```
Method: welcome     | Greet user, show rules, ask to continue
***********
Print welcome_message    | Content in notes below
Print "Ready to play?"
Get response             | Y/N
If response == no, then exit
Get username             | eg. "skunkhunt42"
If no username, then username = "Player"
***********
Method: setup            | define variables for a new game
***********
secret_word = "random_word_from_faker_gem"
attempts_left = 7        | attempts allowed before game over
letters_used = []        | empty array to store guesses
progress = "_ _ _ _ _ _" | secret_word letters replaced by "_"
************
Method: get_user_guess
************
Print "Guess a letter: " and progress
Get user_guess           | single alphabetic character
If user_guess is not a-z or letters_used includes user_guess
then
     Print "Needs to be a single letter.  Try again:"
     Get user_guess      | retry for valid input
Append user_guess to letters_used
If secret_word includes user_guess then
     Loop through secret_word characters
          If user_guess == character then
               progress[character.index] = character
     If progress does not include "_" then
          Call end_game("win")
     Else
          Print "So far, you've guessed: " + letters_used
          Call get_user_guess      | Recursive method call
```

```
Else
     attempts_left = attempts_left - 1
     If attempts_left is not > 0 then
        Call end_game("loss")
     Else
        Print "So far, you've guessed: " + letters_used
        Call get_user_guess        | Recursive method call

************
Method: end_game(result)
************
Def end_game(result)
     If result == "win" then
          Print victory_screen     | Content in notes below
     If result == "loss" then
          Print game_over_screen   | Content in notes below
     Print "Play again? (Type yes or no)"
     Get response
     If response == no, then
          Print "Thanks for playing " + username then exit
     Call setup
     Call get_user_guess


*************
Method: yes?(response)
*************
If response is like "yes" or "y"
     Return true
If response is like "no" or "n"
     Return false
Else
     Print "Please enter yes or no:"
     Call yes?(Get response)
```

****
*
Anything written on the right-hand side of a bar (|) is an example or comment. Mostly for reference if something is a bit too cryptic.

The end points for all paths should be either victory, game over or guess again. I think we should use a recursive function/method handle everything from the first get_user_guess. If either victory or game over conditions are met, the method will return the appropriate message. If neither is met, the method will return a call on itself; restarting the method, getting a new guess from the user.

To Do:

- **welcome_message** content
  - Welcome to Hangman!
  - How To Play (how_to_play)
    - Guess what the word is by suggesting one letter at a time.
    - Each correct letter will be revealed in the word.
    - For each incorrect guess, you will lose a life.
    - You only get 7 lives.
  - Are you ready to play? (Y/N)
- **Username** content
  - Please enter your username:
  - Display default username “Player” if nothing entered
- **victory_screen** content
  - Congratulations – You’re on track to having better vocabulary :)
  - Do you want to play again? (Y/N)

- **game_over_screen** content
  - Game Over – Better luck next time :(
  - Do you want to play again? (Y/N)
- **‘welcome’** method
  - Greet the user.
  - Print the game rules (how_to_play).
  - Ask user if they'd like to play Hangman (Y/N).
  - Get their username (or apply default).
- **'setup’** method
  - Generate a random(ish) word from faker for secret_word
  - Assign value to attempts left (7)
  - “ “ “ letters_used ([])
  - “ “ “ progress (“_ “ * secret_word.length)
- **'get_user_guess’** method
  - Ask user to input a single alphabetic character
  - (1) Get and validate the user’s input
  - (2) Handle point calculation (add to progress or lose a life)
  - Repeat (1) and (2) until win-condition is met (i.e. all secret_word letters guessed) or loss-condition is met (i.e. 0 lives left).