Hangman Game Technical Document

Guy Needham

May 4, 2013

1 Time Taken

In total, this program took me about 16 hours to write. Most of this time was taken up by making the GUI.

2 Major Design Decisions

I used the model-view-controller philosophy when making this project. There is a class called HangmanSwing which acts to control the rest of the classes. The Play class is the model where the game logic occurs. The View class controls how the user views the game. The Control class allows the Play class and the View class to interact. I used the Java Swing components to create the GUI as they were well supported on the web.

3 Data Structures

I chose to represent the hints and words in a list with the same index in each list for each pair of words and hints. This was so that I could use a random number generator to change which word to pick, and so that the hint would be linked to the word.

4 Program Logic

The program works by a random number generator picking a word and hint. The hint is displayed to the user as a series of dashes. The user inputs a character which is compared to the word. If it is present in the word, the dash representing it is changed to that character. If not, the integer stateOfMan increases and the wrong character is added to a hashset. The change in stateOfMan causes the next stage of the hangman to be displayed. If the length of the hashset containing incorrect characters is equal to the integer determining the maximum number of guesses, the player loses the game. The player wins the game by inputting all the characters in the word before inputting too many incorrect guesses.

5 Problems and Limitations

The field is case sensitive, which could annoy users. A major limitation of this game is that to play again, you have you have to shut down and restart the game. A minor problem is that the GUI isn't very attractive.