CS202: PROGRAMMING PARADIGMS & PRAGMATICS

Lab 4: Hangman in Perl!

Aim: Write a program to play a word-guessing game like Hangman.

- **Let's get started!**
  - Create a directory structure to hold your work for this course and all the subsequent labs:
    - Suggestion: `CS202/Lab4`

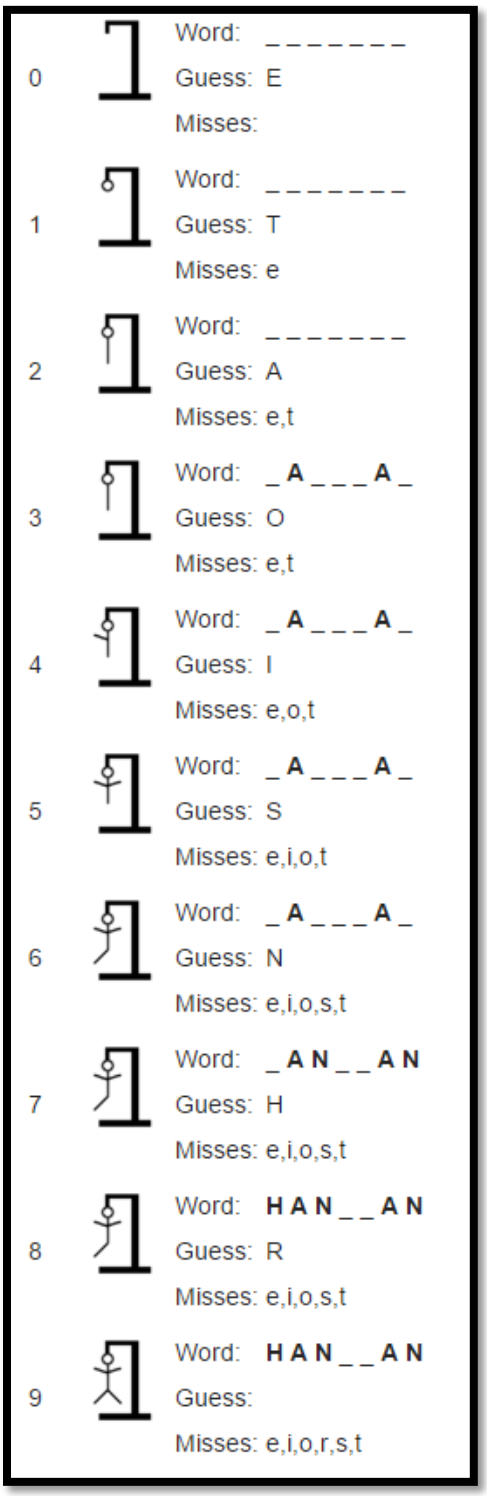
- **Introduction**
  - The word to guess is represented by a row of dashes, giving the number of letters in the word.
  - If the player suggests a letter which occurs in the word, it should write it in all its correct positions
  - If the suggested letter does not occur in the word, it should draw one element of a hanged man stick figure as a tally mark
  - The game is over when:
    - The player completes the word, or guesses the whole word correctly – Player Wins!
    - Number of tries exceeds a limit (completes the hangman diagram) – Player loses!

- **Game Simplified:**
  - It must randomly choose a word from a list of words.
  - It must stop when all the letters are guessed correctly.
  - It must give them limited tries and stop after they run out.
  - It must display letters they have already guessed (either only the incorrect guesses or all guesses).

- **Suggestions:**
  - If you are familiar with OOP in Perl, use classes whenever/wherever possible to imitate real world objects
  - If not, then try to modularize your program as much as possible into meaningful subroutines
  - **First get a simple text version of your game correct!!**
  - Then try drawing the hangman as shown in the examples.

- **Examples:**

```
Here is your word: -------
Guesses so far:
Make a guess: a
Good guess!  you have 6 body parts left

Here is your word: -----a-
Guesses so far: a
Make a guess: l
Bad guess - you have 5 body parts left

Here is your word: -----a-
Guesses so far: al
Make a guess: e
Bad guess - you have 4 body parts

Here is your word: -----a-
Guesses so far: ale
Make a guess: h
Bad guess - you have 3 body parts

Here is your word: -----a-
Guesses so far: aleh
Make a guess: p
Good guess!  you have 3 body parts

Here is your word: p----a-
Guesses so far: alehp
Make a guess: program
Yeah! you got it!

Do you want to play again (y/n)? n
```



- **Sample List of Words**

```
"computer|radio|calculator|teacher|bureau|police|geometry|president|subject|country|environment|classroom|animals|province|month|politics|puzzle|instrument|kitchen|language|vampire|ghost|solution|service|software|virus25|security|phonenumber|expert|website|agreement|support|compatibility|advanced|search|triathlon|immediately|encyclopedia|endurance|distance|nature|history|organization|international|championship|government|popularity|thousand|feature|wetsuit|fitness|legendary|variation|equal|approximately|segment|priority|physics|branche|science|mathematics|lightning|dispersion|accelerator|detector|terminology|design|operation|foundation|application|prediction|reference|measurement|concept|perspective|overview|position|airplane|symmetry|dimension|toxic|algebra|illustration|classic|verification|citation|unusual|resource|analysis|license|comedy|screenplay|production|release|emphasis|director|trademark|vehicle|aircraft|experiment";
```

- **Submitting your work:**
  - All source files and class files as one tar-gzipped archive.
    - When unzipped, it should create a directory with your ID. Example: **2008CS1001** (NO OTHER FORMAT IS ACCEPTABLE!!! Case sensitive!!!)
  - Source files should include the following: (Case-Sensitive file names!!)
    - `Hangman.pl`
    - Any other supporting or required files
  - ***Negative marks for any problems/errors in running your programs***
  - *If any aspect of the game / rules are confusing, make an assumption and state it clearly in your* **README** *file!*
  - **README** file should also have instructions on how to use/run your program!
  - Submit/Upload it to Moodle