

CompSci 201

Data Structures and Algorithms

Jotto

Jotto is a word game played with five-letter words similar to the colored-peg game **Mastermind**. The [Jotto Wikipedia](#) page has information about Jotto and variations on the game including changing the number of letters and providing different information than the number of letters-in-common. An example of a score/play sheet from the original game is shown on the left and a complete page can be found on the Wikipedia site.

For this assignment you'll be writing a computer program to guess a human player's word. The computer program will be really hard to beat even when it's using a straightforward, brute force approach to guessing. For extra credit you can write code that makes the game more "intelligent" by eliminating letters that cannot be part of a word rather than simply eliminating words as described below. An example of this increased intelligence is described below with a link to an online version of the game.

A variation of the game called **bagels** was one of the original [Nifty Assignments](#).

The Assignment

You will add code to the class `JottoModel.java` so that a game can be played using the GUI interface. You will need to implement specific methods and add instance variables in the `JottoModel` class to allow the game to be played. You should then test your code to make sure it works. Details are in the Jotto [How To](#).

For extra credit you can make your program smarter than the standard, simple, **eliminate words** game-playing strategy outlined here and in the [How To](#).

An explanation on the logic of how to play Jotto can be found [here](#).

Duke Jotto

There are two modes for playing Jotto, a simple elimination-strategy mode and a smart/AI mode for extra credit. The elimination-strategy mode, which you will implement for this assignment, is outlined below and a more in-depth description can be found in the [How To](#). The smart/AI mode can be added for extra credit and should use additional strategy and logic.

How to play



A screen shot of me beginning a [New Game](#) from the menu so that the computer can guess my word.

To begin play you must first load in a word file. Go to **File -> Open** and select a .txt file. We have provided you with the file kwords5.txt which is full of 5-letter words for playing Jotto. After loading in your file you must begin the game. Start the game by going to **New Game -> Play New Game**. To the left is a screen shot of beginning a **New Game** from the menu so that the computer can guess my word.

When the computer guesses your word, the code you write will **not** mimic the human logic described [here](#). For

extra credit you can use some human-like reasoning, but as you'll see the method you will implement allows the computer to guess most words very quickly.

The main idea is for the computer to guess a random word from a list of possible secret words. The user then responds with the number of letters in common between the user's secret word and the computer's guessed word. The computer then eliminates every word in its dictionary that doesn't have this many letters in common.

For example, if the computer guesses **fruit** and the user responds that there are two letters in common with the user's secret word, the computer can remove **ghost, ruins, tires** and lots of other words from the dictionary since these words have, respectively, one, three, and three letters in common with **fruit** and thus cannot be the secret word. The computer would leave **flips, track**, and others in the dictionary since these have two letters in common with **fruit** and could possibly be the user's secret word.

The computer continues to guess a word that could be the secret word until either the computer guesses correctly or there are no words left to guess. The latter could happen, for example, if the user's word is not in the computer's dictionary or if the user makes a mistake in responding about the number of letters in common (see below for examples).

Example game

Here are numerous examples of Duke Jotto trying to guess the secret word **radii**. The computer generates a guess and I type in the number of letters in common with **radii**. To signify that the computer has guessed my word I type the number 6 to indicate this is my secret word. This is an easy way to convey "you guessed right" to the computer which in this version of the game expects a number from the user.



Duke Jotto guessing the word **radii**. The computer has guessed my word in nine guesses and I type the number 6 to indicate this is my secret word.



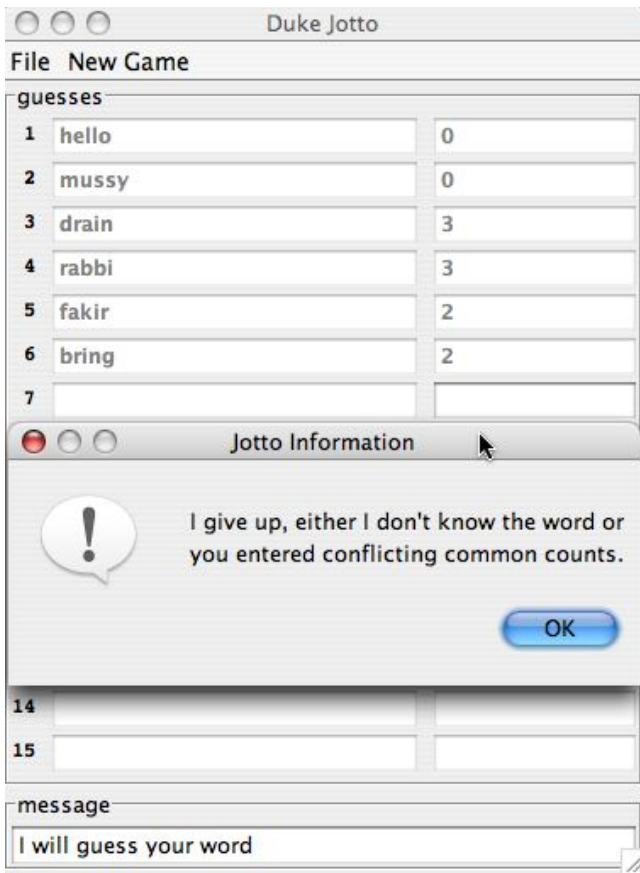
Notice that the computer does not guess the same sequence of words every time.



And sometimes the computer guesses my word very quickly.



The computer can make weird guesses before finding my word.



I made a mistake and entered **3** for the word **drain** which has four letters in common. The computer can't account for this mistake and indicates no more guesses after six guesses.

Submit

Submit your code using the submit name **jotto**.

- Submit all the code you write likely this is in the class `JottoModel.java`.
- submit a README.txt file: which should include all the information [here](#) as specified in the general assignment page.
 - Also include in your README file any bugs or problems you notice in your program or write-up.
- If you are handing in extra credit you must also submit an ExtraCredit.pdf that describes your logic for smart/AI

Things That Might Be Graded

To get full credit on an assignment your code should be able to do the following.

- Being able to play more than one full game.
- Ensuring the computer plays randomly at all times.
- Being able to correctly count common letters and find correctly matched words.
- Your game should be at least as robust as described in the assignment (correctly displaying given errors, correctly interpreting win/loss, multiple games without word loss, etc).
- Correctly handling guessed or mismatched words.
- Correct message text **and** placement.
- Your code should be clear and easy to read.
- Your variable names should make sense. If you name a variable that keeps track of guesses something vague , unrelated, or inappropriate makes grading much harder and you may be penalized.
- You ***MUST*** create and submit a README with your work.
- If you have questions about the assignment that are not specific to your code, please ask on Piazza and we will be happy to clarify. If your question is specific to the code you've written please post it privately on Piazza, or bring it to the Link or Office Hours and don't share it directly with your classmates.

This list is not a complete rubric for grading, and there is a better breakdown on the main page. Please start early, work carefully, and ask any questions you have and we'll do our best to make sure that everyone gets all of the points that they deserve.
