Wordle.java

Objective: Use logic, string manipulation, and display commands to make a working game.

Background:
Wordle is a game in which the user gets six tries to guess a five-letter word. It was originally created as a web-based online game around 2014 and quickly rose in popularity over the years. In 2022 the game was purchased by The New York Times and the paper has kept the game up-to-date since then.

One **Wordle** word is released each day, so you have to wait until the next day to play again. The word does not contain plurals (ending in "s"), or past tense (ending in "ed"). This cuts down on the number of words to guess.

Wordle only accepts guesses that are legitimate words. Made up words are not considered a guess. When a legitimate guess is presented, the game follows a **MasterMind** convention of color-coding the letters for each guess to give the user clues to the goal word. Exact matched guessed letters are colored green and partially matched letters are colored yellow. Guessed letters that are not in the goal word are colored dark gray. For example, let's use the goal word "THREE" and the guess "TREED". The guessed "T" and the second "E" would be green (exact), the "R" and the first "E" would be yellow (partial), and the "D" would be dark gray (no match).

Wordle uses two lists of words for play. The first, shorter list of 5-letter words is used to supply a goal word. The second, longer list of 5-letter words **Wordle** uses as valid guess words. The files have overlap since the goal list is a subset of the guess list.

Let's create a **Wordle** that functions like the original game but can be played many times in a sitting.

Assignment:

1. Download the **Wordle.zip** file from Mr Greenstein's web site and unzip (expand) the file. It will create a **Wordle** directory that contains a number of files. Add **FileUtils.java** to the directory for file IO.
   There are graphics files with extension ".png". These include keyboard backgrounds, guess letter backgrounds, and the **Wordle** game background.
   There are ".java" files. **Constants.java** has constants used for the graphics. **Wordle.java** is the game file you will edit. **StdDraw.java** is a bundle of graphics methods used by **Wordle** that make it easier to draw the graphics.
   There are two text files. **words5.txt** is the smaller list of potential goal words. **words5allowed.txt** is the larger list of valid guesses.
   Finally, there is a fully functioning **Wordle** game in the **Wordle.jar** file. To run the program you can enter:

   ```
   % java -cp Wordle.jar Wordle
   ```

   There are two options for the program. The first option is "show" to show the goal word of the game (for cheating). The second option is to input a goal word (for testing).

   ```
   % java -cp Wordle.jar Wordle show dwarf
   ```

   Play the game and get a feel for the interface. It is also instructive to compile the **Wordle** files you are given and run the program to see how it works before modifying the **Wordle.java** file. The goal word is hard-coded to be "SMART", but you will make the program select the word at random.

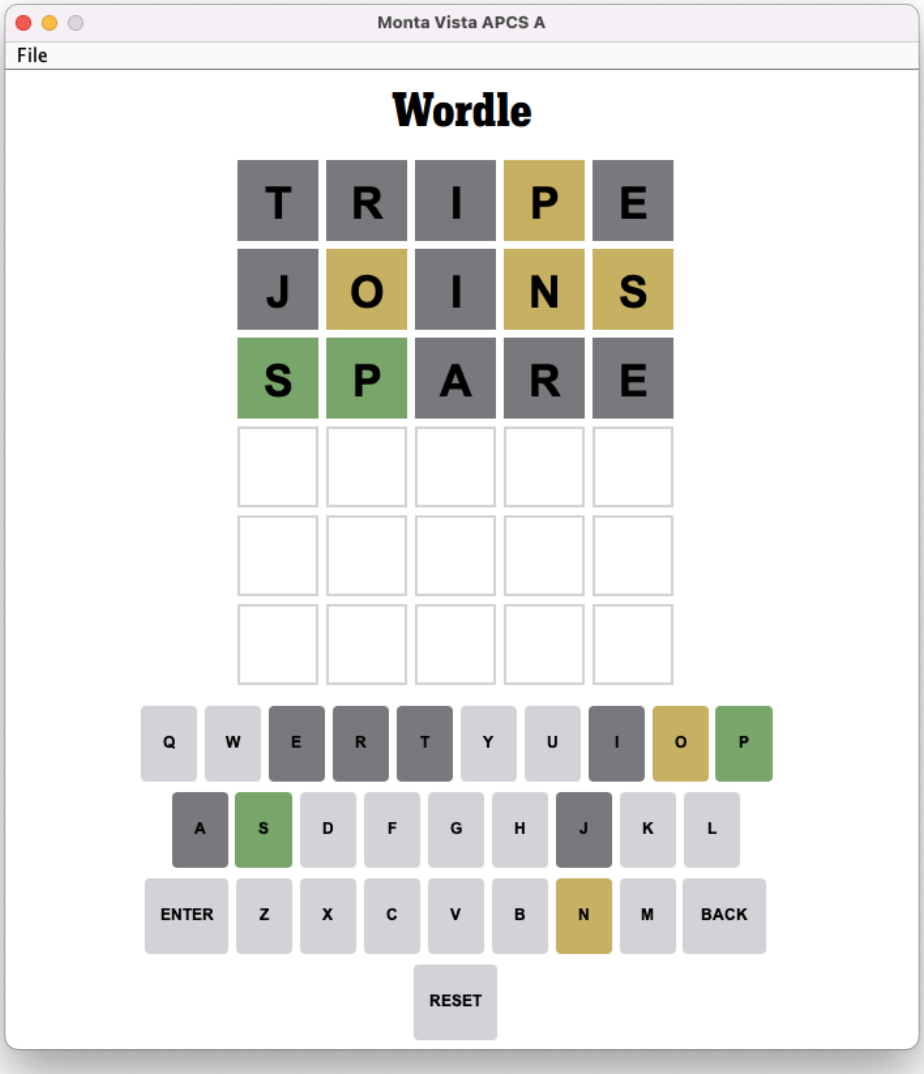
   ```
   % javac *.java
   % java Wordle
   ```

2. We will go through the **Wordle.java** code together. Note the methods labeled "THIS METHOD IS INCOMPLETE." These are the methods you will modify and add code.

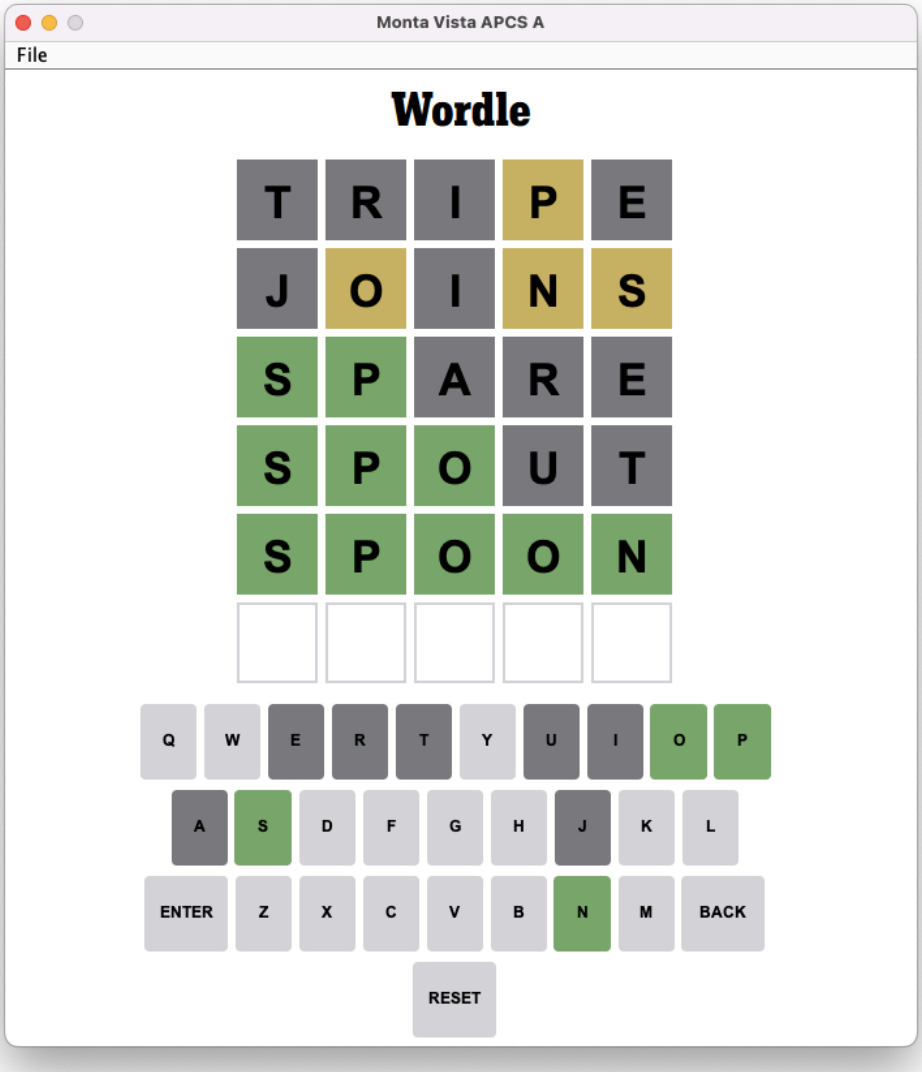
3. You have two primary tasks: (a) deal with the two text files, and (b) color the guess words and the keyboard appropriately. Goal words are taken from the **words5.txt** file and guesses must match words in the **words5allowed.txt** file. The color green is used for exact matches between the guess and goal words. The color yellow is used for partial matches between the words, and dark gray is used for no match between the words. The guess letters and keyboard letters should follow this color scheme. Take particular note of how the **Wordle.jar** file operates in coloring different guess and goal combinations.

4. Your program should handle two command line options. The first option is "show", like in "java Wordle show", that will print the goal word to the terminal. The second is a goal word, like "KNURS" in "java Wordle show knurs", that must match a word in the **words5allowed.txt** file. If the second option is a word not in the **words5allowed.txt** file then **Wordle** picks a word from the **words5.txt** file.

Sample run 1: A winning game. The first three guesses shows the following.

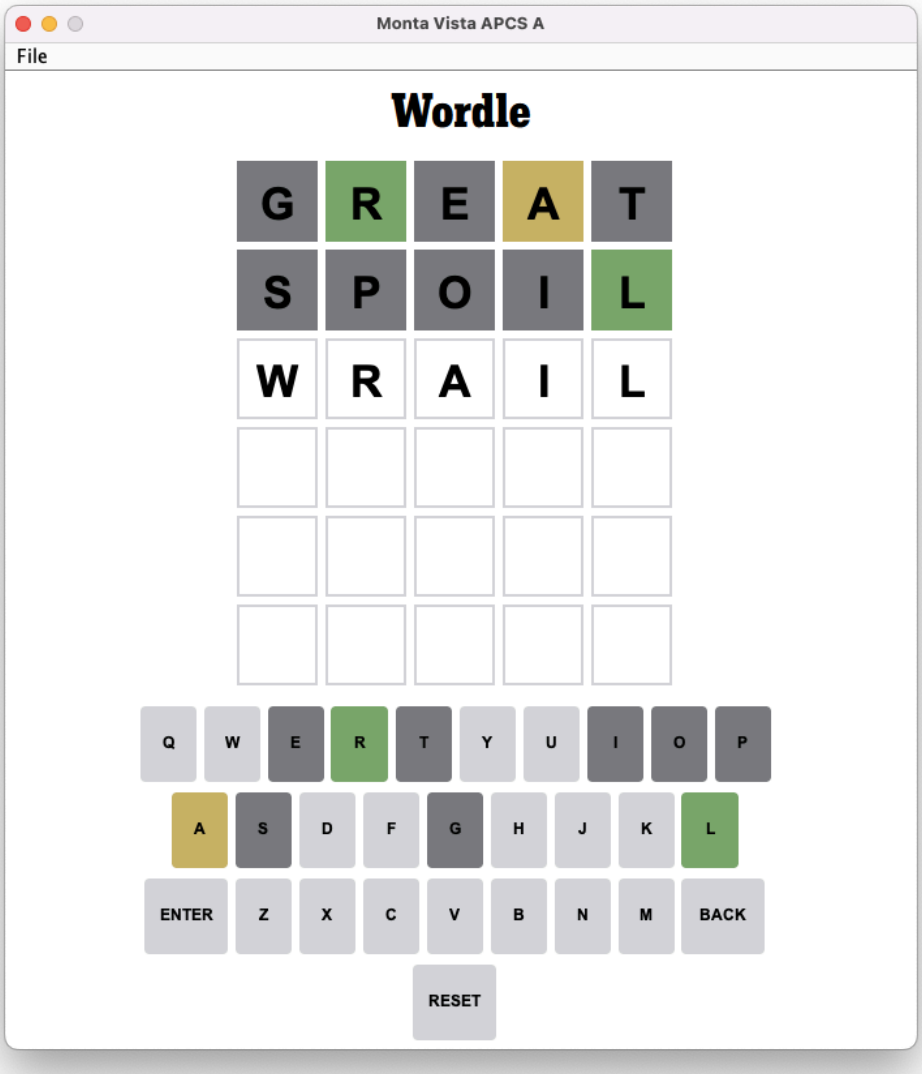


After five guesses the game is won.

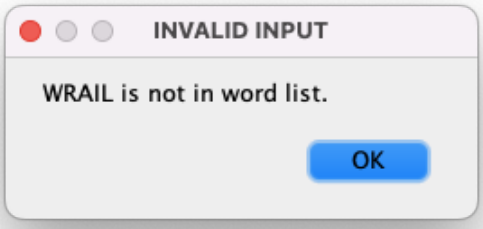


A winning JOptionPane with JDialog box appears like this.

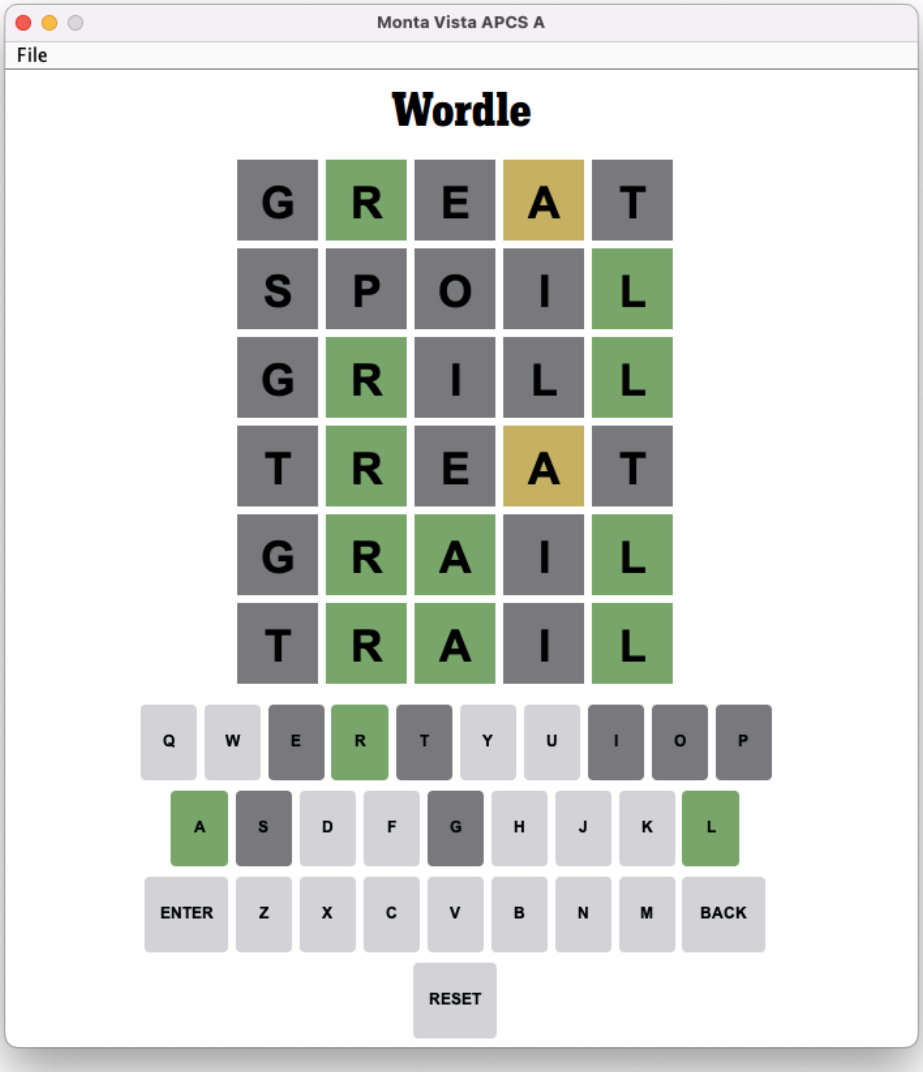
Sample 2: An invalid guess.



An invalid JOptionPane with JDialog box appears like this.



Sample 3: A losing game.



When the player loses, it should provide an JOptionPane with JDialog box like this.