

Unix Tools
Courant Institute of Mathematical Sciences
Project
Due: April 30, 2008

The objective of this project is to design a web-based scrabble game. If you are not familiar with the game, you can first consult:

<http://en.wikipedia.org/wiki/Scrabble>

and try an online version such as

http://www.scrabulous.com/solitaire_scrabulous.php.

Your programs should be written as a series of CGI scripts. The instructions for using CGI scripts on i5 are given under: <http://i5.nyu.edu>.

0. Familiarize yourself with writing CGI scripts under i5.
1. Design a module for username registration and authentication.
2. Write a module to maintain and display user statistics (e.g., number of games played, score, best scores across users) in an ascii file.
3. Write a module for selecting timed or unlimited time game. For the timed game, allow for a number of options for game times: 1 min, 5 min, 10 min.
4. Write a module implementing a version of the game where the user plays against the computer.
 - Together the players have 100 letters. For the English-language edition of scrabble see

http://en.wikipedia.org/wiki/Scrabble_letter_distributions.

The player starts out with 7 letters, drawn independently at random from the 100 letters. Subsequent letters should likewise be drawn independently at random from the remaining letters [remember also to take out the letters used by the computer].

- It should be possible to type a letter on an empty field and thereby remove that letter from the player's available letters. Dragging letters to the desired fields would provide bonus points, but that might require java scripts.

- A blank letter tile can be used to replace any letter. When the player has a blank tile but does not have the letter he is typing in a field, the computer should assume that he is using a blank tile.
 - Remember to check if the first word covers the center field.
 - When the player hits `submit`, the back-end of the game module must check the proposed word against the dictionary. We do not have the official scrabble dictionary online, so we will have to work with the dictionary on `/usr/dict/websters`. The match should be case-insensitive.
 - If the word does not exist, the letters should be moved out of the board again, back to the players area.
 - If the word is valid produce the appropriate scoring according to the rules of the game, update the scoreboard, and draw new letters for the player.
 - Let the computer play its turn by generating a word using the dictionary and the current board configuration and the remaining letters. Update the board, the available tiles, and the score of the computer.
 - If the player cannot form a word, allow for the player to pass.
 - Continue till there are no more letters left, and compute the final score for the player. See <http://en.wikipedia.org/wiki/Scrabble>.
5. Create or use existing images available on the web for the letter tiles and the board fields (2 x word, 2 x letter, etc.)
 6. Create a browser interface for the game. The interface should allow the user to click on a single field. This action will call the game back-end module, which will return a new configuration to be displayed by the browser. The interface should include a button for `new game` and one for `score`.
 7. Use your creativity and programming skills to create the most elegant and effective game. Anything beyond what was mentioned above will give bonus points.