CSE 2010, Term Project, Spring 2024

Due Tue Apr 16 at the start of your lab section; Submit Server: class = cse2010, assignment = termProjectSxInitial

Due Tue Apr 23 at the start of your lab section; Submit Server: class = cse2010, assignment = termProjectSxFinal

x is 14 or 23 (merged section number).

The game of Hangman asks a player to guess a hidden word one letter at a time. The player is allowed up to 6 incorrect guesses (body parts). How would you design a system that can guess letters/words correctly and quickly?

The challenge of the term project is to design and build HangmanPlayer, a system that can guess letters (and hence the hidden word) based on partial (initially blank) words. We desire a system that is accurate, fast and has low memory usage, which can be measured by:

- average accuracy for each hidden word, where

$$accuacy = (1 - numIncorrectGuesses/6) * 100\%$$

- average time used to generate 1 guess and process feedback
- number of bytes used in the memory
- $score = accuracy^2/\sqrt{time * memory}$

Guessing a letter more than once is an incorrect guess.

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

- HangmanPlayer(...) // initialization/constructor
- guess(...) // guess 1 letter
- feedback(...) // from EvalHangmanPlayer

We will provide EvalHangmanPlayer.java (NOT TO BE MODIFIED) to help you evaluate your system. EvalHangmanPlayer.java and a template of HangmanPlayer.java with details of the above functions are on the course website. The initialization/preprocessing should not take more than 3 minutes. Your submission will be evaluated on code01.fit.edu, we strongly recommend you to ensure it functions properly on code01.fit.edu.

You may use any program segment/file from the textbook and any standard Java libraries without extra installation. 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 EvalHangmanPlayer.java in this order:

- filename for a list of known words, one word per line
- filename for a list of hidden words to be guessed

Sample input files are on the course website.

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

Output: Measured performance by EvalHangmanPlayer.java goes to the standand output (screen).

Presentation: Project presentations (about 10 minutes each) will be during the lab on Apr 23. Create a presentation file following this outline:

1. Title, Group Name, and Members
2. Goal and Motivation
3. Initial Approach/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 Approach/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: Accuracy, time, memory, and score: initial vs. final approach/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%)
- Extra Credit (up to 10 points): Final Submission has an accuracy above 70, using less than 0.1 seconds and 1×10^8 bytes (~100 MB), as measured by EvalHangmanPlayer.java on different lists of hidden words. Accuracy of at least 71 earns one extra point, at least 72 earns two extra points, ... Late submission is not applicable.

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

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

To reduce issues on presentation day, we suggest you to send the presentation file in PDF to the TA/GSA via email (in addition to Submit Server). Your laptop might not have a display port compatible with the projector.

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