

Trabajo final WORDLE

Andrés Guido Grillo

Documentacion de ejercicio

• Objetivo	2
• Práctica C	3
• Estructura de datos	4
• Algoritmos	5
• Compilacion y ejecucion	6-7
• Mejoras futuras	8
• Opiniones sobre la materia y comentarios sobre el código	9

Objetivo

El objeto del ejercicio es programar el juego **WORDLE**.

El juego consiste en un tablero con espacio para poder poner un máximo de 5 letras en donde el jugador deberá intentar adivinar una palabra oculta escribiendo un intento en dicho tablero.

Dependiendo de la posición individual de cada letra se le comunicara al jugador si dicha letra está bien posicionada, si pertenece a la palabra o si directamente la letra no está incluida en la palabra secreta. La forma en la que se le comunica al jugador que letras son correctas es de la manera siguiente:

- Si la letra ingresada está en la posición correcta debajo de dicha letra aparecerá el caracter "V"
- Si la letra ingresada pertenece a la palabra pero no está en la posición correcta, debajo de la letra aparecerá el caracter "-"
- Si la letra no pertenece a la palabra entonces aparecerá el caracter "_"

El jugador tiene un máximo de 5 intentos por cada palabra y se le asignará un puntaje en función de la cantidad de intentos, la cantidad de letras acertadas y si se logró descubrir la palabra secreta o nó, siendo el mayor puntaje posible de obtener 10000(diez mil) puntos. La meta del juego es tratar de obtener la mayor cantidad de puntos.

Práctica C

El programa consiste en tres archivos:

- **constantes.h**
con constantes simbólicas, typedefs y estructura de datos
- **funciones.h**
Archivo de cabecera con la declaración de todas las funciones empleadas
- **TrabajoFinal.c**
Incluye constantes.h y funciones.h
Define todas las funciones que implementan el juego

Total de líneas del programa: 230

Estructura de datos

Se emplea para poder almacenar la información de cada partida un array de tipo Resultado (una estructura que contiene un array de caracteres, dos datos de tipo numérico), para cada sesión de juego se utiliza un array bidimensional de caracteres, para mostrar el resultado de cada intento un array de caracteres y un array que almacena la aparición de las letras.

Las variables que se emplean son:

- **RePartidas:** Es donde se recompilan las partidas jugadas con puntaje y la palabra jugada y es lo que se utiliza para mostrar los puntajes obtenidos. Los valores posibles para cada campo de las estructuras son:

- El array que contiene la palabra
- El puntaje de dicha partida
- Un valor que puede ser 1 o 0 para representar si se ganó o no la partida

- **intento:** Es una matriz bidimensional que almacena cada intento del jugador, en esta matriz solo se almacenan caracteres alfabéticos

- **evaluacion:** Es lo que se muestra para comunicarle que tan acertado fue el intento al jugador.

Los valores posibles para cada campo del array son:

- Un carácter 'V' que representa una letra bien colocada
- Un carácter '-' que representa una letra que pertenece a la palabra pero no está bien colocada
- Un carácter '_' que representa una letra que no pertenece a la palabra

Al inicio se carga totalmente con el valor '_'

- **comoVa:** Es un array que almacena en cada espacio el estado de la letra en la posición n, el array admite los siguientes valores:

- Un carácter 'V' significa que la letra ya fue colocada correctamente
- Un carácter '-' significa que la letra ya apareció pero en una posición incorrecta
- Un carácter '_' significa que la letra todavía no apareció

Algoritmos

Los pasos generales del programa son los siguientes:

- Preguntar cuantas partidas piensa jugar el usuario
- Empieza la sesion
- Bucle principal
 - Elegir una palabra del archivo "palabras.txt"
 - Pedir el intento al jugador
 - Convertir todas las letras ingresadas en mayúsculas
 - Calcular el puntaje y marcar las letras que salieron
 - Si acerto, muestra el mensaje y termina la partida
 - Si no acerto y la cantidad de intentos es 5, termina la partida, en caso contrario repite 'Pedir intento'
 - Si la cantidad de partidas jugadas es menor a la cantidad pedida por el jugador, pregunta al jugador si quiere seguir jugando
 - Si quiere seguir jugando, suma 1 a la cantidad de partidas jugadas y repite
 - Si no quiere seguir jugando, suma la cantidad de partidas pedidas al total de partidas jugadas
 - Si la cantidad de partidas jugadas es mayor a la cantidad pedida termina la sesion
- Restarle a la cantidad de partidas jugadas el total de partidas que quería jugar el usuario
- Si la el resultado de esa resta es 0 entonces se jugó todas las que el usuario pidió y se va a mostrar esa cantidad de partidas, en caso contrario se va a mostrar el resultado de la suma
- Se muestran todas las partidas jugadas con el puntaje obtenido en cada una de ellas
- Se muestra la partida con el puntaje más alto y la partida con el puntaje más bajo
- Se muestra el promedio de puntajes en donde se logró una victoria
- Se libera la memoria utilizada

Compilación y ejecución

Para compilar el programa se deben realizar las siguientes acciones:

1. Descomprimir en un directorio el archivo **WORDLE.zip**
2. Acceder al directorio generado **# cd WORDLE**
3. Compile
gcc -Wall TrabajoFinal.c -o WORDLE
4. Ejecute
./WORDLE

Ingrese la cantidad de veces que va a jugar (MAX 8 veces):

>3

Se van a jugar un total de 3 partidas.

Partida numero: 1 de 3

>gatos

>---

>sonar

>v---

>sabio

>v---v

>bueno

>_vv_v

>suelo

>vvvvv

Acerto!!!

Seguir jugando?(Y/N)

>Y

Partida numero: 2 de 3 >gotas

>vvvvv

Acerto!!!

Seguir jugando?(Y/N)

>N

Partida Nro 1

Palabra:SUELO

Puntaje:5600

Partida Nro 2

Palabra:GOTAS

Puntaje:10000

Presione ENTER para continuar...

ANALISTA UNIVERSITARIO EN SISTEMAS TALLER DE PROGRAMACION I

La/s mejor/es partida/s:
Partida Nro 2
Palabra:GOTAS
Puntaje:10000

La/s peor/es partida/s:
Partida Nro 1
Palabra:SUELO
Puntaje:5600

El puntaje promedio de las partidas ganadas es 7800.00
Presione ENTER para finalizar.

En este ejemplo el usuario adivinó dos palabras y decidió no continuar jugando

Mejoras futuras

En esta sección haré algunas propuestas que podrían mejorar la experiencia del usuario.

- Se podría hacer uso de memoria dinámica para poder aumentar la cantidad de jugadas posibles
- Se podría mejorar la interfaz gráfica
- Añadir un archivo que guarde las mejores sesiones de todos los tiempos para poder hacer un ranking

Opiniones sobre la materia y comentarios sobre el código

Quisiera hacer un par de comentarios al respecto al desarrollo del trabajo práctico y mi opinión sobre la materia.

Sobre la materia en sí no tengo ningún tipo de queja en particular ya que el nivel de los contenidos y la explicaciones que se fueron dando sobre los mismos no impidió en ningún momento la posibilidad de aprender.

Una cosa que quiero agradecer es la decisión de tomar un trabajo práctico en lugar de únicamente un examen para evaluar la materia ya que no solo me resulta más cómodo, sino que me permite siempre poder indagar más a la hora de realizar el código y hacer cosas que me resulten más prolijas, interesantes y en general mejores.

Con respecto al trabajo práctico me resultó muy aportativo para poder comenzar a utilizar herramientas que venía haciendo tiempo queriendo usar y que por falta de un proyecto y motivación no había utilizado hasta ahora. Siendo estas herramientas: Debian (lo instalé en una notebook que tenía sin usar hace mucho tiempo por ser obsoleta), VIM el editor de texto para programar, git (en este caso github) y LaTeX (herramienta con la que estoy escribiendo este informe). Quiero aclarar que cualquier tilde que se me haya olvidado hacer es porque el teclado que utilizo no me deja hacerlas cómodamente por lo que hubo veces en las que las dejé para otro momento y me olvidé de colocarlas/