# Homework, Week 7, part D

lightprogramming.org

October 21, 2016

We have completed the basic structure of Hangman. All that's left is some final touches. The top level is a procedure named **start-hangman**. Technically this is *not* a function. Why? Because it doesn't take any arguments or return any values. For right now, we can call this a subprocedure, and many languages call these named blocks of code subprocedures, but mostly we call them functions, even though they are not really functions.

This procedure declares ands initializes four variables, *tries* initially set to 6, *answer* set by a call to **get-answer**, *good-guesses* initially set to an empty list, and *bad-guesses* also set initially to an empty list. It calls **greeting** for a friendly greeting to the players, and then calls **game-loop** passing it the answer, good-guesses, and bad-guesses.

```
1  (defun start−hangman ()
2    (setf tries 6)
3    (greeting)
4    (setf answer (get−answer))
5    (setf good−guesses ())
6    (setf bad−guesses ())
7    (game−loop answer good−guesses bad−guesses))
```

**greeting** also is not a function. Why not? It's purpose is to be friendly to the players.

```
1  (defun greeting ()
2    (format t "Let's play hangman!~%")
3    (format t "First player enters a word or pharse to be guessed.~%")
4    (format t "Second player guesses letters (six tries).~%"))
```

What have you learned?

Mainly, I hope you saw that no set of requirements is complete, at least in the beginning. You don't really know the complexity of a problem until you start to solve it. In this case, we had to deal with the fact that some words or phrases contain characters other than alphabetical characters, such as the space in "common lisp." We have to account for the problem that some characters are upper case and some are lower case, which we solved by converting all the characters to upper case. We had to figure out a data structure to use, in this case a list, and convert bretween lists and character strings. We had to solve the problem of deciding when the game was over, and whether the second player won or lost.

We have some problems that we didn't solve, such as — how to hide the initial answer entered by the first player from the second player. We will see one way to solve this in a few weeks. We may have some other improvements to make, such as preventing the guessing player from entering non-alphabetical characters. Still, we solved many of the problems, and you have a Hangman game that is at least playable, even if it's not perfect.