Práctica curso 2017/18

Laberinto

La práctica de este curso consiste en recorrer un laberinto.
Este laberinto será proporcionado en un fichero de texto “laberinto(p/i/v).txt” que se tendrá que leer para poder construirlo y mostrarlo al usuario.

Recorrer el laberinto tendrá diferentes objetivos:

- Capturar la mayor cantidad de números pares y conseguir llegar a la salida con la máxima puntuación y sin cometer errores.
- Capturar la mayor cantidad de números impares y conseguir llegar a la salida con la máxima puntuación y sin cometer errores.
- Capturar la mayor cantidad de vocales y conseguir llegar a la salida con la máxima puntuación y sin cometer errores.

Se presentará al usuario un menú con las opciones del juego siguientes:

1. Usuario Nuevo (Registrarse).
2. Usuario Existente.
3. Caza de números pares.
4. Caza de números impares.
5. Caza de vocales.
6. Mostrar fichero de jugadores.
7. Salir

Para empezar a jugar, el jugador si es la primera vez que juega, deberá registrarse seleccionando la opción del menú “1. Usuario Nuevo”, y deberá introducir la siguiente información:

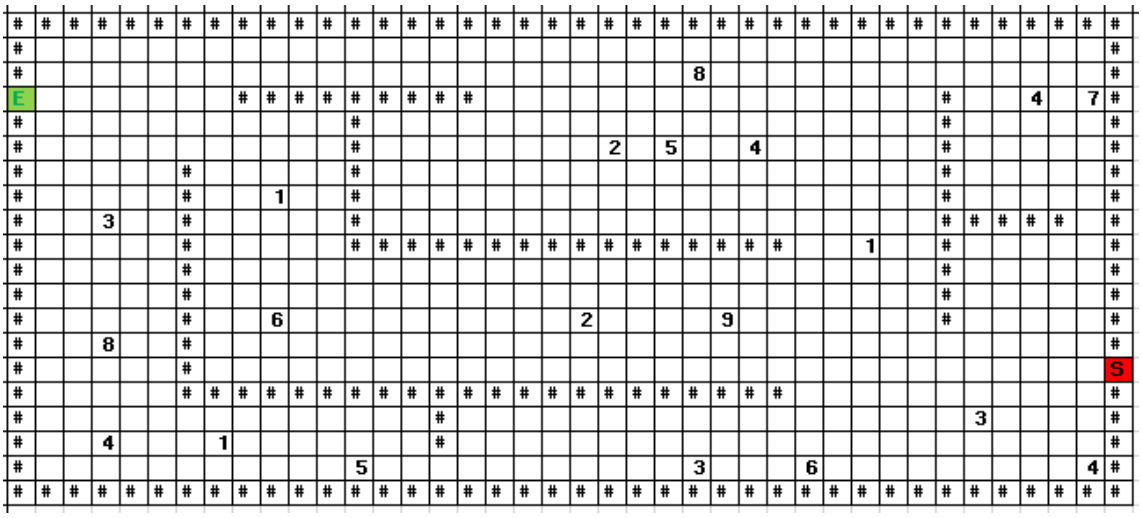
- **Nick** que no exista en el fichero de jugadores.txt (10 caracteres máximo)
- **Nombre y apellidos** (50 caracteres máximo, sin espacios en blanco)
- **Nacionalidad**: **Esp** (Española)/ **Ext** (Extranjera)
- **Edad**
- Puntuación máxima obtenida (esta información no la deberá introducir el jugador, se calculará después de cada partida)

Si el usuario ya está registrado, deberá entrar por la opción del menú “2. Usuario existente” e introducir su Nick, si existe deberá salir por pantalla su nombre y apellidos y la puntuación máxima obtenida, en caso de no existir se deberá sacar por pantalla un mensaje “*Este usuario no existe, debe registrarse primero*”.

Una vez registrado el jugador, podrá seleccionar el juego que desee, dependiendo de la opción elegida se cargará el fichero oportuno:

- **Laberintop.txt** (juego de números pares)
- **Laberintoi.txt** (juego de números impares)
- **Laberintov.txt** (juego de vocales)

Ejemplo de laberinto (par e impar)



#: Pared E: entrada S: salida

El laberinto **sólo** tendrá una entrada y una salida. Es una matriz de 40 columnas x 20 filas.
El usuario deberá recorrer el laberinto de la siguiente forma:

- Usará las teclas de desplazamiento para moverse (←↑→↓) por el laberinto y las teclas:
  - F1 para saltar derecha
  - F2 saltar izquierda
  - F3 saltar arriba
  - F4 saltar abajo.
- Si encuentra un valor correcto y pasa por encima (←↑→↓) se sumarán 10 puntos.
- Si pasa por encima de un valor incorrecto se restarán 10 puntos, los valores incorrectos deberá saltarlos con las teclas F1, F2, F3 o F4, dependiendo del sentido hacia el que desee moverse.
- Si el jugador no tiene puntos y pasa por un valor incorrecto, perderá la partida (devolviendo -10).
- Si choca contra pared no se moverá.
- No puede salir de los límites del laberinto.

Al terminar la partida, se guardará la puntuación obtenida considerando que:

- Si es la primera vez que juega, se guardará en el fichero jugadores.txt dicha puntuación, junto con sus datos.
- Si no es la primera vez que juega, se comparará con la puntuación máxima que tenga, si es mayor la conseguida, se actualizará su puntuación máxima.

****El programa, se podrá ejecutar tantas veces como desee el jugador.*

Opcional:

- Uso limitado de las teclas de salto.
- Penalización si toca pared o salta y no se puede completar la acción.

**** Todas las mejoras que el alumno realice se tendrán en cuenta para la nota final (colores, sonidos, extras…)**

Cada estudiante deberá seguir las siguientes pautas:

1.- Realizar un estudio previo de como resolvería el problema. En un papel y a lápiz poner a groso modo el funcionamiento del programa. Me deberéis mandar una foto del papel donde lo hagáis.

2.- Constantes y tipos de Datos

¿Qué constantes y tipos de datos se van a definir y por qué?

3.- Módulos propuestos

Recomendaciones: Cada vez que se programe un módulo hay que probar su correcto funcionamiento, una vez comprobado y verificado, se pasará a programar el siguiente módulo.

Menu: Módulo que muestra las opciones del menú y devolverá la opción correcta que elija el usuario.

1. Usuario Nuevo (Registrarse).
2. Usuario Existente.
3. Caza de números pares.
4. Caza de números impares.
5. Caza de vocales.
6. Mostrar fichero jugadores
7. Salir

Elija una opción:

Prototipo: ***int Menu(void);***

Usuario_Nuevo: Módulo en el que se leerá por teclado el Nick, se comparará con los existentes en el fichero jugadores.txt, si existe, se visualizará mensaje "*Nick existente, introduzca otro*", en caso contrario se permitirá que introduzca el resto de los campos (nombre y apellidos, nacionalidad y edad) y se guardarán en el registro jug de tipo jugador o se devolverá al programa principal.

Prototipos permitidos:

void Usuario_Nuevo (ifstream * jugadores, jugador & jug);

jugador Usuario_Nuevo (ifstream * jugadores);

Usuario_Existente: Módulo en el que se leerá por teclado el Nick, se comparará con los existentes en el fichero jugadores.txt, si no existe, se visualizará mensaje "*Nick inexistente, introduzca otro*", en caso contrario se mostrará en pantalla su nombre y apellidos y puntuación máxima obtenida.

void Usuario_Existente (ifstream * jugadores, jugador & jug);

jugador Usuario_Existente (ifstream * jugadores);

Laberinto_Numeros_Pares: Módulo que deberá leer el archivo "Laberintop.txt", construir la matriz con la información leída (laberinto l) y mostrar por pantalla el laberinto propuesto.

Prototipo: ***void Laberinto_Numeros_Pares (ifstream * fich, laberinto l);***

Jugar_Lab_Pares: Módulo al que se le pasará el laberinto, se colocará al jugador en la entrada y comenzará a jugar, se sumarán 10 puntos por cada número par que consiga (borrándose del laberinto) y se restarán 10 puntos por cada número impar. El módulo devolverá el total de puntos conseguidos por el jugador. La partida terminará cuando llegue a la salida, cuando se quede en negativo o si pulsa Esc (en este caso no se guardará nada).

El módulo devolverá los puntos conseguidos, si devuelve -10 significará que ha perdido y si devuelve -1 significará que ha pulsado Esc.

Prototipo: ***int Jugar_Lab_Pares (laberinto l);***

Actualizar_Fichero: Módulo al que se le pasará el registro del jugador y los puntos conseguidos (siempre que sean positivos), se deberá actualizar el fichero "jugadores.txt".

Si es jugador nuevo y tiene puntos positivos, se creará un nuevo registro en el fichero, en caso de un jugador existente, sólo se actualizará si la puntuación obtenida es mayor que su máxima puntuación.

Prototipo: ***void Actualizar_Fichero (jugador jug, int puntos, ofstream & fich);***

Mostrar_Fichero: Módulo al que se le pasará un parámetro de tipo fichero de entrada, y se mostrarán los datos grabados en el fichero "jugadores.txt".

Prototipo: ***void Mostrar_Fichero (ifstream & fich);***

Nota: Esta práctica puede estar sujeta a cambios, que la profesora notificaría en clase o en el campus virtual de la asignatura.