Primer Proyecto

Programación Paralela y Concurrente

Profesor: Jose Andres Mena Arias

Intregrantes:

Alejandro Ramirez B96367

Oscar Navarro B95549

Manual de usuario:

Para compilar el código fuente se escribe lo siguiente en la consola,dentro del path donde se encuentra las carpetas de modelo, controlador y vista:

```
gcc `pkg-config --cflags gtk+-3.0` Controller/mapper.c Controller/GrandMaster.c Controller/pokemon.c View/pokemon_ui.c -lm -pthread -o poke_battle_i `pkg-config --libs gtk+-3.0`
```

Para ejecutar el programa se escribe en consola lo siguiente:

```
./poke_battle_i
```

Sin embargo para efectos de conveniencia y con archivo Makefile adjunto en la entrega basta con poner en la consola para compilar:

```
make
```

Con respecto a la entradas esperadas se mostraran 6 cajas de texto para que el usuario seleccione los 3 ids de pokemones de cada jugador, el valor de dichas ids tienen que ser entre 0 y 49 dado que esos son los pokemones disponibles, y se cada jugador no puede repetir ids entre sus pokemones dado que no pueden tener pokemones de la misma especie en su equipo.
Una vez elegidos los pokemones el usuario deberá presionar el botón start para que comience la simulación, en pantalla se mostrarán los 3 pokemones de cada jugador seguido de el pokémon activo su vida y su energía acumulada, y en el medio la información de los ataques, también cabe mencionar que a cada lado del botón de start se encuentra el panel de avisos para cada jugador.

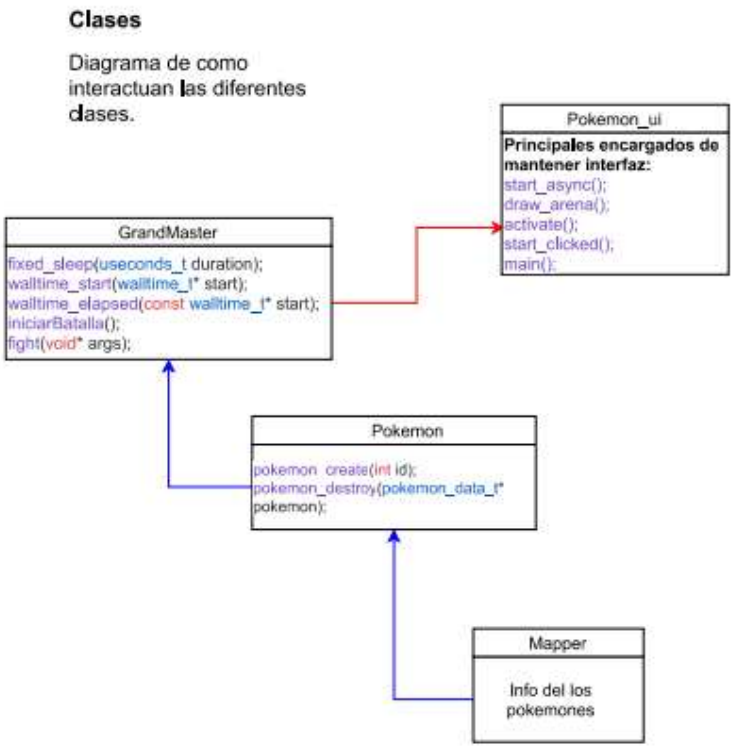
Detalles de la solución:
La estructura se divide en 3 folders, uno de modelo, uno del controlador y otra de la vista, dentro del de modelo se encuentra los archivos:

- **mapper:**
  el cual tiene el objetivo de utilizarlo para obtener la información de cada especie de pokémon a la hora de crearlo.

- **Pokemon:**
  el cual es el encargado generar y destruir cada pokemon que va a ser utilizado en la batalla, tiene tres tipos definidos los cuales son: **pokemon_data_t** el cual tiene toda la información básica del pokémon y dos punteros a otros dos tipos definidos lo cuales son **charged_move_t** el cual tiene la información del ataque cargado de dicho pokémon, y **fast_move_t** el cual tiene la información del ataque rapido.

- **GrandMaster:**
  el cual es el encargado de organizar la pelea y avisar a los pokémon para que peleen cuando deben hacerlo, dentro de esta clase se encuentran dos tipos definidos: **thread_data_t** que es utilizado para contener la información propia de cada hilo y el tipo **shared_data_t** el cual contiene toda la información compartida por los hilos.

  En la carpeta controlador se encuentra la implementación de dichos modelos respectivamente, además en la carpeta de vista se encuentra un archivo pokemon_ui.c el cual es el encargado de montar el interfaz.



Métodos de sincronización utilizados:

Variables de condición:
Se utilizaron 7 variables de condición en total, 6 para cada hilo encargadas de mantener el paralelismo y 1 para el hilo del GrandMaster.

Mutex:

Se utilizaron 7 mutex en total, 2 mutex fueron utilizados para que solo hubiese un hilo por equipo peleando a la vez. Otros dos mutex fueron utilizados y distribuidos de la siguiente manera: uno para que cada jugador bloquee su vida para revisarla y el otro para que cada jugador bloquee la vida del oponente para realizar un ataque, dichos son recíprocos para cada jugador. Los dos siguientes mutex fueron utilizados para chequear si hay cambio de pokemon, dichos son recíprocos para cada jugador. El mutex restante fue utilizado para que el GrandMaster empiece el cronómetro de la duración de la batalla

Barrier:
Se utilizó un barrier para asegurar que los dos primeros pokemones de cada equipo en pelear comenzasen a pelear al mismo tiempo.

Problemas:

Starvation
Hubo un problema de starvation en el caso de que los dos primeros hilos que iban a pelear llegasen de últimos, dado que el último hilo en llegar es el encargado de dar la señal al GrandMaster para que le de la señal a los dos primeros hilos para que comiencen a pelear, pero si el que realizaba esto era alguno de los primeros en pelear no les daba tiempo de ponerse en posición a la hora de que llegara la señal del GrandMaster de empezar a pelear entonces se quedaban esperando la señal para siempre.
La solución fue un if que si dicho hilo en llegar fuese alguno de los primeros en pelear los pasaba a un barrier para que esperasen a su oponente.