COS 126 Programming Exam 1 Fall 2020: WordGuess

Description. WordGuess is an interactive guessing game in which a player must discover a secret word by guessing the characters it contains. The player wins by correctly guessing all the characters in the word before exhausting a number of allowed incorrect character guesses. Your job is to fix the `main` function given `WordGuess.java`. (Tip: Compile and try the provided version first. Comments in it highlight some of the problems.) Do not use any library functions not already used in `main`.

Input and Output. WordGuess is an interactive game played in the terminal on standard input and output. Example 1 on the next page shows a complete game session ending with a loss for the player. Example 2 shows a second game session ending in a win. Your program must behave exactly as shown.

Arguments. WordGuess takes two arguments. The first argument is the filename of the dictionary from which a random word will be selected. The second argument is an integer value indicating the total number of allowed incorrect character guesses before the player loses.

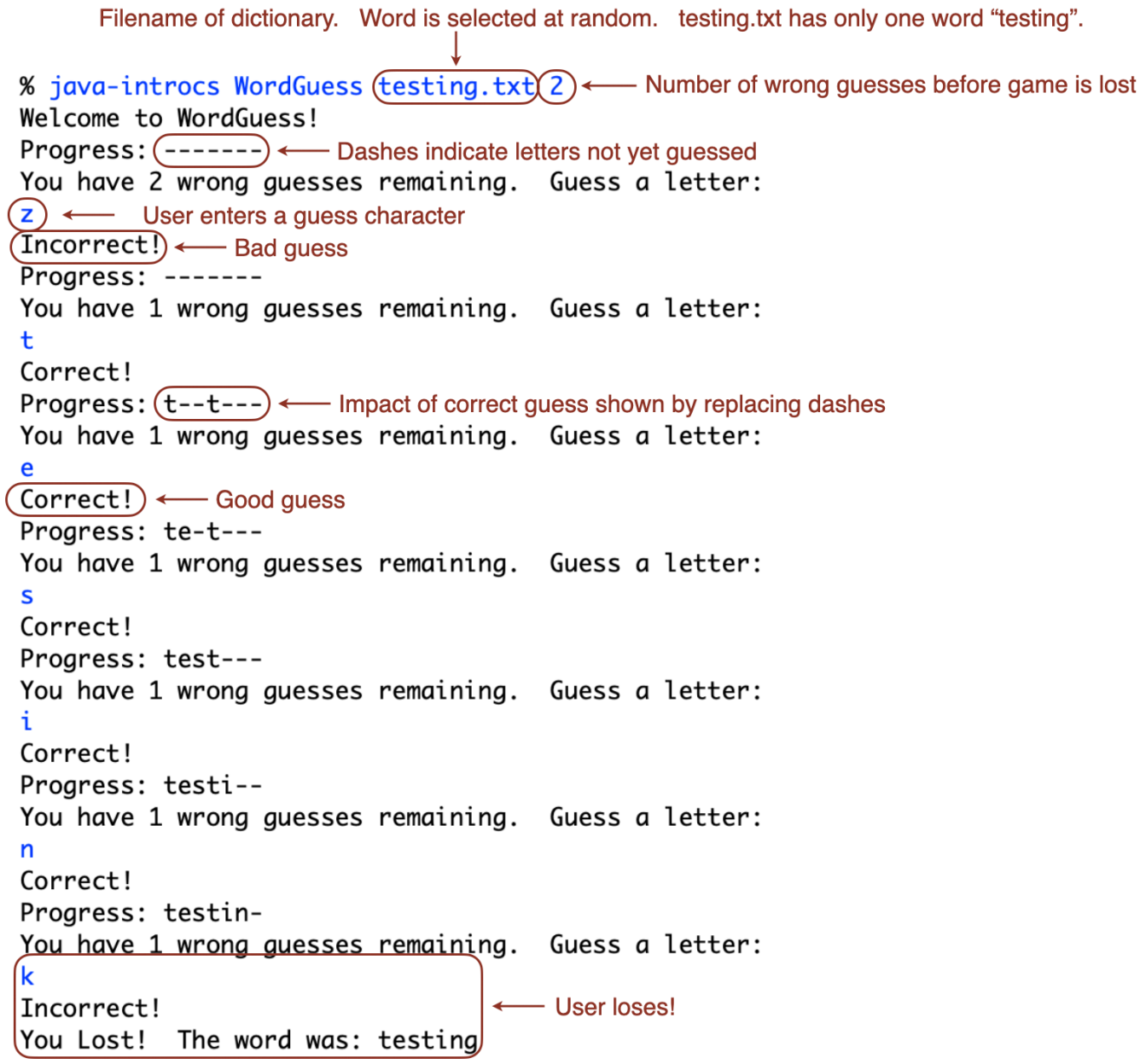
Zip File Contents (other than the PDFs).

- `WordGuess.java` is a broken implementation of WordGuess, but it is a good starting point for you. Fix its `main` function, and submit this file. Do not modify any other function.
- `StdGame.java` implements functions used by `WordGuess.java`. Understand each function from its context in `WordGuess.java`. You do not need to examine `StdGame.java`. Do not modify it.
- `dict.txt` contains a list of 21 words.
- `testing.txt` contains only one word. Tip: Use this dictionary for testing your code with predictability.
- `HONORCODE.TXT` is for signing the honor code. Submit this file along with `WordGuess.java`.

Algorithm. Your fixed `WordGuess.java` should perform the algorithm described here. Some of these steps are already correctly implemented in the given `WordGuess.java`. Comments in `WordGuess.java` highlight some of the problems.

1. Using `StdGame`, get a random word (as a `char` array, not as a `String`) from the dictionary specified by the first command line argument. Also using `StdGame`, create a parallel `progress` character array of the same size initialized to hold a dash for each character of the word.
2. Welcome the user to the game: "`Welcome to WordGuess!`"
3. While the number of incorrect guesses remaining (initially set by the second command line argument) is not zero and the user has not correctly guessed all of the characters, do:
   - **3.1.** Using the `getNextGuess` function, report the progress with the number of guesses remaining and solicit a character guess. (`getNextGuess` will convert uppercase letters to lower case.)
   - **3.2.** If the character exists one or more times in the word, replace each corresponding dash in the progress array with that character and report that the guess was correct: "`Correct!`" Note: You can handle repeated guesses of the same character however you would like.
   - **3.3.** If the character does not exist in the word, report the guess as incorrect ("`Incorrect!`") and update the number of incorrect guesses remaining. Note: You can handle repeated guesses of the same character however you would like.
4. When done with Step 3, if no incorrect guesses remain, report "`You Lost! The word was: <word>`". If no unguessed characters remain in progress report "`You Won! The word was: <word>`".

Example 1: Player Loses



Example 2: Player Wins

```
% java-introcs WordGuess testing.txt 2
Welcome to WordGuess!
Progress: -------
You have 2 wrong guesses remaining.  Guess a letter:
t
Correct!
Progress: t--t---
You have 2 wrong guesses remaining.  Guess a letter:
e
Correct!
Progress: te-t---
You have 2 wrong guesses remaining.  Guess a letter:
s
Correct!
Progress: test---
You have 2 wrong guesses remaining.  Guess a letter:
i
Correct!
Progress: testi--
You have 2 wrong guesses remaining.  Guess a letter:
n
Correct!
Progress: testin-
You have 2 wrong guesses remaining.  Guess a letter:
g
Correct!
You Won!  The word was: testing
```

← User wins!