word in as few attempts as possible. You have six tries. For each guess of the word, the game tells you if you have a letter in the correct spot or if you have a letter in an in-correct spot. In the original game it does this with colour coding.



Requirements

To recreate the game you should use Java, the Swing library and other techniques we have learnt in the module. In the original game a new word is selected each day to guess. Your version must be implemented in the following ways:

- A new word to guess is generated each time the program is run (or new game is selected by the player if your implementation includes this.)
- The game should run entirely locally (without using Internet services)
- There should be no links to social media services.
- Your GUI should be written in Swing as demonstrated in class.
- The game must display the target word in the console (so that we can mark them without having to solve hundreds of wordles!)

You should start your implementation using the provided IntelliJ template. This includes two text files and a Main.java file.

File	Use
Main.java	includes a main method that loads in the two files. You can launch your program specific classes from here. Also includes another static method that returns the target word for players to guess.
gameDictinoary.txt	A dictionary of 5 letter words in English
targetWords.txt	A subset of the dictionary to be used as words the player has to guess. Archaic words and unusual spellings aren't included here.

Marking guide

Your implementation will be marked out of 50 as follows.

- **15 marks** for basic functionality. Have you implemented a playable version of wordle.
- **10 marks** for error checking and stability. Does the game correctly detect incorrect input and not crash.
- **10 marks** for an effective GUI and use of Swing components and layout
- **10 marks** for extending beyond the basic game implementation and adding additional features
- **5 marks** for code quality and modularity.

It's up to you how you implement the game but for the extension marks we will be looking for you to have used the language features and techniques shown in class to make the game more interesting. But please remember the requirements above and keep your game markable (e.g. printing solutions to the console).

In the accompanying assignment video I demonstrate an implementation that would score around 50% on the above mark scheme.