Scrum con Python

Agustín Penagos López, Ayelen Celis Cardona, Emmanuel López, Esneider Giraldo, Keiner Serna

Análisis y Desarrollo de Software, Ficha 2696118. Centro de Diseño y Manufactura del Cuero

Servicio Nacional de Aprendizaje. Itagüí, Regional Antioquia

Instructor, Alexander Zapata

6/08/2024

Emmanuel López Saldarriaga – Scrum Team

**Tarea:** Implementación Inicial del Juego de Ahorcado

**Descripción:**

Me encargué de iniciar el desarrollo del juego de Ahorcado. Comencé en establecer la funcionalidad básica para elegir una palabra aleatoria y gestionar la lógica inicial del juego.

1. **Inicialización del Código:**
   * **Configuración del Entorno:** Implementé la estructura básica del juego, asegurando que el entorno de desarrollo estuviera preparado para la integración de la lógica del juego de Ahorcado.
   * **Importación de Dependencias:** Utilicé la librería random para manejar la selección aleatoria de palabras.



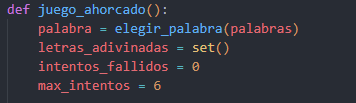
1. **Funcionalidad de Selección de Palabras:**
   * **Definición de Palabras:** Creé una lista de palabras relacionadas con la programación para utilizar en el juego.



* + **Implementación de la Función elegir\_palabra():** Esta función selecciona aleatoriamente una palabra de la lista, utilizando la función choice() de la librería random.



1. **Desarrollo del Juego de Ahorcado:**
   * **Definición de Variables Iniciales:** Inicialicé variables para almacenar la palabra seleccionada, las letras adivinadas, el conteo de intentos fallidos y el número máximo de intentos permitidos.



**Estado Actual:** El código está parcialmente implementado con la selección de palabras y la configuración inicial del juego. La próxima fase incluirá la implementación de la lógica completa del juego, como el manejo de las letras adivinadas, la verificación de intentos y la detección de condiciones de victoria o derrota.

**Nombre:** Keiner Serna

**Rol:** Scrum Team

Fui el encargado del código que proporciona la lógica para manejar los intentos fallidos del jugador en el juego del Ahorcado. Se asegura de que el jugador tenga un número limitado de intentos para adivinar la palabra, actualiza el estado del juego en cada intento y proporciona retroalimentación al jugador sobre el progreso y los intentos restantes. Al final, si el jugador agota todos sus intentos sin adivinar la palabra, se le informa de la pérdida y se muestra la palabra correcta.

- **Mensajes de Bienvenida:**

 print("¡Bienvenido al juego del Ahorcado!")

    print(f"Tienes {max\_intentos} intentos para adivinar la palabra.")

* **Propósito:** Estas líneas imprimen un mensaje de bienvenida al usuario y le informan sobre el número máximo de intentos permitidos para adivinar la palabra.

- **Bucle Principal del Juego:**

  while intentos\_fallidos < max\_intentos:

* print(mostrar\_progreso(palabra, letras\_adivinadas))
* letra = input("Adivina una letra: ").lower()
* letras\_adivinadas.add(letra)
* intentos\_fallidos += 1
* print(f"Letra incorrecta. Te quedan {max\_intentos - intentos\_fallidos} intentos.")
* **Propósito:** Este bucle gestiona la interacción del juego, permitiendo al usuario adivinar letras hasta que se acaben los intentos.
* **Funcionamiento:**
  + Imprime el estado actual de la palabra usando la función mostrar\_progreso, que muestra las letras adivinadas y guiones bajos para las letras restantes.
  + Solicita al usuario que ingrese una letra y la convierte a minúsculas para estandarizar la entrada.
  + Añade la letra adivinada al conjunto letras\_adivinadas.
  + Incrementa el contador de intentos\_fallidos.
  + Imprime un mensaje indicando la cantidad de intentos restantes.

**Condición de Fin del Juego:**

* if intentos\_fallidos == max\_intentos:
* print(f"Lo siento, has perdido. La palabra era: {palabra}")
* **Propósito:** Este bloque de código se ejecuta cuando se ha alcanzado el número máximo de intentos fallidos.

**Funcionamiento:** Informa al usuario que ha perdido el juego y revela la palabra oculta.