The Hangman Game implementation

Your task is to implement the simple hangman according to functional requirements and your best knowledge.

Source code must be located in the github repository. If you don’t have a github account, create one at https://github.com/. During hangman development work with git according to your best known practices.

Write your solution in C#. You have 4 days to deliver a given application. When the task is finished please send a github link as a reply.

Functional requirements were delivered by the client.

1. Create a list of European capitals, pick one of them randomly and let the user guess it. At the beginning the program should represent each letter as a dash __("_")__ and display them at the screen. Additionally, the program should show the player's life points (let's say, 5).
2. Program should ask the user if he/she would like to guess a letter or whole word(s). Next, the program waits for the user to enter a letter or word. If the entered letter doesn't exist in word or entered word is not correct - player will lose a life point. If this action brings player life to zero - the game is over!
3. If the player survives a wrong letter guess - that letter should be added to the "not-in-word" list and displayed on the screen.
4. If the player guesses the final letter or whole word(s) - he/she is the winner!
5. Add a question about restarting the program after wins or loses.
6. **OPTIONAL** Add an information about guessing count and guessing time at the end of game (i.e. "You guessed the capital after 12 letters. It took you 45 seconds").
7. There is a file attached ”countries-and-capitals.txt" containing a list of countries and their capitals (i.e. Poland | Warsaw). Your program should read that file at the beginning and randomly select one country-capital pair. Then, the capital should be the target word(s) to guess. The country should also be remembered - if player will reached his/her life points program should display a hint (i.e. "The capital of Poland").

8. Guessing the whole word should be more-risk-more-reward. So, a successful guess can save some time, but failing the whole word guess should result in losing 2 life points!
9. Add a high score - some people take pride in their score. At the end of successful game program should ask user for his/her name and save that information to a file - name| date | guessing_time | guessing_tries | guessed_word (i.e. Marcin | 26.10.2016 14:15 | 45 | Warsaw).
10. **OPTIONAL** Expand high score - program should remember 10 best scores (read from and write to file) and display them at the end, after success / failure.
11. **OPTIONAL** Add ASCII art! How awesome it will be if after each wrong guess a part of hangman appears?

In case you are not familiar with hangman flow - you can check out its description on wikipedia - Hangman (game)

Remember - there is no one perfect solution for the application.

If you reach this far - please add to your Readme.md just a few sentences on why do you code - is it just a professional matter or something more? We would like to know the answer!

Good luck!
Motorola Academy team