

CSE 2010, Term Project, Fall 2016

Due Wed Nov 30 at the start of your lab section; Submit
Server: class = cse2010, assignment = termProjectSxInitial

Due Wed Dec 7 at the start of your lab section; Submit
Server: class = cse2010, assignment = termProjectSxFinal
 x is 1, 2, 3, or 4—your section number.

On a smartphone, the keyboard is small and typing is not as easy as on a regular-size keyboard. Wouldn't you like to be able to type easier and faster? How would you design a system that can help you and other users type quickly and accurately?

The challenge of the term project is to design and build SmartWord, a system that can guess words based on partial words. For each word, given a letter at a time, your system will provide 3 guesses. If one of the guesses is correct, the rest of the letters are skipped (yeah!). You and other users desire accuracy, speed and low memory usage, which can be measured by:

- accuracy (average % of letters skipped for each word)
- average time used to generate 3 guesses for a word
- number of bytes used in the memory
- $score = accuracy / \sqrt{time * memory}$

Note that an iPhone user can type 30 words per minute¹ and the average word length is 5.1 letters². Hence, a user types around 1 letter per 0.4 seconds. See how fast you can type at <http://10fastfingers.com/mobile>.

Program Files: Among your program files, you will provide SmartWord.java that supports at least:

- SmartWord(...) // constructor
- void processOldMessages(...) // user's old messages
- String[] guess(...) // generates 3 guesses
- void feedback(...) // feedback from EvalSmartWord

We will provide EvalSmartWord.java (not to be modified) to help you evaluate your system. EvalSmartWord.java and a template of SmartWord.java with details of the above methods are on the course website.

You may use any program segment/file from the textbook. However, you may *NOT* use program segments/files from other sources. You may borrow ideas on data structures and algorithms from other sources, but you need to clearly cite them at the top of your program files and implement them on your own.

Input: Input is from the command-line arguments for EvalSmartWord.java in this order:

- filename for a list of known words, one word per line
- filename for old messages to simulate old user input, one message per line
- filename for new messages to simulate new user input, one message per line

Sample input files are on the course website.

You may have additional “hardcoded” (not from command line or prompt) input data files for SmartWord.java and/or other program files.

Output: Measured performance by EvalSmartWord.java goes to the standard output (screen).

Presentation: Project presentations will be during the lab on Dec 7. Create a presentation file following this outline:

1. Title, Group Name, and Members
2. Goal and Motivation
3. Initial Submission
 - (a) Algorithms and supporting data structures
 - (b) Additional input data and reasons
 - (c) Ideas devised within the group
 - (d) Ideas discussed in the course/book, cite them
 - (e) Ideas borrowed from other sources, cite them
4. Final Submission
 - (a) Changes in algorithms and supporting data structures
 - (b) Changes in additional input data and reasons
 - (c) Ideas devised within the group
 - (d) Ideas discussed in the course/book, cite them
 - (e) Ideas borrowed from other sources, cite them
5. Evaluation
 - (a) Accuracy, time, memory, and score: initial vs. final submission for each data set
6. Analysis
 - (a) Why more/less accurate?
 - (b) Why faster/slower (including time complexity— n words, each word has m characters)?
 - (c) Why more/less memory?
 - (d) Possible further improvements

Evaluation:

- Initial submission (20%) – including performance on accuracy and score
- Final submission (40%) – including performance on accuracy and score
- Presentation (20%, oral and written)
- Teammate evaluation (20%)

Submission: Initial submission has SmartWord.java, other program files, and additional data files (if any).

Final submission has SmartWord.java, other program files, additional data files (if any), and presentation (preferably in PDF).

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top of the assignment.

¹<http://www.pocketables.com/archives/mobile-device-keyboard>

²<http://www.wolframalpha.com/input/?i=average+english+word+length>