

# Game - Guess the word

---

It is recommended to first Implement the core functionality of the game, then do the full requirements and then add the extra features.

## Core functionality

The basic idea of the game is to guess a word. If we assume there are 2 players playing then the game would go like this:

1. The game selects a word that is unknown to both of the players, and then prints the word with all the letters replaced by `_` (or `*` or whatever character you choose).
2. The first player guesses a letter.
  - If the letter is in the word, the game replaces the `_` with the letter in the correct position, and prints the word with the letter in the correct position. This player also gets a point.
  - If the letter is not in the word, the game prints a message saying that the letter is not in the word.
  - in either case the game prints the word with the letters that have been guessed so far, and the turn goes to the next player.

## Full requirements

- The game should be activated from the command line (using `argparse`, or something similar)
  - The parameters should be path to the file with the word list
  - number of players
- round:
  - In each round, the word as it was guessed so far is being printed
    - each letter that was correctly guessed will appear, all other letters will be hidden as printed as `*` or some other character.
  - the current player gets to guess a letter.
    - If he provides an input that is not a single letter, ask for it again until he provides a single letter
    - If he provides a letter that was guessed before for this word, (correctly or not) ask for it again until he provides a letter that was not guessed for this word.
    - Once he provided a valid input (a single letter that was not guessed before) if this letter exists in the word, this player gets a point. If not he does not.

Please note that the game is **Case Insensitive** meaning that whether the user provides a capital or non-capital letter it is considered the same letter.

- the turn now goes to the next player, and we go through the whole process again (printing the word as it was guessed so far, asking for a letter, and so on).
  - If the word was fully guessed, it is printed with some clear output that it was fully guessed, the computer chooses a new word from some word bank, prints it with all `*` and the game continues from where it left off.
    - All "guessed letters" are obviously erased, as they are relevant on a per-word basis.

- Once all the words in the word bank are guessed, the game ends and the player with the most points wins.

## Extra features

- Instead of using number of players, use names of players (and not **Player 1, Player** etc.)
- The words list should be a list of words the "categories" - so every time a word is shown, it's "category" or "hint" is also shown
- If a player guessed correctly a letter that appears multiple times in the word, he will get amount of points equal to the number of appearances.
- Load the list of words from a json file
- Allow to set the number of words in the game in the command line.
  - The game will end after that amount of words was guessed.
  - If the file has less words than the number of words in the game, let the players know about it and end it immediately.