Hangman: Software Requirements Specification

Lars Cordewener
Software Engr.
Department CS
VU University

Arthur de Fluiter
OS and Security
Department CS
VU University

Radu Jica
Big data Engr
Department CS
VU University

Roy Overbeek
Foundations
Department CS
VU University

March 6, 2017

Requirements

To make the requirements more precise, we introduce the following definitions.

1. The *alphabet* consists of exactly the letters contained in the range a–Z.
2. A *word* is a sequence of non-whitespace characters with a length of at least 2 and at most 45.
3. A *word guess* is a word entered by the user on the command prompt, possibly surrounded by leading and trailing whitespace characters.
4. A *letter guess* is a single character entered by the user on the command prompt, possibly surrounded by leading and trailing whitespace characters.
5. A *guess* is either a word guess or a letter guess.
6. The *stripped guess associated with a guess x* is the result of removing all leading and trailing whitespace from x.
7. The *secret* is the word selected by the system for the user to guess.
8. The *partially revealed secret* is the secret in which 0 or more positions have been hidden from the user.
9. A word guess is *consistent* with the partially revealed secret iff the following propositions hold:
    - the length of the associated stripped guess matches the length of the secret;

- every revealed character at position i in the partially revealed secret is also at position i of the associated stripped guess.

10. Let x be the associated stripped guess of a letter guess y. Then y is *valid* iff the following propositions hold:

    - x is in the alphabet;
    - x is not in the list of previously guessed letters.

11. Let x be the associated stripped guess of a word guess y. Then y is *valid* iff the following propositions hold:

    - all the characters in x are in the alphabet;
    - x is not in the list of previously guessed words;
    - x is consistent with the partially revealed secret.

ID	Requirement	Description
FR_01	Guess word	The user must be able to enter a word guess.
FR_02	Guess letter	The user must be able to enter a letter guess.
FR_03	Display partial secret	While a game is in progress, the system must at all times display a partially revealed secret.
FR_04	Display guessed letters	While a game is in progress, the system must at all times display a list of guessed letters.
FR_05	Display guessed word	While a game is in progress, the system must at all times display a list of guessed words.
FR_06	Display hangman	While a game is in progress, the system must at all times display the hangman.
FR_07	Partial secret init	The partially revealed word initially hides all positions of the secret.
FR_08	Guessed letter init	The list of guessed letters is initially empty.
FR_09	Guessed word init	The list of guessed words is initially empty.
FR_10	Hangman init	The hangman is initially empty.
FR_11	Letter guess 1	If a valid letter guess c is made, and c is among the hidden positions of the revealed word, then the system should reveal all positions that contain the character c.
FR_12	Letter guess 2	If a valid letter guess c is made, and c is not among the hidden positions of the revealed word, then the system should extend the hangman.
FR_13	Letter guess 3	If a valid letter guess c is made, then c should be added to the list of guessed letters.
FR_14	Letter guess 4	If an invalid letter guess is made, then the user should get feedback on exactly one violated property (see definitions), and be prompted to try again.
FR_15	Word guess 1	If a valid word guess w is made, and w matches the secret, then the system should reveal all positions of the partially revealed secret.
FR_16	Word guess 2	If a valid word guess w is made, and w does not match the secret, then the system should extend the hangman.
FR_17	Word guess 3	If a valid word guess w is made, then w should be added to the list of guessed words.
FR_18	Word guess 4	If an invalid word guess is made, then the user should get feedback on exactly one violated property (see definitions), and be prompted to try again.
FR_19	Termination 1	If the partially revealed secret contains 0 hidden characters, then the game ends and the player wins.
FR_20	Termination 2	If the hangman is complete, then the game ends and the player loses.
FR_21	Play again	If the game ends, then the player is asked whether they want to play again.

Table 1: Functional Requirements

ID	**Requirement**	**Description**
QR_01	Ease of use	It should be easy to use, with only the use of a terminal/bash.
QR_02	Correctness	While checking the word for guesses the system should not make any mistakes in allowing or failing letters that should be in the word or the correct guess of the full word.
QR_03	Input	The system should present an error when an unacceptable input is given.
QR_03	Speed	Due to the simple nature, the system should not take long picking a new word or checking each choice of letter or guessed word.
QR_04	Game length	The system should ensure that each game allows for the same amount of mistakes.
QR_05	Presentation	The system should, at all times, correctly display the current state of the game. This includes the chosen colour, letters, words, current word, and representation of the amount of errors.

Table 2: Quality Requirements