DOCUMENTO DE ANÁLISIS DEL PROYECTO

Juegos: Tatetí-Ahorcado

6 DE JULIO DE 2021

SOFÍA SANABRIA Y MANUEL BUSLÓN

Índice

1 Planteamiento del problema

Se parte de la propuesta de crear un programa que permita jugar al ta-te-ti entre dos personas y el juego del ahorcado, el cual a su vez planteaba más condiciones, como poseer palabras en una lista que se seleccionaran de forma aleatoria y tener un número límite de intentos, además de a su vez imprimir en pantalla el estado actual. El primer juego, ta-te-ti, trae consigo la tarea de ir alternando el turno entre los jugadores, realizar la verificación de si es posible colocar la ficha o si no, también el punto más importante es la condición de victoria, tres en línea, esta tiene que verificarse siempre que un jugador coloque una ficha nueva, para luego si se da un resultado del juego avisar de este y que el usuario tenga una respuesta. El otro juego, ahorcado, implica la necesidad de implementar de alguna forma una lista sobre la cual se seleccionen, decir el largo de la palabra seleccionada e imprimir cuando se descubra una letra los lugares en los que aparece, tanto como los lugares en los cuales aún están ocultas las letras. Se puede dar la opción de arriesgar a decir la palabra entera, o intentar con las letras una por una, y se considera algo útil el poder conocer las letras incorrectas que ya se ingresaron. Al terminar el juego este debe decir en cuantos intentos se logro, y si se perdió, la palabra que se necesitaba adivinar.

2 Análisis de soluciones alternativas

2.1 Ahorcado

2.1.1 Para el problema de la lista de palabras

- Para solucionar el problema se puede implementar directamente una lista de palabras sobre la cual el programa elija aleatoriamente una, otra forma de hacerlo podría ser implementar varias listas permitiendo elegir al usuario que lista de palabras quiere usar y a partir de ella elegir una.
- Otras formas tendrían en cuenta el uso de archivos de texto con las palabras, en estos casos una forma sería utilizar etiquetas que definan una sección de palabras que el programa cargará en una lista o tener una carpeta que contenga los archivos de texto representando las categorías con sus palabras dentro.

2.1.2 Para esconder la palabra

- Como primera opción a partir de la palabra seleccionada se puede imprimir el carácter que lo esconde por la cantidad de letras que tenga la palabra, para luego ir reemplazando los caracteres por la letra acertada en su respectivo lugar.
- Una solución diferente sería almacenar las letras de la palabra y las letras descubiertas en listas, para luego reemplazar las letras no descubiertas de las palabras por el carácter que las esconde.
- Para solucionar el problema de colocar el guion bajo y que este se confunda formando solo una raya, se coloca un espacio entre cada letra.

2.1.3 Para imprimir el estado de la horca

- Se puede plantear imprimir línea por línea el dibujo de la horca y agregar cada parte del cuerpo a las líneas que correspondan dependiendo de la cantidad de errores.
- Una manera un poco extraña sería imprimir texto con el estado de la horca, el problema de esta opción es que es algo perturbadora e incomoda de imaginar al pensar que describe que aparece cada parte con los errores.
- Otra opción es almacenar cada parte del cuerpo en una lista y agregarlas a otra a medida que se van obteniendo errores.

2.2 Ta Te Ti

2.2.1 Guardar el estado actual

- La forma en la que se guardará el estado del tablero puede estar representada de distintas formas, como: una lista con la ficha que pertenece a esa posición, otra lista que tenga tres elementos y a su vez otros tres elementos.
- Un diccionario que tenga como claves las coordenadas en el tablero y como valor el elemento que está en ese lugar.

2.2.2 Finalizar partida

- Para ello se puede recorrer cada una de las filas, columnas y diagonales para ver si coinciden o es un empate.
- Comparar directamente los valores de las posiciones que se necesitan para ganar y si ninguna de estas coincide y no están vacíos es un empate.

3 Justificación de la solución elegida

Debido al tiempo que se tenía para la realización del proyecto, varias opciones se vieron limitadas; existía la idea de realizarlo con interfaz gráfica, pero no era posible en el tiempo que se tenía.

3.1 Ahorcado

- Se optó por hacer una carpeta con archivos de texto con cada categoría como nombre de este, es más limpia en cuanto a código y organización, ya que cuando aumenta el número de palabras disminuye la legibilidad del código.
- La forma elegida de ocultar la palabra es óptima en cuanto a eficiencia, ya que coloca un guion bajo por cada letra oculta junto a un espacio para evitar confusiones a la hora de ver la ubicación de las letras.
- Elegimos la opción de tener la horca prediseñada para ir agregando cada parte del cuerpo según cuántos errores se cometen, las partes del cuerpo están en una lista.

3.2 Ta Te Ti

- Se eligió guardarlo en un diccionario ya que es más fácil para acceder de esta manera.
- Para saber si la partida finalizó se recorre y compara cada casilla utilizando estructuras de repetición.

4 Desarrollo de la solución

4.1 Launcher

El launcher del programa funciona como un menú en el cual el usuario se puede loguear para después acceder a los juegos sin perder sus estadísticas.

Para hacer esto, utiliza un archivo en el cual se guarda el estado actual de una lista de usuarios. Este se carga al iniciar el programa y se actualiza cada vez que cambia una estadística o se agrega un usuario.

Para ejecutar los juegos se importa el archivo .py y se llama a su función principal para que comience el juego.

4.2 Ahorcado

Para implementar la solución se creó una carpeta que el programa establece como base para leer los archivos que contiene, de los cuales tomará el nombre como categoría y su contenido como palabras. En el código se utiliza un bucle for, este recorrerá todos los archivos en esta carpeta agregando a un diccionario el nombre como clave y una lista de las palabras como valor.

Para la función que imprime la horca se reciben los errores, letras encontradas y la palabra a esconder, lo primero es reemplazar todas las letras que aún no han sido descubiertas por un guion bajo, luego formatea una cadena que tiene la figura de la horca con la cantidad de partes del cuerpo equivalente al número de errores.

4.3 Ta Te Ti

Este juego implementa un diccionario que utiliza de claves las coordenadas desde A hasta C, seguidas del número del uno al tres. El tablero se inicializa con un espacio vacío en cada valor y luego a medida que el jugador va colocando fichas compara si

hay una repetición de tres fichas vertical, horizontal o diagonalmente, y si estas fichas no contienen un espacio en blanco. Si lo anterior no se cumple, no hay un usuario ganador y compara si el tablero está lleno, lo que significaría un empate.

5 Conclusiones

El trabajo resultó interesante, aunque suponía algunos retos debido a la solución elegida, también algunas complicaciones al querer realizar más cosas de las que se tenía conocimiento en el momento, por lo que adaptarlo con lo que se contaba fueron varios intentos fallidos.

El resultado final fue agradable, puede tener muchas soluciones pero con las herramientas que se contaban es lo que mejor se nos ocurrió en el tiempo disponible.