Segundo Proyecto

Programación Paralela y Concurrente

Profesor: José Andrés Mena Arias

Integrantes:

Alejandro Ramírez B96367

Oscar Navarro B95549

Manual de usuario:

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

```
mpic++ Controller/World.cpp Controller/mario.cpp Controller/Main.cpp -o mario_royale
```

Para ejecutar el programa se escribe en consola lo siguiente:

```
Mpiexec -n 6 ./mario_royale #MarioAEspectar #Strategia
```

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

```
Make
```

Con respecto a las entradas esperadas funcionara de la siguiente manera, a la hora de ejecutar el programa el usuario va a indicar primero su Mario a jugar, el que se va a mostrar en consola, y después el usuario va a indicar la estrategia de dicho Mario. Si dicho Mario muere el programa va a pedirle al usuario ingresar otro Mario al cual espectar, la entrada esperada sería un Mario mayor a cero y menor a la cantidad de Marios, además el programa verifica que dicho Mario siga vivo.

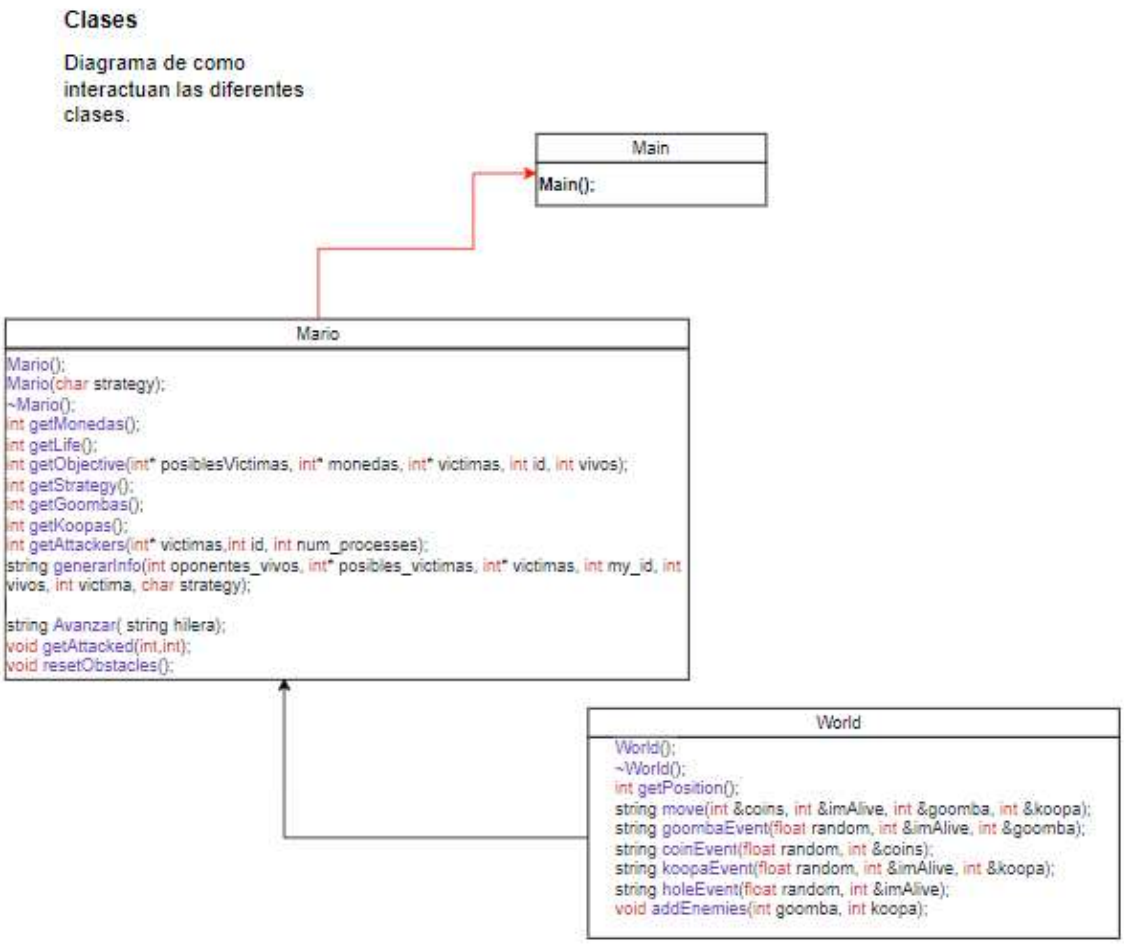
Detalles de la Solución

La estructura se divide en dos folders uno de modelo y otro de controlador, dentro de la carpeta de modelo se encuentran los siguientes archivos:

Mario: La cual es la encargada de generar y destruir cada Mario que va a participar en el battle royale, cada Mario posee los siguientes atributos: int coins, int isAlive, int koopas, int goombas, char strategy y World* world. Cabe aclarar que koopas y goombas es donde se van a ir guardando los enemigos que ese Mario mate para ser agregados al Mario enemigo. El atributo world es el mundo de ese Mario.

World: La cual es la encargada de contener el mundo de cada Mario con sus elementos el cual va a ser representado con dos colas, una estática que posee el mundo con el que empieza el Mario y la volátil que posee el mundo que va a ir cambiando a medida que Mario sea atacado. World posee los siguientes atributos: queue<string>* colaEstatica, queue<string>* colaVolatil y int position.

En la carpeta controlador se encuentra la implementación de dichos modelos respectivamente, además del Main con el cual se empieza y controla la batalla.



Funciones MPI utilizadas

MPI_Init(&argc, &argv) : Inicializa el ambiente de MPI.

MPI_Comm_size(MPI_COMM_WORLD, &num_processes) : Obtiene el rango de ese proceso en específico.

MPI_Comm_rank(MPI_COMM_WORLD, &my_id) : Obtiene el número total de procesos.

MPI_Allgather(&message, 1, MPI_INT, recv_buffer, 1, MPI_INT, MPI_COMM_WORLD) : Recolecta todos los datos de los diferentes Marios y los coloca en un arreglo, dicho arreglo lo tienen todos los procesos.

MPI_Allreduce(&im_alive, &vivos, 1, MPI_INT, MPI_SUM, MPI_COMM_WORLD): Recolecta todos los datos de los diferentes Marios y realiza una operación sobre de dichos datos, en este caso una suma de todos los datos y dicho resultado lo tienen todos los procesos.

MPI_Bcast(&espectador, 1, MPI_INT, 0, MPI_COMM_WORLD): hace que el proceso root haga un broadcast de algún dato a todos los procesos dentro del comunicador.

MPI_Send(&message_sent, 1 /*count*/, MPI_INT, my_id + 1 /*dest*/, 123 /*message id*/, MPI_COMM_WORLD): Envía un mensaje por medio de comunicación de punto a punto.

MPI_Recv(&message_received, 1 /* count*/, MPI_INT, my_id - 1 /*source*/, 123 /*message id*/, MPI_COMM_WORLD, &status) : Recibe un mensaje por medio de comunicación de punto a punto.

MPI_Finalize() : Cierra las comunicaciones de MPI.

Especificación de cantidad de llamados y su propósito:

Con respecto a los allgather se realizan varias veces durante la batalla tres tipos: uno para recolectar todas las monedas de los diferentes Marios, otro para recolectar sus vidas y el ultimo para obtener a quien atacan. Para el allreduce se utiliza varias veces con el objetivo de sumar todas las vidas de los varios para verificar que haya más de uno vivo. El Broadcast se utiliza con el objetivo de que el proceso root le comunique durante la batalla a los demás procesos el mario que debe motrar su información en la consola.

Las comunicaciones de punto a punto se utilizan varias veces durante la batalla para que cada Mario le envíe a su Mario enemigo los enemigos para que este lo coloque en su mundo, además se utiliza para que el Mario ganador le comunique al proceso root que el es el ganador de la batalla.

Extra

¡¡¡Nuevo feature!!! Se agrega a la simulación nueva habilidad especial y nuevo elemento:

Super Star: Elemento que al haber sido encontrado por un Mario este le proporciona la habilidad de no morir por koopas o por goombas durante 15 movimiento (Star Power). La probabilidad de que aparezca es de una 1%, sin embargo, esta habilidad no protege a Mario de caer en un Hole.