Algoritmos y Programación II
Curso Mendez

.UBAfiuba
FACULTAD DE INGENIERÍA

Batalla de Pokémones

TP1

Fecha de Presentación: 24/04/2020

Fecha de Vencimiento: 01/05/2020

1. Introducción

Los pokémon son unas criaturas que se encuentran en la naturaleza. Existen 18 tipos de clasificación pokémon (planta, fuego, roca, volador, acero, etc). Cada pokémon tiene una cantidad determinada de movimientos o ataques que puede aprender, los cuales se pueden clasificar por su tipo o elemento (por ejemplo, el ataque trueno es un ataque tipo eléctrico). Además, los pokémon cuentan con diferentes atributos que los ayudan en las batallas: Fuerza, Agilidad, Inteligencia, Velocidad, entre otros.

Los **entrenadores pokémon**, tienen como meta cumplir dos objetivos: capturar a todas las especies de pokémon disponibles y completar así la información de todos los pokémon en la Pokédex (Enciclopedia Pokémon). Por otro lado, deben **entrenarlos y enfrentarlos** a pokémones pertenecientes a otros entrenadores para demostrar sus habilidades, fortalezas, talentos y así convertirse en un **Maestro Pokémon**.

Para cumplir esta meta, los entrenadores viajan a lo largo y ancho de las regiones del mundo Pokémon, recolectando medallas de gimnasio. Éstas se obtienen tras ganar en una **batalla Pokémon**, en la que tanto el entrenador como el líder del gimnasio enfrentan a sus pokémon para probar sus habilidades especiales.

2. Objetivo

El presente trabajo tiene como finalidad que el alumno repase algunos de los conocimientos adquiridos en Algoritmos y Programación I, así como también que comience a familiarizarse con las herramientas a utilizar a lo largo de la materia, tales como el manejo de memoria dinámica y la utilización de punteros a funciones.

3. Enunciado

En ciudad Carmín se llevará a cabo un torneo pokémon. Los entrenadores que se anoten, deberán registrar 3 pokémones con los cuales competirán en el mismo.

Luego de realizar el sorteo, se generó un archivo de texto con los datos de cada entrenador y los pokémones registrados para competir.

Antes de llevar a cabo el torneo, se quiere simular el desarrollo del mismo y para ésto es necesario que alguien desarrolle algunas funciones que ayuden a los organizadores del torneo a predecir el ganador.

Las funcionalidades que se deben implementar se detallan a continuación:

```c
/*
 * Creará la estructura torneo_t, reservando la memoria necesaria para el mismo.
 * Devolverá un puntero a un torneo, en el cual los participantes ya estarán cargados.
 * En caso de no poder crearlo, o que la ruta del archivo no sea válida, devolverá NULL.
 */
torneo_t* torneo_crear(char *ruta_archivo);

/*
 * Hará competir a los entrenadores, dejando en la siguiente ronda al ganador de la batalla.
 * El entrenador de la posición 0, lucha contra el de la posición 1, el de la 2 contra el de
 * la 3, etc.
 * Cuando un entrenador no tiene con quien luchar (cantidad impar de entrenadores en esa ronda)
 * pasa directamente a la siguiente.
 * El vector de entrenadores quedará solo con aquellos que ganaron la batalla (su tamaño se
 * ajustará luego de cada ronda).
 * Devolverá 0 si se jugó exitosamente, o -1 en caso contrario.
 * Si en el torneo hay sólo un entrenador, no se puede jugar y la funcion devuelve -1.
 */
int torneo_jugar_ronda(torneo_t* torneo, int (*ganador_batalla)(entrenador_t* ,entrenador_t*));

/*
 * Mostrará por pantalla el nombre de cada entrenador, junto a sus pokémones.
 * Dependiendo de la función enviada por parámetros (la cual refiere a una batalla), mostrará
 * distintas características de dichos pokémones.
 */
 void torneo_listar(torneo_t* torneo, void (*formatear_entrenador)(entrenador_t*));

/*
 * Destruirá la estructura del torneo, liberando la memoria reservada para él y los entrenadores.
 */
void torneo_destruir(torneo_t* torneo);
```

A su vez, se cuenta con los siguientes registros a utilizar en el presente trabajo:

```
typedef struct pokemon {
        char nombre[MAX_NOMBRE];
        int fuerza;
        int agilidad;
        int inteligencia;
} pokemon_t;

typedef struct entrenador {
        char nombre[MAX_NOMBRE];
        pokemon_t* pokemones;
} entrenador_t;

typedef struct torneo {
        entrenador_t* entrenadores;
        int cantidad_entrenadores;
        int ronda;
} torneo_t;
```

Los entrenadores y sus pokémones registrados se encontrarán en el archivo cuya ruta llega como parámetro.

Cada línea del archivo tendrá la siguiente información:

```
nombre_entrenador;nombre_p1;fuerza_p1;agilidad_p1;inteligencia_p1;nombre_p2;fuerza_p2;agilidad_p2;
    inteligencia_p2;nombre_p3;fuerza_p3;agilidad_p3;inteligencia_p3
```

Un ejemplo se muestra a continuación

```
Ash;Pikachu;40;90;70;Charizard;76;60;90;Snorlax;200;10;10
Giovanni;Nidoqueen;100;40;90;Nidoking;160;62;76;Rhydon;180;77;42
...
```

4. Resultado esperado

Se espera que el trabajo creado sea compilado sin errores con la siguiente línea:

```
gcc *.c -Wall -Werror -Wconversion -std=c99 -o torneo_pokemon
```

Luego, que sea ejecutado y permita simular el desarrollo de un torneo, utilizando las funciones de batalla que se especifiquen.

Para verificar el correcto funcionamiento, se recomienda utilizar un programa principal con el siguiente flujo:

```
/*
* Funciones que determinan el ganador de una batalla:
* - Como minimo es necesario crear 5.
* - Ejemplos:
* - - Pelea el pokemon 1 del jugador 1 con el pokemon 2 del jugador 2, gana el de menor fuerza, si
    son iguales, el de mayor agilidad.
* - - Pelean los pokemones 2 y 3 de cada entrenador, ganan los que sumen más inteligencia.
* - - Etc. sean creativos!
*/

int main(){
        /* Crear el torneo */
        /* Jugar ronda 1 con cierta función */
        /* Listar los entrenadores que siguen en el torneo */
        /* Jugar ronda 2 con cierta función */
        /* Listar los entrenadores que siguen en el torneo */
        ...
        /* Jugar ronda N con cierta función */
        /* Listar el ganador*/
        return 0;
}
```

Y, ser corrido con **valgrind**:

```
valgrind --leak-check=full --track-origins=yes --show-reachable=yes ./torneo_pokemon
```

Por ejemplo, si partimos del siguiente archivo participantes.txt, donde solo hubo 2 inscriptos...

```
Ash;Pikachu;40;90;70;Charizard;76;60;90;Snorlax;200;10;10
Giovanni;Nidoqueen;100;40;90;Nidoking;160;62;76;Rhydon;180;77;42
```

Y elegimos la función en la que luchan los pokemones 2 y 3 de cada entrenador y ganan los que sumen más inteligencia, debería obtenerse el siguiente resultado.

```
==5602== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==5602== Command: ./torneo_pokemon participantes.txt
==5602==

Torneo Pokèmon: Ronda 1 - Batalla de inteligencia - 2 entrenadores:
Ash: Charizard(90), Snorlax(10)
Giovanni: Nidoking(76), Rhydon(42)

Fin del torneo Pokèmon: Cantidad de rondas 1
Ganador del torneo: Giovanni (Nidoqueen, Nidoking, Rhydon)

==5602==
==5602== HEAP SUMMARY:
==5602==     in use at exit: 0 bytes in 0 blocks
==5602==   total heap usage: 9 allocs, 9 frees, 6,592 bytes allocated
==5602==
==5602== All heap blocks were freed -- no leaks are possible
==5602==
==5602== For counts of detected and suppressed errors, rerun with: -v
==5602== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

5. Entrega

La entrega deberá contar con todos los archivos necesarios para compilar y ejecutar correctamente el TP. Se recomienda que la misma conste de un archivo .c en donde transcurra el flujo principal del programa, y otro que contenga las implementaciones de la biblioteca brindada por la cátedra.

Dichos archivos deberán formar parte de un único archivo **.zip** el cual será entregado a través de la plataforma de corrección automática **Kwyjibo**.

El archivo comprimido deberá contar además con:

- Un **README.txt** que contenga:
  - Una breve introducción sobre el funcionamiento del trabajo presentado.
  - Una explicación de como compilarlo (línea de compilación) y como ejecutarlo (línea de ejecución).
  - Un breve desarrollo sobre los siguientes conceptos, y ejemplos de los lugares en donde fueron utilizados:
    1. Punteros.
    2. Aritmética de punteros.
    3. Punteros a funciones.
    4. Malloc y Realloc.
- El enunciado.

6. Referencias

- https://pokemon.fandom.com/es/wiki/Estadisticas
- http://www.juegosdb.com/guia-del-entrenador-pokemon-caracteristicas-de-un-pokemon-nintendo-ds-nintendo-3ds/
- https://es.wikipedia.org/wiki/Pokémon