INTRODUCCIÓN

1.1 Descripción del juego

El Ahorcado es un juego clásico de adivinanza en el que un jugador debe descubrir una palabra secreta letra por letra, con un número limitado de intentos incorrectos.

Este proyecto presenta una versión digital para un solo jugador, desarrollada en C++ y ejecutada en consola. El jugador dispone de seis intentos antes de perder la partida y gana al adivinar todas las letras de la palabra oculta. El propósito principal es recrear la mecánica original del juego en un entorno interactivo y educativo.

1.2 Objetivos del proyecto

Los objetivos principales son:

- Implementar un sistema completo del juego del Ahorcado en C++.
- Utilizar estructuras de datos fundamentales como arreglos de caracteres.
- Diseñar un sistema robusto de validación de entradas del usuario.
- Aplicar principios de programación modular mediante funciones específicas.
- Ofrecer una interfaz de consola clara y amigable para el usuario.

2. DISEÑO DE LA SOLUCIÓN

2.1 Estructuras de datos utilizadas

El programa utiliza tres arreglos principales:

- Arreglo de palabras predefinidas: contiene diez palabras disponibles para el juego.
- Palabra secreta y estado de la palabra: la primera guarda la palabra seleccionada y la segunda su representación visible con guiones bajos.
- Letras intentadas: almacena las letras ingresadas para evitar repeticiones

.

Además, se manejan variables de control que gestionan la cantidad de intentos, la detección de victoria y la longitud de la palabra actual.

2.2 Justificación del uso de arreglos de caracteres

Se eligieron arreglos de caracteres en lugar de cadenas tipo string para reforzar la comprensión de la gestión de memoria y el uso del terminador nulo '\0'. Este enfoque permite manipular los datos de forma directa y eficiente, controlando cada posición del arreglo sin dependencias de clases externas.

2.3 Flujo principal del programa

El juego se organiza en tres fases:

- Inicialización: se selecciona aleatoriamente una palabra, se calcula su longitud y se llena el estado visible con guiones bajos.
- Bucle principal: se muestra el progreso, se validan las letras ingresadas y se actualiza la palabra. Si el jugador acierta todas las letras, gana; si agota los intentos, pierde.
- Finalización: el programa muestra el resultado y ofrece la opción de jugar nuevamente sin reiniciar el sistema.

3. IMPLEMENTACIÓN

3.1 Lógica de verificación de letras

Cada letra ingresada se almacena en un registro y se compara con la palabra secreta. Si coincide, se actualiza el estado visible; de lo contrario, se reduce el número de intentos restantes.

El algoritmo revisa todas las posiciones de la palabra, lo que permite identificar letras repetidas sin errores.

3.2 Validación de entrada

El sistema de validación evita tres tipos de errores comunes:

- Números o símbolos: solo se permiten letras del alfabeto.
- Entradas múltiples: se acepta una letra por turno.
- Repetición de letras: el programa informa si una letra ya fue utilizada.
- Esto garantiza que el usuario reciba retroalimentación inmediata y mantenga una interacción fluida con el juego.

3.3 Estrategia para evitar letras repetidas

El programa mantiene un historial de letras utilizadas y realiza una búsqueda preventiva antes de procesar cada intento.

Gracias a esto, no se penaliza al jugador por repetir una letra y se conserva la integridad del contador de intentos.

4. CONCLUSIONES

4.1 Desafíos enfrentados

Durante el desarrollo se presentaron varios retos técnicos, como el manejo correcto del terminador nulo '\0', la sincronización entre los arreglos de palabra secreta y palabra visible, y la creación de un sistema de validación capaz de gestionar entradas inválidas y repetidas.

También fue necesario garantizar que el algoritmo de victoria detectara de forma precisa cuando el jugador había completado la palabra.

4.2 Soluciones implementadas

El programa se estructuró de forma modular, con funciones independientes que facilitan la comprensión y el mantenimiento.

Se usaron variables globales de control para manejar el estado del juego, y una función específica permite reiniciar las partidas sin cerrar el programa.

Además, se empleó la conversión automática a mayúsculas para unificar el formato de entrada y evitar errores de comparación.

4.3 Aprendizajes clave

Este proyecto permitió reforzar conocimientos sobre:

- Diseño y manipulación de arreglos de caracteres.
- Validación de datos y control de errores en tiempo de ejecución.
- Programación modular y estructurada en C++.
- Importancia de planificar antes de implementar la lógica principal.

En conclusión, la implementación del juego del Ahorcado cumple con los requerimientos técnicos, ofrece una interacción clara y demuestra el dominio de conceptos fundamentales de programación en C++.