此课程: Programming Languages, Part C

Guess The Word

We'll toy a bit with a simple word guessing game (known under the rather morbid name of "Hangman"). Our implementation has a simple text-based interface, and doesn't aim to be fully playable -- it doesn't make much effort to hide the secret word or phrase from the player, for example.

Save the provided code implementing a particularly half-baked version of the game in a file named section-7-provided.rb, put the template solution file nearby, inspect the code, run the game, and play a little. Use the template solution file to implement the changes described below. Note that you should only change the template file.

Change 1: Ignore Punctuation

The first player may enter a phrase or sentence instead of a single word. The current implementation doesn't treat spaces or punctuation marks in any special way. Change the game so that punctuation marks and spaces are not hidden from the second player. You should also reject non-letter characters as guesses.

Change 2: Case Insensitivity

The game currently treats lowercase and uppercase letters as being different. Change that so that entering either a lowercase or an uppercase letter as a guess would uncover all the corresponding letters in the secret word or phrase, regardless of their case.

NOTE: You shouldn't just convert both the secret word and the guesses to lower or upper case -- that's not neat.

Change 3: Forgive Repeated Guesses

Player may try to guess the same letter absent from the secret phrase multiple times. The current implementation will consider all such guesses to be incorrect, and reduce the number of remaining attempts accordingly. Change the game so that repeated guesses are rejected as invalid instead.

EXAMPLE: After implementing all the changes described above, a sample game session could look as follows:

```
Welcome to Guess The Word!
Enter the secret word of phrase:
Alas, poor Yorick!
```

```
Secret word:
----, ---- ------!
9 incorrect guess(es) left.
Enter the letter you want uncovered:
A
Secret word:
A-a-, ---- ------!
9 incorrect guess(es) left.
Enter the letter you want uncovered:
y
Secret word:
A-a-, ---- Y-----!
9 incorrect guess(es) left.
Enter the letter you want uncovered:
a
I'm sorry, but that's not a valid letter.
Secret word:
A-a-, ---- Y-----!
9 incorrect guess(es) left.
Enter the letter you want uncovered:
,
I'm sorry, but that's not a valid letter.
Secret word:
A-a-, ---- Y-----!
9 incorrect guess(es) left.

```

PROVIDED CODE (to save to file section-7-provided.rb**):**

```ruby
## Provided code for Guess The Word practice problem (section 7)

class GuessTheWordGame
  def initialize secret_word_class
    @secret_word_class = secret_word_class
    @game_over = false
    @secret_word_entered = false
    @mistakes_allowed = 9
  end

  def play
    puts "Welcome to Guess The Word!"
    while !@game_over
      tick!
    end
    if @secret_word.is_solved?
      puts "Congratulations, you won."
    else
      puts "Sorry, but you failed to guess the word."
      puts "It was:"
      puts @secret_word.word
    end
    puts "Thank you for playing."
  end

  private

  def tick!
    if @secret_word_entered
      ask_for_guessed_letter
    else
      ask_for_secret_word
    end
  end

  def ask_for_secret_word
    puts "Enter the secret word of phrase:"
    word = gets.chomp
    if is_valid_secret_word? word
      @secret_word_entered = true
    end
  end

  def ask_for_guessed_letter
    puts "Secret word:"
    puts @secret_word.pattern
    puts @mistakes_allowed.to_s + " incorrect guess(es) left."
    puts "Enter the letter you want uncovered:"
    letter = gets.chomp
    if @secret_word.valid_guess? letter
      if !@secret_word.guess_letter! letter
        @mistakes_allowed -= 1
        @game_over = @mistakes_allowed == 0
      else
        @game_over = @secret_word.is_solved?
      end
    else
      puts "I'm sorry, but that's not a valid letter."
    end
  end

  def is_valid_secret_word? word
    @secret_word = @secret_word_class.new word
    !@secret_word.is_solved?
  end
end

class SecretWord
  attr_accessor :word, :pattern

  def initialize word
    self.word = word
    self.pattern = '-' * self.word.length
  end

  def is_solved?
    self.word == self.pattern
  end

  def valid_guess? guess
    guess.length == 1
  end

  def guess_letter! letter
    found = self.word.index letter
    if found
      start = 0
      while ix = self.word.index(letter, start)
        self.pattern[ix] = self.word[ix]
        start = ix + 1
      end
    end
    found
  end
end

```

TEMPLATE SOLUTION (to save to another file):

```ruby
## Solution template for Guess The Word practice problem (section 7)

require_relative './section-7-provided'

class ExtendedGuessTheWordGame < GuessTheWordGame
  ## YOUR CODE HERE
end

class ExtendedSecretWord < SecretWord
  ## YOUR CODE HERE
end

## Change to `false` to run the original game
if true
  ExtendedGuessTheWordGame.new(ExtendedSecretWord).play
else
  GuessTheWordGame.new(SecretWord).play
end

```

□ 完成