

Proyecto Semestre 2 LS Racing



ÍNDEX

1 Enunciado	3
1.1 Objetivos	4
2 Funcionamiento	5
2.1 Opción 1: Configuración del coche	7
2.1.1 Panel de configuración: Pieza	10
2.1.2 Panel de configuración: Configuración del coche	11
2.2 Opción 2: Carrera	11
2.3 Opción 3: Clasificación	18
2.4 Opción 4: Guardar temporada	21
2.5 Ficheros	21
2.5.1 Fichero de piezas	21
2.5.2 Fichero de GPs	22
2.5.3 Fichero de corredores	23
2.5.4 Fichero base	24
2.6 Diseño	24
2.7 Opcionales	25
3 Consideraciones	26
4 Requisitos mínimos	27
5 Normativa general	28
6 Contenido de la memoria	29
7 Fechas de entrega	30

1 ENUNCIADO

Gracias al buen trabajo de los alumnos de PGM1 con el programa LS Strategist, *Scuderia Ferrari* se llevó el campeonato de constructores y sus dos pilotos quedaron primero y segundo en la clasificación de la temporada.

Debido a este éxito, *Ferrari* nos puso en contacto con la FIA, que buscaban a alguien para desarrollar un nuevo videojuego de simulación de Formula 1.

Hemos recibido el encargo de crear un nuevo juego, llamado LS Racing, que permita al jugador disputar una temporada completa de F1 contra otros jugadores y alzarse con el premio.

El juego debe permitir, a parte de correr los GPs, configurar el coche con diferentes piezas para adaptarse lo mejor posible al circuito que toque correr, así como poder ver y guardar la clasificación de las carreras.

De momento, la FIA nos ha pedido que tengamos una versión no online del juego, por lo que en vez de jugar contra otras personas, nuestros oponentes serán *bots*.

1.1 OBJETIVOS

Esta práctica ha estado diseñada con la intención de aplicar los siguientes conceptos y objetivos:

- Tratamiento de tanto fichero de texto como binarios.
- Trabajar con el *clock* del ordenador mediante la librería *time.h*.
- Uso de memoria dinámica.
- Estructurar el código en módulos.
- Implementar y utilizar correctamente estructura de datos como colas o listas.
- Trabajar con librerías externas, en este caso, Allegro.
- Aprender a pasar argumentos al main.
- Trabajar con un IDE más complejo que vim, CLion.

2 FUNCIONAMIENTO

Al ejecutar nuestro programa, lo primero que deberemos hacer es leer cuatro ficheros, los cuales contendrán los datos necesarios para el correcto funcionamiento del programa. Los ficheros son los siguientes:

- Fichero de piezas: fichero de texto, con extensión *.txt*. Contiene la información de todas las piezas disponibles para la configuración del coche. Se tiene que guardar la información en **arrays dinámicos**.
- Fichero de GPs: fichero de texto, con extensión *.txt*. Contiene la información de todos los grandes premios. Éstos se encuentran desordenados, por lo que será necesario utilizar una **lista ordenada** para guardar su información.
- Fichero de corredores: fichero binario de extensión *.bin*. Contiene la información de todos los corredores, a excepción del nuestro. Se tiene que guardar la información en un **array dinámico**.
- Fichero base: fichero binario de extensión *.bin* que contiene las estadísticas base de los coches.

El nombre de los ficheros es libre, lo único que es obligatorio es utilizar la extensión indicada. Dado que el nombre de un fichero puede variar, tenemos que indicarle a nuestro programa qué ficheros tiene que leer. Para esto, le pasaremos como argumentos del programa los cuatro ficheros, en el mismo orden que la explicación, es decir, primero el de piezas y por último el fichero base.

Por lo tanto, tendremos que comprobar que se hayan pasado 4 argumentos, en caso de que falte algún argumento, o haya de más, habrá que mostrar un mensaje de error informativo y finalizar la ejecución del programa.

Error. El programa tiene que recibir 4 argumentos.

Output 1: Error en el numero de argumentos

En caso de que se encuentren todos los argumentos, deberemos proceder a leer los cuatro ficheros y guardar sus datos. Si durante este proceso hay cualquier error, o el fichero no existe, deberemos mostrar un mensaje de error informativo y finalizar la ejecución del programa.

Error. Ha ocurrido un error durante el procesamiento de los ficheros.

Output 2: Error durante el procesamiento de ficheros

También tenemos que comprobar que los ficheros no estén vacíos.

Error. Hay un fichero vacío

Output 3: Fichero vacío

Si hemos podido conseguir toda la información de forma satisfactoria, mostraremos el mensaje de bienvenida y el menú de la aplicación.

```
Bienvenidos a LS Racing!
```

- ```
1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada
```

```
Que opcion quieres ejecutar?
```

Output 4: Menú

El menú se deberá volver a mostrar después de ejecutar cualquiera de las opciones. También se deberá volver a mostrar en caso de introducir una opción no válida, junto a su correspondiente mensaje de error.

```
Bienvenidos a LS Racing!
```

- ```
1. Configurar coche  
2. Carrera  
3. Ver clasificacion  
4. Guardar temporada
```

```
Que opcion quieres ejecutar? 1
```

```
[ ... ]
```

- ```
1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada
```

```
Que opcion quieres ejecutar? 1
```

Output 5: Repetición del menú

En las entradas de información al menú, igual que en todas las otras entradas por teclado de la aplicación, se os puede introducir una cadena de texto, por lo que el programa deberá aceptarlo.

```
Bienvenidos a LS Racing!
```

- ```
1. Configurar coche  
2. Carrera  
3. Ver clasificacion  
4. Guardar temporada
```

```
Que opcion quieres ejecutar? hfjdoajdjdjd
```

```
Error. Opcion no valida.
```

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

```
Que opcion quieres ejecutar?
```

Output 6: Opción errónea

Para finalizar la ejecución del programa, deberemos introducir *exit*. La entrada es *case sensitive*, por lo que si no está todo en minúsculas, no se finalizará el programa.

```
Bienvenidos a LS Racing!
```

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

```
Que opcion quieres ejecutar? exit
```

```
Hasta pronto!
```

Output 7: Finalizar la ejecución del programa

Todos estos mensajes se tienen que mostrar por línea de comandos, no utilizando Allegro.

2.1 OPCIÓN 1: CONFIGURACIÓN DEL COCHE

Al seleccionar la opción 1, deberemos cargar la pantalla de configuración del coche.

Antes de esto, pediremos la información del piloto:

- Nombre del piloto: cadena de caracteres de hasta 25 caracteres que indica el nombre del piloto.
- Nombre de la escudería: cadena de caracteres de hasta 25 caracteres que indica el nombre de la escudería.
- Dorsal: entero que indica el dorsal del piloto. Puede tener un valor de entre 1 y 99, ambos incluidos. Cualquier otro valor será considerado incorrecto y se volverá a pedir el dorsal de nuevo. Recordad que se os puede introducir una cadena de caracteres.

```
Bienvenidos a LS Racing!
```

- ```
1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada
```

```
Que opcion quieres ejecutar? 1
```

```
Nombre del piloto? Kasill Ego
```

```
Nombre de la escuderia? Scuderia Perrari
```

```
Dorsal? 100
```

```
Error, el dorsal tiene que ser un entero entre 1 y 99
```

```
Dorsal? Tres
```

```
Error, el dorsal tiene que ser un entero entre 1 y 99
```

```
Dorsal? 3
```

```
[...]
```

```
Cargando configurador ...
```

Output 8: Ejecución de la opción 1

También pediremos las cuatro estadísticas del piloto:

- Reflejos: entero entre 0 y 10, ambos incluidos, que representa la capacidad de reacción del piloto.
- Condición física: entero entre 0 y 10, ambos incluidos, que representa la condición física del piloto.
- Temperamento: entero entre 0 y 10, ambos incluidos, que representa el temperamento del piloto.
- Gestión de neumáticos: entero entre 0 y 10, ambos incluidos, que representa la técnica de conducción para aprovechar los neumáticos del piloto.

Hay que tener el mismo control de errores utilizado para el dorsal.

```
Bienvenidos a LS Racing!
```

- ```
1. Configurar coche  
2. Carrera  
3. Ver clasificacion
```



```
4. Guardar temporada

Que opcion quieres ejecutar? 1

Nombre del piloto? Kasill Ego
Nombre de la escuderia? Scuderia Perrari
Dorsal? 3

Reflejos? 8
Condicion fisica? 11

Error, la condicion fisica tiene que ser un entero entre 0 y 10, incluidos.

Condicion fisica? 9

Temperamento? afghj

Error, el temperamento tiene que ser un entero entre 0 y 10, incluidos.

Temperamento? 2
Gestion de neumaticos? 7

Cargando configurador ...
```

Output 9: Ejecución de la opción 1

Toda esta información del piloto solo de deberá pedir la primera vez que pulsemos la opción 1, si volvemos otra vez a seleccionar la opción 1 únicamente deberemos mostrar el configurador que se explicará a continuación.

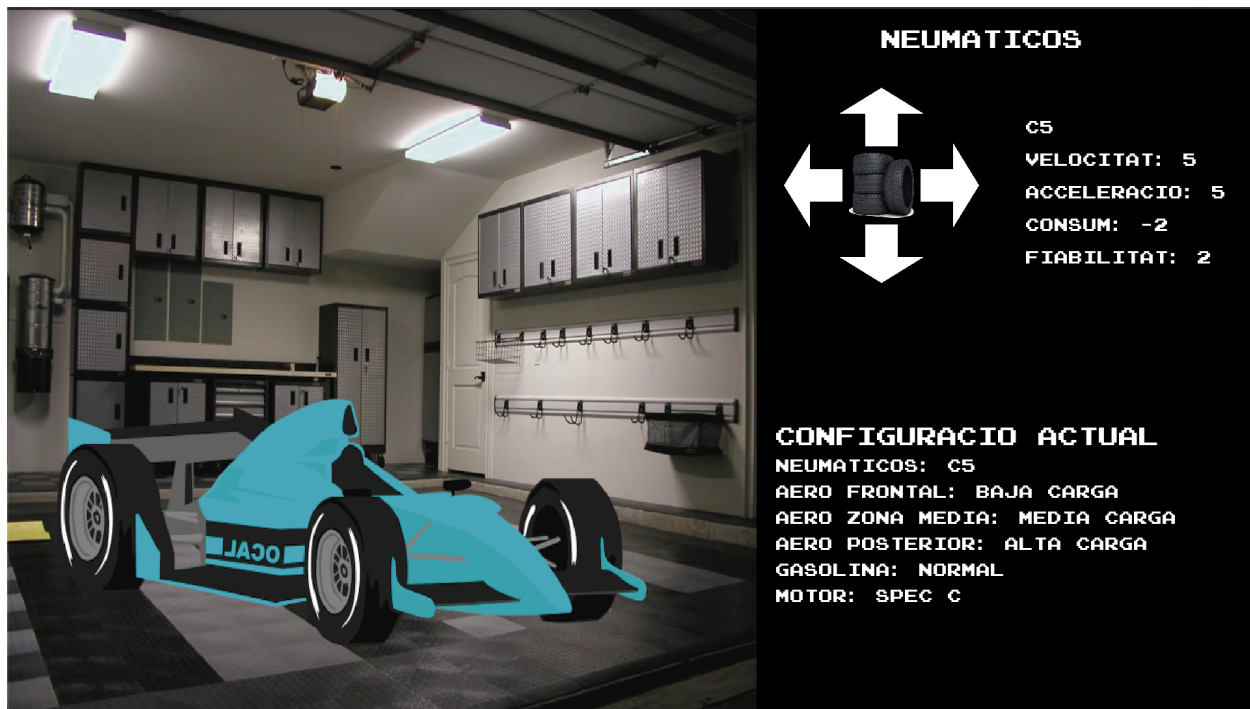


Figure 1: Vista del configurador

Como podemos observar en la imagen superior, la pantalla se divide en dos partes. En la parte izquierda tenemos nuestro garaje con el coche y en la derecha el panel de configuración.

Referente a la parte izquierda, en el eStudy encontraréis una imagen de ejemplo que podéis utilizar, pero está permitido utilizar otra imagen que os guste más, siempre y cuando se respete el funcionamiento de la práctica.

La única funcionalidad que tendrá la parte izquierda será mostrar la imagen de nuestro garaje.

El panel de configuración está formado por dos elementos.

2.1.1 PANEL DE CONFIGURACIÓN: PIEZA

En la parte superior se situará la información de la pieza que se está mostrando. Como podemos observar, junto a la imagen de la pieza se encuentran cuatro flechas, que representan las cuatro acciones que podemos realizar.

Las flechas de izquierda y derecha nos permiten mostrar la pieza anterior y siguiente, respectivamente, de la misma categoría, es decir, si estamos mirando motores, al pulsar la tecla de la flecha derecha se

mostrará el siguiente motor y al pulsar la de izquierda, el anterior. Su funcionamiento es circular, por lo que cuando nos encontremos en el último motor y pulsemos derecha, se deberá mostrar el primero, y cuando nos encontremos en el primer motor y pulsemos izquierda, se mostrará el último.

Las flechas de arriba y abajo nos permiten cambiar la categoría de pieza que estamos configurando, es decir, si estamos configurando el motor y pulsamos la flecha de abajo, cambiaremos a, por ejemplo, suspensión. Su funcionamiento también es circular.

La pieza escogida para el coche será la pieza seleccionada antes de cambiar de categoría. Si, por ejemplo, estamos mostrando el tercer motor y pasamos a mostrar las suspensiones, el motor escogido para nuestro coche será el tercero. Una vez guardada una pieza, se puede volver a cambiar tantas veces como el usuario quiera.

Para salir y volver al menú, se tendrá que pulsar la tecla ESC. Al hacerlo, antes de cerrar la ventana habrá que guardar la pieza que se está mostrando en pantalla en la configuración del coche.

Al lado de la imagen de la pieza, se mostrarán sus estadísticas.

2.1.2 PANEL DE CONFIGURACIÓN: CONFIGURACIÓN DEL COCHE

En la parte inferior se encuentra la configuración que estamos haciendo del coche. En ella, se mostrará el nombre de la pieza escogida para cada categoría disponible. Se tendrá que actualizar cada vez que realicemos un cambio en la configuración, es decir, que escojamos una pieza.

2.2 OPCIÓN 2: CARRERA

Al seleccionar la opción 2, cargaremos la pantalla de carrera.

Si es la primera vez que ejecutamos la opción 2, el gran premio escogido será el primero de la lista, si es la segunda vez, se escogerá el segundo y así sucesivamente hasta el último. Si hemos completado todos los grandes premios, en vez de cargar la pantalla deberemos mostrar, por línea de comandos, un mensaje informativo.

Bienvenidos a LS Racing!

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

Que opcion quieres ejecutar? 2

```
Preparando carrera #1: Australia ...
```

Output 10: Primera vez que seleccionamos la opción 2

```
Bienvenidos a LS Racing!
```

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

```
Que opcion quieres ejecutar? 2
```

```
Ya has finalizado la temporada.
```

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

```
Que opcion quieres ejecutar?
```

Output 11: Todas los grandes premios completados

También tenemos que controlar que hayamos configurado el coche, es decir, ejecutado la opción 1. En caso contrario, mostraremos un mensaje de error y volveremos al menú.

```
Bienvenidos a LS Racing!
```

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

```
Que opcion quieres ejecutar? 2
```

```
Aun no has configurado el coche
```

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

```
Que opcion quieres ejecutar?
```

Output 12: La opción 1 nunca se ha ejecutado



Figure 2: Vista de la carrera inicial

Como podéis observar, se mostrará el nombre del gran premio junto con sus estadísticas. También aparecerá un mensaje indicando que se pulse *R* para empezar.

Cada gran premio tiene unas estadísticas determinadas. Cuanto más cercanas sean las estadísticas de un coche a las del circuito, más probabilidades de ganar tendrá.

Una vez cargada la pantalla, no se hará nada hasta que pulsemos la tecla *R*. Una vez pulsada, aparecerá el semáforo en pantalla.

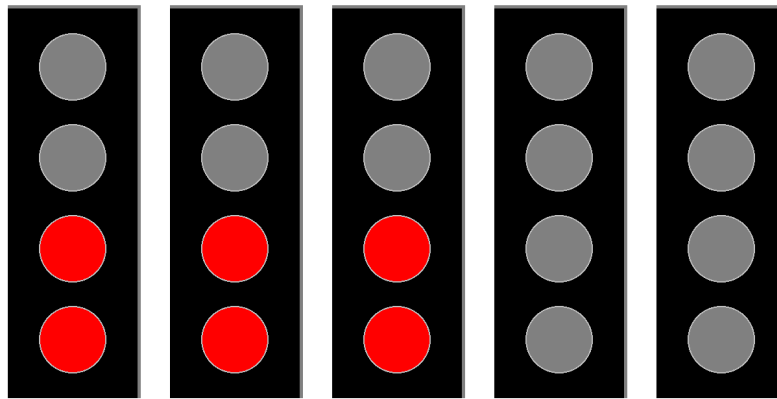


Figure 3: Vista del semáforo

El funcionamiento del semáforo es el siguiente. Primero estarán todas las luces apagadas. Luego, cuando pase 1 segundo, se encenderán las dos luces inferiores del semáforo de la izquierda, cuando pase otro segundo, se encenderán las del siguiente semáforo, y así sucesivamente hasta llegar al último. Finalmente, cuando estén todos encendidos, esperaremos 2 segundos y se apagaran todos los semáforos, esperaremos medio segundo y daremos inicio a la carrera.



Figure 4: Vista de la carrera

Como podemos observar en la imagen, la pantalla está dividida en tres partes. En la parte izquierda superior encontramos la representación de la carrera, en la que podemos encontrar tantas líneas como corredores, en la parte izquierda inferior se encuentra una leyenda de las dos acciones que podemos realizar y en la parte derecha encontramos la información del piloto (nombre, escudería y dorsal) y del coche, así como el tiempo total que ha pasado desde el inicio de la carrera y el número de paradas en boxes realizadas sobre el total que tiene que realizar.

Cuando empiece la carrera habrá que actualizar, en tiempo real y con precisión de 1 segundo, el tiempo transcurrido.

En cualquier momento el piloto puede realizar dos acciones. Estas tienen que ser instantáneas, es decir, en el momento que se pulsa la tecla hay que ejecutar la acción, no se puede esperar a que pase 1 segundo para hacerlo. Las dos acciones son:

- Radio: Tecla *R*. Llama por radio al equipo y pide que preparen el pit stop.
- Pit stop: Tecla *P*. Realiza el pit stop para cambiar los neumáticos y poner combustible. Cada vez que para, se aplica una penalización de tiempo, ya que no está corriendo. Es decir, durante la duración de la parada, nuestro coche no se moverá. Si no se ha pedido la parada previamente por radio, se aplica la penalización de tiempo pero no se cambian los neumáticos ni se pone

combustible, por lo que no cuenta como parada de pit stop. Si ya hemos realizado todos los pit stop, al pulsar *P* no pasará nada.

Hay que realizar tantas paradas en pit stop como requiera la configuración del coche y del circuito. El tiempo de cada parada dependerá del circuito, y está indicado en el fichero de grandes premios.

Cada GP tiene un tiempo base de carrera, que es el tiempo que tarda un piloto en finalizar la carrera en circunstancias ideales. Para saber cuanto tarda cada piloto en finalizar la carrera, deberemos calcular el total de la diferencia entre las estadísticas del GP y las de su configuración, y sumarlo al tiempo base. También hay que sumar el tiempo de los pit stops y restar el coeficiente de habilidad del piloto. Pongamos un ejemplo.

El Gran Premio de Australia se completa en 60 segundos, tiene 3 pit stops, que tardan 5 segundos cada uno, y tiene las siguientes estadísticas:

- Velocidad adecuada: 108
- Aceleración adecuada: 67
- Consumo adecuado: 50
- Fiabilidad adecuada: 120

Y nuestro piloto las siguientes:

- Velocidad: 100
- Aceleración: 70
- Consumo: 48
- Fiabilidad : 120
- Reflejos: 8
- Condición física: 9
- Temperamento: 2
- Gestión de neumáticos: 7

Entonces, deberemos sumar las diferencias (8 de velocidad, 3 de aceleración, 2 de consumo y 0 de fiabilidad) al tiempo base, es decir, $60 + 8 + 3 + 2 + 0$, obteniendo un total de 73 segundos.

Para saber el número de pit stops que hay que realizar, deberemos coger el número de pit stops del circuito y si el consumo es inferior al adecuado restaremos uno, y si es superior sumaremos uno. Siguiendo con el ejemplo anterior, el GP de Australia pide 3 pit stops y tiene un consumo adecuado de

50, como el consumo de nuestro piloto es 48, por lo tanto inferior, deberá realizar 2 pit stops. Por lo tanto, sumaremos el tiempo de los pit stops ($5 * 2 = 10$), al tiempo total, obteniendo un total de 83 segundos.

Finalmente, le restaremos el coeficiente de habilidad. Éste se calcula haciendo la media entera de las cuatro estadísticas y dividiendo entre 2. Siguiendo el ejemplo, 8 de reflejos + 9 de condición física + 2 de temperamento + 7 de gestión de neumáticos da un total de 26, entre 4, 6. Luego dividimos 6 entre 2, obteniendo 3, que es lo que restaremos del total, obteniendo así un tiempo total de 80 segundos.

La representación de la carrera se compone de tantas líneas como corredores, es decir, cada corredor tiene asignada una línea. Para identificar cada corredor, en el inicio de la línea se encontrará su dorsal.

El movimiento del coche por la línea tiene que simular la carrera, por lo que cada coche se moverá a la velocidad que le toque. Siguiendo con el ejemplo, nuestro corredor tarda 80 segundos en finalizar la carrera, de los cuales 10 son de pit stop (no avanza en la carrera). Por lo tanto, tardará 70 segundos en recorrer todo el circuito (línea). Si la línea tiene una longitud de, por ejemplo, 700 píxeles, cada segundo el coche avanzará 10 píxeles.

Como no podemos saber en que momento realizarán los pit stops el resto de pilotos, para su movimiento utilizaremos su tiempo total, es decir, no restaremos el tiempo que están parados realizando pit stops.

La carrera finalizará cuando se llegue al tiempo que tarda nuestro piloto, el resto se simularán. Por lo tanto, siguiendo el ejemplo anterior, cuando pasen 80 segundos, se deberá finalizar la carrera y mostrar nuestra posición. Para volver al menú, pulsaremos ENTER.

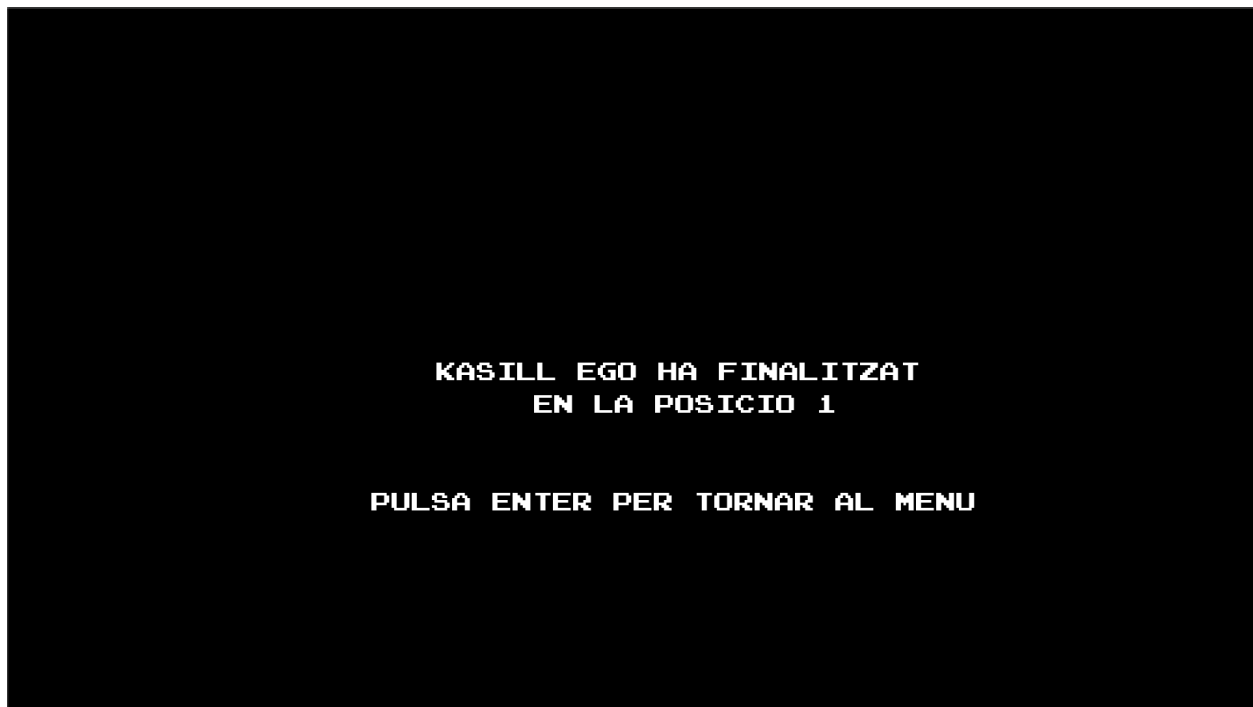


Figure 5: Carrera finalizada

Por último, en caso que nuestro piloto no haya efectuado todos los pit stops, se le sumara a su tiempo de carrera una penalización de 5 veces el tiempo de pit stop, en nuestro ejemplo, 25 segundos ($5 * 5$).

2.3 OPCIÓN 3: CLASIFICACIÓN

La opción 3 del programa será la encargada de mostrar la clasificación de todos los grandes premios completados hasta el momento.

Si aun no se ha completado ninguno, deberemos mostrar un mensaje informativo.

Bienvenidos a LS Racing!

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

Que opcion quieres ejecutar? 3

La temporada aun no ha empezado.

1. Configurar coche
2. Carrera
3. Ver clasificacion
4. Guardar temporada

Que opcion quieres ejecutar?

Output 13: Temporada no empezada

En cambio, si ya hay al menos un gran premio completado, cargaremos la pantalla de clasificación.



Figure 6: Clasificación de un GP

Al cargar, se deberá mostrar la clasificación del último gran premio completado. La clasificación deberá mostrar la posición en la carrera, el nombre del piloto, el tiempo de carrera y el total de puntos conseguidos durante la competición.

Además, podemos utilizar las teclas *A* y *D* para mostrar la clasificación de la carrera, respectivamente, anterior y posterior. Su funcionamiento no es circular, por lo que si nos encontramos en la última carrera y pulsamos la tecla *D*, o nos encontramos en la primera y pulsamos la tecla *A*, no deberá pasar nada.

Como hay que llevar un historial de las clasificaciones de todos los grandes premios, será necesario añadir una funcionalidad extra en la opción 2. Cada vez que se simule el gran premio, antes de volver al menú deberemos guardar la clasificación de esa carrera de forma que nos permita mostrar en la opción 3 la posición, nombre, tiempo y puntos totales en ese gran premio.

En la opción 3, la clasificación se tiene que mostrar en orden, por lo que se deberá utilizar un método de ordenación, a vuestra elección, para ordenar a los pilotos. El piloto con menor tiempo será el ganador y el piloto con mayor tiempo será el último.

Finalmente, y únicamente cuando se haya completado el último gran premio, si nos encontramos en éste y pulsamos la tecla *D*, se mostrará la clasificación final de la competición, ordenada según el número total de puntos. Ésta mostrará la posición, nombre del piloto, dorsal y el total de puntos.



CLASSIFICACIO FINAL DE LA TEMPORADA			
1.	PILOTO 5	(#15)	- 79 PUNTS
2.	KASILL EGO	(#3)	- 78 PUNTS
3.	PILOTO 1	(#29)	- 63 PUNTS
4.	PILOTO 4	(#84)	- 45 PUNTS
5.	PILOTO 7	(#53)	- 34 PUNTS
6.	PILOTO 2	(#35)	- 26 PUNTS
7.	PILOTO 6	(#90)	- 21 PUNTS
8.	PILOTO 3	(#31)	- 14 PUNTS

Figure 7: Clasificación final de la temporada

El sistema de puntos es el siguiente:

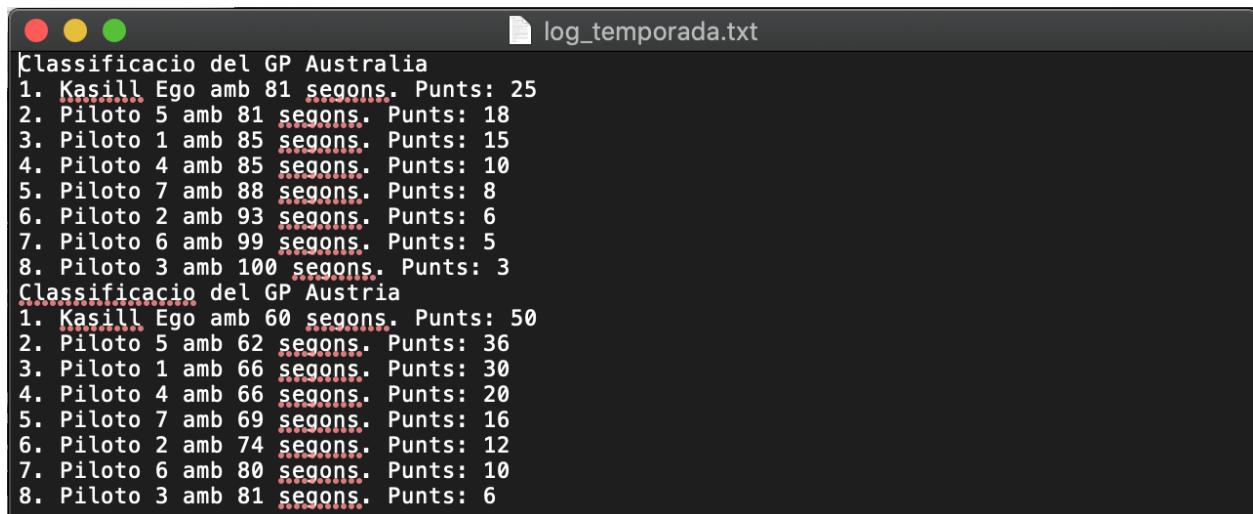
- Posición 1: 25 puntos
- Posición 2: 18 puntos

- Posición 3: 15 puntos
- Posición 4: 10 puntos
- Posición 5: 8 puntos
- Posición 6: 6 puntos
- Posición 7: 5 puntos
- Posición 8: 3 puntos

Para volver al menú se deberá pulsar la tecla ESC.

2.4 OPCIÓN 4: GUARDAR TEMPORADA

Esta opción será la encargada de generar un *log* de la temporada. Para esto, deberá guardar en un fichero de texto la clasificación de cada carrera y los puntos totales de cada piloto en ese momento, es decir, la información mostrada en la opción 3. El fichero tiene que ser entendible para un humano, por lo que se deberá indicar a que GP pertenece cada clasificación. Un ejemplo de fichero generado es:



```
log_temporada.txt
Classificacio del GP Australia
1. Kasill Ego amb 81 segons. Punts: 25
2. Piloto 5 amb 81 segons. Punts: 18
3. Piloto 1 amb 85 segons. Punts: 15
4. Piloto 4 amb 85 segons. Punts: 10
5. Piloto 7 amb 88 segons. Punts: 8
6. Piloto 2 amb 93 segons. Punts: 6
7. Piloto 6 amb 99 segons. Punts: 5
8. Piloto 3 amb 100 segons. Punts: 3
Classificacio del GP Austria
1. Kasill Ego amb 60 segons. Punts: 50
2. Piloto 5 amb 62 segons. Punts: 36
3. Piloto 1 amb 66 segons. Punts: 30
4. Piloto 4 amb 66 segons. Punts: 20
5. Piloto 7 amb 69 segons. Punts: 16
6. Piloto 2 amb 74 segons. Punts: 12
7. Piloto 6 amb 80 segons. Punts: 10
8. Piloto 3 amb 81 segons. Punts: 6
```

Figure 8: Ejemplo de fichero de *log*

2.5 FICHEROS

2.5.1 FICHERO DE PIEZAS

El fichero contiene datos sobre todas las piezas disponibles agrupadas según su categoría. Lo primero que encontraremos es el número de categorías, un entero que indica el número de categorías que se

encuentran en el fichero. Después se encontrarán las categorías. Os garantizamos que habrá tantas categorías como indique el número de categorías. Cada categoría está formada por la siguiente información:

- Nombre categoría: cadena de caracteres de hasta 25 caracteres que indica el nombre de la categoría en cuestión.
- Número de piezas: entero que indica el número de piezas que contiene esta categoría.

Os garantizamos que habrá tantas piezas como indique el número de piezas. Cada pieza sigue el formato siguiente:

- Nombre pieza: cadena de caracteres de hasta 25 caracteres que indica el nombre de la pieza en cuestión.
- Velocidad: *int* que indica como afecta el uso de esta pieza en la velocidad del coche. Puede ser un número negativo. El rango de valores será de -5 a 5 incluidos.
- Aceleración: *int* también entre -5 y 5 que indicará cómo contribuye la pieza a la aceleración del monoplaça.
- Consumo: *int* entre -5 y 5 que expresa si la pieza mejora o empeora el consumo de combustible del coche.
- Fiabilidad: *int* de -5 a 5 que indica si esta pieza hace más propenso que el coche se rompa.

2.5.2 FICHERO DE GPs

Este fichero dará la información necesaria sobre los distintos grandes premios que componen la temporada. Lo primero que se encontrará es un entero positivo que indicará la cantidad de grandes premios sobre los que se deberá leer la información. Garantizamos que este número coincidirá con la cantidad de GP que contiene el fichero. Por cada Gran Premio habrá los siguientes datos:

- Posición en el calendario: Valor entero que irá de 1 a la cantidad total de GP de la temporada (se garantiza que no será superior al total ni menor a 1) y que indicará la posición del GP dentro del calendario.
- Nombre: cadena de caracteres de hasta 25 caracteres que indicará el nombre del gran premio. Por ejemplo: GP de Austria.
- Velocidad adecuada: Entero que indicará la velocidad adecuada de ese circuito.
- Aceleración adecuada: Entero que indicará el valor total de la aceleración óptima para el circuito.
- Consumo adecuado: Entero que indicará el valor de consumo ideal para el circuito.
- Fiabilidad adecuada: Entero que indica el valor adecuado de fiabilidad en función de lo exigente que sea el circuito concreto.

- Tiempo base: *float* que indica los **minutos** totales que se tarda de base en realizar el gran premio.
- Tiempo *pit-stop*: *int* que indica los **segundos** que se tarda en realizar una parada a boxes, también conocido como *pit-stop*.
- Numero de pit stops: Entero que indica el numero de paradas óptimas.

2.5.3 FICHERO DE CORREDORES

Este fichero proporcionará toda la información necesaria en referente a los pilotos. Por cada piloto habrá:

- Nombre: Cadena de caracteres de hasta 25 caracteres que indicará el nombre del piloto.
- Dorsal: Entero que indicará el dorsal del piloto (y en consecuencia del coche) en cuestión.
- Escudería: Cadena de caracteres de hasta 25 caracteres que indicará el nombre de la escudería con la que compite el piloto.
- Velocidad: *int* que indica como afecta el uso de esta pieza en la velocidad del coche. Puede ser un número negativo. El rango de valores será de -5 a 5 incluidos.
- Aceleración: *int* también entre -5 y 5, ambos que indicará cómo contribuye la pieza a la aceleración del monoplace.
- Consumo: *int* entre -5 y 5 que expresa si la pieza mejora o empeora el consumo de combustible del coche.
- Fiabilidad: *int* de -5 a 5 que indica si esta pieza hace más propenso que el coche se rompa.
- Reflejos: *int* entre 0 y 10, ambos incluidos, que representa la capacidad de reacción del piloto.
- Condición física: *int* entre 0 y 10, ambos incluidos, que representa la condición física del piloto.
- Temperamento: *int* entre 0 y 10, ambos incluidos, que representa el temperamento del piloto.
- Gestión de neumáticos: *int* entre 0 y 10, ambos incluidos, que representa la técnica de conducción para aprovechar los neumáticos del piloto.

Si el fichero no está vacío, se os garantiza que siempre habrá 7 pilotos. Aun así, es obligatorio guardar la información de los pilotos en un **array dinámico**.

2.5.4 FICHERO BASE

Este fichero proporcionará las estadísticas base de los coches. Está formado por los siguientes campos:

- Velocidad: entero que indica la velocidad base de los coches. Puede ser cualquier valor entero.
- Aceleración: entero que indica la aceleración base de los coches. Puede ser cualquier valor entero.
- Consumo: entero que indica el consumo base de los coches. Puede ser cualquier valor entero.
- Fiabilidad: entero que indica la fiabilidad base de los coches. Puede ser cualquier valor entero.

2.6 DISEÑO

Este proyecto tiene que estar obligatoriamente estructurado en módulos, pero al inicio de su desarrollo aun no los habéis visto en clase.

Por tal de facilitaros más adelante el cambio de un único fichero (main.c) a módulos, os recomendamos que separéis el código en funcionalidades y cada una la separéis en funciones y procedimientos. Finalmente, todo este conjunto de funciones y procedimientos los agrupéis juntos en el código. Un ejemplo de esto es:

```
1  #include <stdio.h>
2
3  //GRAFICOS: Encargado de gestionar las funciones visuales de Allegro
4  void dibujaPanelConfiguracion(){}
5  void dibujaSemaforo(){}
6  void dibujaPodio(){}
7
8  //FICHEROS: Encargado de gestionar la lectura y escritura de los ficheros
9  void leePiezas(){}
10 void leeGrandesPremios(){}
11
12 //CLASIFICACION: Encargado de gestionar la clasificacion de los pilotos
13 void ordenaPilotos(){}
14 void calculaPuntos(){}
15
16 //ETC
17
18 int main(int argc, char *argv[]) {
19
20     return 0;
21 }
22
```

Figure 9: Ejemplo de estructuración del main

2.7 OPCIONALES

De forma opcional, podéis añadir cualquier funcionalidad que permite Allegro, como poner sonidos, siempre y cuando se respete el funcionamiento del programa.

Como bien dice su nombre, esto es opcional, por lo que no se penalizará su ausencia. En cambio, si se implementa sí que repercutirá de forma positiva en la nota del proyecto.

Recordad que cualquier funcionalidad opcional que implementéis deberá estar explicada en la memoria del proyecto.

3 CONSIDERACIONES

Para la implementación de esta práctica tened en cuenta las siguientes consideraciones:

1. Se debe seguir el formato mostrado en los ejemplos.
2. Los nombres de los ficheros se tienen que pasar como argumentos del programa, no se pueden escribir directamente en el código.
3. Cuando el valor a introducir sea un número, no se os garantiza que se introduzca un número, por lo que todas las entradas por teclado deberán aceptar cadena de caracteres.
4. Está permitido utilizar la función `atoi`.
5. El valor seleccionado para las opciones del menú tiene que ser un **entero** entre 1 y 4, ambos incluidos. Cualquier otro valor será considerado erróneo.
6. El programa tiene que estar estructurado en módulos.
7. Cada módulo tiene que estar estructurado en funciones y procedimientos.
8. Todos los nombres (categoría, pieza, GP, corredor, etc) serán **cadenas de caracteres** de como máximo **25** caracteres. Recordad que también se necesita espacio para guardar el carácter `\0`.
9. La información del fichero de piezas se tiene que guardar en **arrays dinámicos**.
10. La información del fichero de GPs se tiene que guardar en una **lista ordenada**.
11. La información del fichero de corredores se tiene que guardar en un *array dinámico*.
12. Las características que definen los componentes del coche (velocidad, consumo, fiabilidad y aceleración) se asegura que van a ser enteros con valores válidos, es decir `[-5,5]`.
13. Los enteros que indican la velocidad, aceleración, fiabilidad y consumo adecuados en el fichero de grandes premios se asegura que tendrán valores válidos.
14. Las características que definen las características del piloto (reflejos, condición física, temperamento y gestión de neumáticos) tienen que tener un valor entre 0 y 10, ambos incluidos.
15. El formato y contenido de los ficheros siempre será correcto, siempre y cuando no estén vacíos.
16. Hay que comprobar que los ficheros no estén vacíos y, si lo están, mostrar un error y finalizar la ejecución del programa.

4 REQUISITOS MÍNIMOS

Esta práctica debe cumplir con una serie de requisitos mínimos de calidad para poder evaluarla:

- El código debe estar correctamente comentado de manera que sea legible sin dificultades.
- Se debe seguir la guía de estilos de la asignatura.
- El programa debe desarrollarse íntegramente en el entorno CLion.
- No se puede usar ninguna herramienta ni instrucción del lenguaje C que no se haya tratado en clase
- No se aceptará ninguna práctica que no cumpla la normativa de prácticas.

5 NORMATIVA GENERAL

El proyecto del 2n semestre consta de diversas funcionalidades que se evaluarán como una única práctica. Todas las funcionalidades tienen que estar implementadas correctamente. En el supuesto de no entregar el proyecto, la nota de la práctica en el sistema de información debe constar como NP (No Presentado). En caso de que alguno de los elementos a evaluar (calidad de código, calidad de ejecución, memoria o entrevista) no esté aprobado, la nota de la práctica en el sistema de información debe constar como la nota más baja entre los diferentes apartados.

Del proyecto se evaluará por separado la calidad de ejecución y la calidad de código. La nota de la memoria será común para ambos y la nota de entrevista individual para cada miembro del grupo.

Para poder evaluar el proyecto, éste tiene que seguir la nomenclatura y formato especificados en el apartado 7.

Nota Proyecto 2n Semestre = Calidad Ejecución (40 %) +
Calidad Código (20 %) + Memoria (20 %) + Entrevista (20 %)

6 CONTENIDO DE LA MEMORIA

La memoria debe contener **OBLIGATORIAMENTE** los siguientes apartados:

- Portada de la practica rellena.
- Índice numerado.
- Breve resumen del enunciado del proyecto y cómo se ha decidido diseñar la práctica.
- Estructura de la práctica
 - Explicación de los módulos.
 - Explicación de cada función dentro del módulo correspondiente.
- Estructuras de datos
 - Explicación de las estructuras de datos utilizadas.
 - Justificación de la elección.
 - Diagrama de actividades del funcionamiento de la estructura en vuestro programa.
- Problemas observados y cómo se han solucionado.
- Tiempo de realización de la práctica con una gráfica incluida y una explicación de ésta (Fases y Memoria).
- Contenido extra que se haya podido añadir. Este apartado es obligatorio para aquellos que hayan puesto algo de contenido extra a la práctica que no esté en este documento.
- Conclusiones elaboradas.

Cualquier memoria que no cumpla el formato será devuelta y la práctica será considerada no apta para aprobarla hasta que el alumno no la entregue correctamente.

7 FECHAS DE ENTREGA

Esta práctica contiene diferentes *checkpoints* que son opcionales pero recomendables para el correcto progreso de la práctica. A diferencia de una pre-entrega, los *checkpoints* no tienen nota. Simplemente se revisará tanto el código como la ejecución y se darán recomendaciones para poder mejorar la implementación. Este proceso se hará de forma **presencial** en los horarios de dudas de los becarios. Todos aquellos que quieran optar a hacer esta revisión deberán entregar antes del cierre del pozo correspondiente al *checkpoint*. Las fechas serán las siguientes:

- *Checkpoint* 1: diseño de módulos (08/03/2020).
- *Checkpoint* 2: opción 1 (22/03/2020).
- *Checkpoint* 3: opción 2 (19/04/2020).
- Entrega sobre 10 (10/05/2020). Entrevistas 14/05/2020 y 15/05/2020.
- Entrega sobre 8 (07/06/2020). Entrevistas del 10/06/2020 al 12/06/2020.
- Entrega sobre 6 (05/07/2020). Entrevistas del 13/07/2020 al 15/07/2020.

El formato de entrega para los *checkpoints* deberá ser el siguiente:

Archivo comprimido con el nombre **PJ2_<login1>_<login2>.zip** y contendrá el proyecto de CLion .

Cualquier entrega que no cumpla con los objetivos de cada *checkpoint* no será aceptada.

El formato de entrega final deberá ser el siguiente:

Archivo comprimido con el nombre **PJ2_<login1>_<login2>.zip** y contendrá:

- Proyecto de CLion
- Memoria en PDF

Cualquier entrega fuera de plazo o por correo no será aceptado. Esto se aplica tanto a los *checkpoints* como a las entregas finales.