# Assignment 2

Due: 31.10.2025

## 1. Introduction

In this assignment, your task is to program the classic game “Hangman” in R. On ILIAS, you will find two files: `assignment2_demo.R` and `assignment2_functions.R`. Please download both and save them in your R working folder.

1. `assignment2_functions.R`: This is the file where you will write all the logic for the game. A skeleton structure is provided to guide you.
2. `assignment2_demo.R`: This file is used to run your code. It first loads your functions and then starts the game. The first line in `assignment2_demo.R` is:

   ```
   source("assignment2_functions.R")
   ```

   This command loads the helper functions you will write. The second line executes the main game function, which starts the game in the R console:

   ```
   play_hangman()
   ```

### The Logic of the Hangman Game

The game logic is as follows:

1. A random word is chosen from a predefined list.
2. The word is displayed to the player as a series of underscores, one for each letter.
3. The player guesses a letter.
4. If the guessed letter is in the word, the corresponding underscores are replaced with that letter.
5. If the letter is not in the word, a part of the hangman figure is drawn. The player has a limited number of incorrect guesses (typically 6).
6. The game ends when the player either guesses the entire word (win) or runs out of guesses (loss).

To build this game, you will need to use loops (`while`), conditionals (`if/else`), user input (`readline()`), and functions for string and vector manipulation.

## 2. Your Task

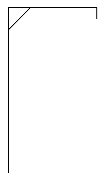
Your task is to complete the two functions in the `assignment2_functions.R` file: `draw_hangman()` and `play_hangman()`.

1. **Implement `draw_hangman(wrong_count)`** This function is responsible for drawing the hangman figure. It takes a single argument, `wrong_count`, which represents the number of incorrect guesses made so far.

   The drawing is cumulative. With 0 wrong guesses, only the gallows should be drawn. For each additional wrong guess, a new body part is added. You will use a series of `if`-statements to achieve this.

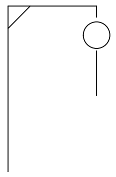
   For example, `draw_hangman(0)` should produce the gallows:

   ```
   plot.new()
   plot.window(xlim = c(0, 10), ylim = c(0, 10))
   draw_hangman(0)
   ```

   If `wrong_count` is 2, the head and body should be added to the gallows:

   ```
   plot.new()
   plot.window(xlim = c(0, 10), ylim = c(0, 10))
   draw_hangman(2)
   ```

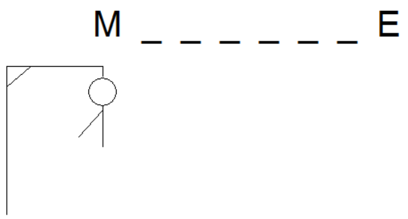
   

2. **Implement the `play_hangman()` function**

   This is the main function that controls the entire game flow. You will need to structure your code to handle the game's setup and the main game loop. Please follow the comments in the skeleton file `assignment2_functions.R` to implement the different parts of the function.

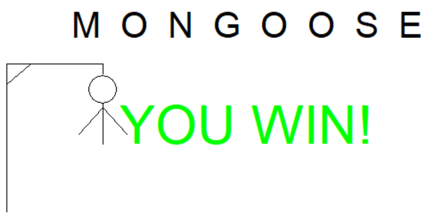
   The screenshots below show examples of what the game could look like in the plot window.

   **During the game:**



M _ _ _ _ _ _ E

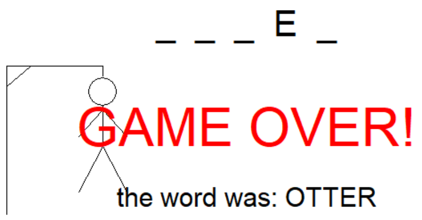
   **When it is won:**



M O N G O O S E

YOU WIN!

   **When it is lost:**



_ _ _ E _

GAME OVER!

the word was: OTTER