

Set de Problemas 3

Introducción

Nota: ¡No te dejes intimidar por este problema! En realidad es más fácil de lo que parece. Te voy a guiar a través de la creación de funciones auxiliares antes de que implementes el juego real.

Para este problema, implementarás una variación del clásico juego de palabras Hangman. Para aquellos de ustedes que no están familiarizados con las reglas, pueden leer todo al respecto aquí: [https://es.wikipedia.org/wiki/Ahorcado_\(juego\)](https://es.wikipedia.org/wiki/Ahorcado_(juego)). En este problema, el segundo jugador siempre será la computadora, quien elegirá una palabra al azar.

En este problema, implementarás una función, llamada *hangman*, que iniciará y llevará a cabo un juego interactivo del ahorcado entre un jugador y la computadora. Antes de llegar a esta función, primero implementaremos algunas funciones de ayuda para que puedas comenzar.

Para este problema, necesitarás los archivos de código `ps3_hangman.py` y `words.txt` (Yo les proporcionaré estos). Haz clic derecho en cada uno y presiona "Guardar enlace como". Asegúrate de guardarlos en el mismo directorio. Abre y ejecuta el archivo `ps3_hangman.py` sin modificarlo para asegurarte de que todo esté configurado correctamente. Por "abrir y ejecutar" me refiero a hacer lo siguiente:

- Ve a tu IDE. En el menú Archivo, selecciona "Abrir".
- Busca el archivo `ps3_hangman.py` y selecciónalo.
- El archivo de plantilla `ps3_hangman.py` ahora debería estar abierto. Ejecuta el archivo.

El código que les he dado carga una lista de palabras de un archivo. Si todo funciona bien, después de un pequeño retraso, deberías ver lo siguiente impreso:

```
Loading word list from file...  
  
55909 words loaded.
```

Si ves un `IOError` en su lugar (por ejemplo, "No existe tal archivo o directorio"), debes cambiar el valor de la constante `WORDLIST_FILENAME` (definida cerca de la parte superior del archivo) al nombre de ruta completo para el archivo `words.txt` (Esto variará en función de dónde guardaste el archivo). Usuarios de Windows, cambien las barras diagonales inversas a barras diagonales, como se muestra a continuación.

Por ejemplo, si guardaste `ps3_hangman.py` y `words.txt` en el directorio `"C:/Users/Ana/"`, cambia la línea: a algo como

```
WORDLIST_FILENAME = "words.txt" a algo como
```

```
WORDLIST_FILENAME = "C:/Users/Ana/words.txt"
```

Esta carpeta variará dependiendo de dónde guardaste los archivos.

El archivo `ps3_hangman.py` tiene una serie de funciones ya implementadas que puedes usar mientras escribes tu solución. Puedes ignorar el código entre los siguientes comentarios, aunque debes leer y comprender cómo usar cada función auxiliar leyendo las cadenas de documentación (docstrings!):

```
# -----  
# Helper code  
# You don't need to understand this helper code,  
# but you will have to know how to use the functions  
# (so be sure to read the docstrings!)  
.  
.  
.  
# (end of helper code)  
# -----
```

Vas a querer hacer toda tu programación para este problema dentro de este archivo también porque estarás escribiendo un programa que depende de cada función que escribes.

Requisitos

Estos son los requisitos para tu juego:

1. La computadora debe seleccionar una palabra al azar de la lista de palabras disponibles que se proporcionó en `words.txt`. Las funciones para cargar la lista de palabras y seleccionar una palabra aleatoria ya se encuentran disponibles en `ps3_hangman.py`.
2. El juego debe ser interactivo; el flujo del juego debe ser el siguiente:
 - Al comienzo del juego, informe al usuario cuántas letras contiene la palabra de la computadora.

- Pide al usuario que suministre un guess(es decir, una letra) por ronda.
- El usuario debe recibir comentarios inmediatamente después de cada guess sobre si su guess aparece en la palabra de la computadora.
- Después de cada ronda, también debes mostrarle al usuario la palabra adivinada parcialmente hasta el momento, así como las letras que el usuario aún no ha adivinado.

3. Algunas reglas adicionales del juego:

- Un usuario tiene permitido 8 intentos. Asegúrate de recordarle al usuario cuántos intentos le quedan después de cada ronda. Asume que los jugadores solo enviarán un carácter a la vez (A-Z).
- Un usuario pierde un guess **solo** cuando adivina incorrectamente.
- Si el usuario adivina la misma letra dos veces, no quites un guess; en su lugar, imprime un mensaje para informarle que ya ha adivinado esa letra y pídele que intente nuevamente.
- El juego debe terminar cuando el usuario construye la palabra completa o se queda sin adivinanzas. Si el jugador se queda sin guesses ("pierde"), revela la palabra al usuario cuando finalice el juego.

Salida de Muestra

El resultado de un juego ganador debería verse así...

```

Loading word list from file...
55900 words loaded.
Welcome to the game, Hangman!
I am thinking of a word that is 4 letters long.
-----
You have 8 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: a
Good guess: _ a _ _
-----
You have 8 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: a

```

```

Oops! You've already guessed that letter: _ a _ _
-----
You have 8 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: s
Oops! That letter is not in my word: _ a _ _
-----
You have 7 guesses left.
Available letters: bcdefghijklmnopqrtuvwxyz
Please guess a letter: t
Good guess: ta_ t
-----
You have 7 guesses left.
Available letters: bcdefghijklmnopqruvwxyz
Please guess a letter: r
Oops! That letter is not in my word: ta_ t
-----
You have 6 guesses left.
Available letters: bcdefghijklmnopquvwxyz
Please guess a letter: m
Oops! That letter is not in my word: ta_ t
-----
You have 5 guesses left.
Available letters: bcdefghijklnopquvwxyz
Please guess a letter: c
Good guess: tact
-----

Congratulations, you won!

```

Y el resultado de un juego perdido debería verse así...

```

Loading word list from file...
55900 words loaded.
Welcome to the game Hangman!
I am thinking of a word that is 4 letters long
-----
You have 8 guesses left
Available Letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: a
Oops! That letter is not in my word: _ _ _ _
-----
You have 7 guesses left

```

```

Available Letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: b
Oops! That letter is not in my word: _ _ _ _
-----
You have 6 guesses left
Available Letters: cdefghijklmnopqrstuvwxyz
Please guess a letter: c
Oops! That letter is not in my word: _ _ _ _
-----
You have 5 guesses left
Available Letters: defghijklmnopqrstuvwxyz
Please guess a letter: d
Oops! That letter is not in my word: _ _ _ _
-----
You have 4 guesses left
Available Letters: efghijklmnopqrstuvwxyz
Please guess a letter: e
Good guess: e_ _ e
-----
You have 4 guesses left
Available Letters: fghijklmnopqrstuvwxyz
Please guess a letter: f
Oops! That letter is not in my word: e_ _ e
-----
You have 3 guesses left
Available Letters: ghijklmnopqrstuvwxyz
Please guess a letter: g
Oops! That letter is not in my word: e_ _ e
-----
You have 2 guesses left
Available Letters: hijklmnopqrstuvwxyz
Please guess a letter: h
Oops! That letter is not in my word: e_ _ e
-----
You have 1 guesses left
Available Letters: ijklmnopqrstuvwxyz
Please guess a letter: i
Oops! That letter is not in my word: e_ _ e
-----

Sorry, you ran out of guesses. The word was else.

```

En las siguientes secciones, dividiré el problema en subáreas lógicas, creando funciones auxiliares que deberás tener para que este juego funcione.