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.



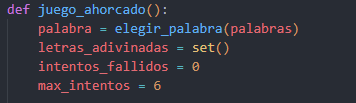
1. **Funcionalidad de Selección de Palabras:**
   * **Definición de Palabras:** Creé una lista de palabras relacionadas con la programación para utilizar en el juego.



* + **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.

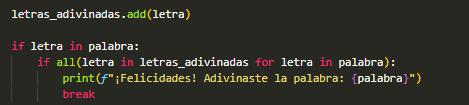
**Nombre:** Ayelen Celis

**Rol:** Scrum Team

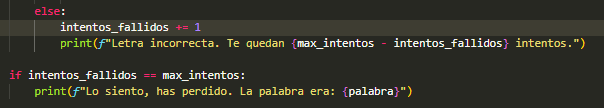
Me encargué de la parte final del código está contando con la función de avisar al jugador cuando adivinó la palabra, decir cuantos intentos quedan en caso de ingresar letras incorrectas y avisar si perdió. Por último, utiliza un llamado a la clase juego\_ahorcado.

**Mensaje:**

Le avisa al jugador que adivinó la palabra con un mensaje y una variable.



Avisa al jugador en caso de equivocarse en una letra sobre la cantidad de intentos disponibles y en caso de perder todos sus intentos disponibles muestra un mensaje diciendo que perdió y la variable con la que puede mostrar la palabra que debía adivinar.



Este código ejecuta la función “juego\_ahorcado”.



Esneider Giraldo González – Product owner

Como Product Owner, mi rol fue coordinar y definir las especificaciones para el desarrollo de un juego simple en Python “Ahorcado”.

**Descripción del Producto**

**Nombre del Producto:** Juego del Ahorcado en Python

**Objetivo:** Desarrollar un juego interactivo de Ahorcado en la consola que permita a los usuarios adivinar una palabra eligiendo letras, con un número limitado de intentos antes de perder el juego.

**Requisitos del Producto**

1. **Selección de Palabras:**
   * Utilizamos una lista predefinida de palabras relacionadas con la programación, por ejemplo, "python", "java", "javascript".
   * El juego debe seleccionar una palabra aleatoriamente de esta lista para cada partida que sea ejecutada.
2. **Mostrar Progreso:**
   * El juego debe mostrar el progreso del jugador revelando las letras adivinadas y manteniendo ocultas las letras no adivinadas como guiones bajos.
3. **Lógica del Juego:**
   * El usuario tiene un número limitado de 6 intentos para adivinar la palabra.
   * Se deben manejar las letras ya adivinadas para evitar repeticiones innecesarias.
   * Se debe notificar al usuario sobre el estado de los intentos restantes y si la letra adivinada es correcta o incorrecta.
   * El juego debe finalizar si el usuario adivina la palabra correctamente o agota todos los intentos.
4. **Interacción del Usuario:**
   * El usuario ingresa letras para adivinar.
   * Se deben mostrar mensajes sobre el estado del juego y el progreso.

**Implementación**

**Funciones Principales:**

1. **elegir\_palabra(lista\_palabras):**
   * Selecciona una palabra aleatoriamente de la lista de palabras proporcionada utilizando la función random.choice.
2. **mostrar\_progreso(palabra, letras\_adivinadas):**
   * Genera una cadena que muestra las letras adivinadas en su lugar correcto y guiones bajos para las letras no adivinadas.
3. **juego\_ahorcado():**
   * Maneja la lógica principal del juego.
   * Inicializa variables como la palabra a adivinar, las letras adivinadas, el contador de intentos fallidos y el máximo de intentos permitidos.
   * Controla el flujo del juego con un bucle que se ejecuta hasta que el usuario gane o pierda.
   * Ofrece retroalimentación al usuario sobre el estado del juego y verifica si ha adivinado la palabra o agotado los intentos.

**Proceso de Desarrollo:**

* Coordiné el diseño del juego, asegurándome de que los requisitos se cumplieran y que la implementación proporcionara una experiencia de usuario clara y funcional.
* Colaboré con los desarrolladores para asegurarme de que el código siguiera las mejores prácticas y estuviera libre de errores.
* Validé la funcionalidad del juego probando diferentes escenarios para garantizar que el juego se comportara como se esperaba.

**Resultados**

El resultado es un juego del Ahorcado funcional que ofrece una experiencia de usuario intuitiva y entretenida. Los jugadores pueden interactuar con el juego a través de la consola, adivinando letras y siguiendo el progreso hasta que ganen o pierdan. La implementación es simple, pero efectiva, y cumple con todos los requisitos establecidos.