

Proyecto Final Programación 2

Grupo 7

Facultad de Ingeniería y Tecnologías

Integrantes:

José Varela

Nicolás Lorenzo

Manuel Baldomir

Juan Campos

Docentes:

Matias Olave

María Parapar

Felipe López

ÍNDICE

Funcionalidades añadidas.....	3
Funcionamiento	5
Test Cases	9
Roles	10

Funcionalidades añadidas

Se nos pide añadir nuevas funcionalidades al proyecto, por lo tanto se optó por agregar algunos modos de juegos diferentes al juego clásico y además agregarle algunas funcionalidades extras que se muestran a continuación.

- Bomb
- TimeTrial
- Challenge
- Quickchat
- Ranking

(La mayoría de estas funcionalidades serán implementadas para la tercer entrega ya que priorizamos el funcionamiento base del juego para esta segunda entrega, además el funcionamiento de algunas de ellas va a ir vinculado con el Bot)

Bomb: El objetivo de esta clase es afectar directamente la forma básica de ataque que tienen los usuarios mientras juegan a la guerra naval. Al aplicar este modo, luego de que alguno de los jugadores erre 3 tiros seguidos en sus turnos correspondientes, su próximo disparo no será de solo 1 slot sino que será un cuadrado de dimensión 3x3.

TimeTrial: Esta clase afecta directamente en la modalidad de juego y en quien gana la partida. No solo gana quien hunde todos los barcos enemigos, sino también el que haya golpeado a más barcos enemigos en el tiempo límite que tendrá la partida contra reloj. Al

finalizar el tiempo se verificará quien derribó y golpeó a más barcos enemigos y así se determinará quién ganó la partida si no fueron derribados todos los barcos. También afectará el tiempo que tiene cada jugador en realizar su ataque

Challenge: Tiene como objetivo realizar un reto con otro jugador y no solo jugar 1 partida entre sí, sino un mejor de 3.

QuickChat: Esta clase brinda la funcionalidad de poder intercambiar líneas rápidas de chat. Tendrá un pequeño repertorio de textos predefinidos que podrán ser utilizados por el jugador durante la partida para molestar al contrincante o simplemente ser amistoso.

Ranking: Con esta clase, siempre se va a mantener una única instancia de una lista de estadísticas de usuarios, ordenada según la cantidad de victorias de cada usuario y en caso de que coincidan, ordenadas también por su porcentaje de victorias. El administrador debe usar distintos métodos para mantener actualizada esta lista, mientras que el jugador puede ver el ranking global de todos los usuarios, el top 10 o su propia posición en el ranking global.

Funcionamiento

Para abordar este proyecto en el cual se debería crear o implementar un juego clásico llamado “Batalla Naval” para la primera entrega se decidió realizar un diagrama UML para así tener un primer acercamiento a lo que sería la solución de este. Para esta segunda entrega nos vimos forzados a realizar cambios en dicho diagrama, ya que nuestra visión para el correcto funcionamiento del programa tomó una mejor y más específica forma al comenzar a escribir su código.

La solución tiene varias clases para facilitar su funcionamiento desde el momento en que ingresa un usuario hasta que finaliza el juego.

Al comenzar, se muestra un menú, en el cual el usuario puede seleccionar alguna de las opciones disponibles, entre estas están las opciones de registrarse, jugar, salir, etc. Cuando un usuario se registra, la clase administrador verifica si ya existe ese usuario en la lista de usuarios registrados y si no lo está genera un nuevo objeto User. Al mismo tiempo, se crean sus estadísticas de partidas que comienzan en 0 y se le coloca en el ranking global del juego. En caso que el usuario ya esté registrado, le permite iniciar sesión ingresando el ID que se generó cuando creó el usuario y en caso de que no lo esté, se le pide que lo haga.

Una vez dentro, tendrá tres opciones, jugar, estadísticas y salir. La opción estadísticas le permite al jugador visualizar las estadísticas generales de los usuarios que hayan jugado, los juegos ganados por cada uno de ellos y el porcentaje de victorias de cada uno. En cambio, si selecciona jugar se le pide que seleccione el modo de juego que se desea jugar y el usuario es asignado en una lista de espera para el modo de juego seleccionado. Cuando dos o más usuarios estén en la misma lista, es decir que estén esperando para ingresar al mismo modo de juego, se toma a ambos usuarios y la clase Administrator genera una nueva instancia de

Game, y comienzan a jugar en dicha sesión. A su vez se los elimina de la lista de espera en la que estaban y el administrador monitorea todas las partidas que están en juego sin importar el modo.

Como observación de lo detallado anteriormente, cabe aclarar que en esta segunda entrega no se pudo implementar los distintos modos de juego que teníamos previstos. Esto se debe a que intentamos resolver la rigidez de nuestro código observado por los profesores en la primera entrega, en donde si en un futuro agregamos un modo de juego nuevo tendríamos que modificar las clases de “Menu” y “Administrator”, lo cual no es óptimo.

Intentamos implementar un Singleton como sugirió Fernando, pero no fue posible debido a su complejidad que luego fue confirmada por parte del mismo y que en una próxima clase nos explicaría cómo se podría aplicar en nuestro caso. Si bien no fue posible aplicar este cambio para la segunda entrega, buscaremos la forma de resolverlo para la próxima entrega.

Una vez iniciada la instancia de Game, se instancian dos nuevos tableros de la clase Board y quedan vinculados a cada jugador, esto permitirá hacer los chequeos de donde se encuentra posicionados los barcos enemigos y propios, así como almacenar en un listado todos los tiros realizados y la lógica de impresión del tablero que corresponda. Luego se le asigna a cada usuario 5 naves por defecto (Lancha, Crucero, Submarino, Acorazado, Portaaviones), las cuales deberá posicionar en su tablero de dimensión 10x10 indicando en primera instancia la coordenada inicial y luego la dirección hacia donde lo quiere posicionar (Arriba, Abajo, Izquierda y Derecha). Habrán chequeos internos verificando si la posición seleccionada por el jugador es una posición válida para la nave seleccionada, teniendo en cuenta que el barco no sea posicionado fuera del tablero ni tampoco que se solape con otro barco ya posicionado. es decir, que no se permita colocar el barco en una posición ya ocupada por otro.

Luego de posicionar los barcos comienza la batalla naval. Al jugar existen 2 acciones básicas, una de ellas es atacar y la otra es mostrar los tableros. Los tableros accesibles por los usuarios corresponden a su propio tablero y al tablero enemigo. En el tablero propio se visualizarán los ataques del oponente y los barcos del usuario que solicita el tablero, mientras que en el tablero del enemigo, serán visibles los ataques realizados por el usuario con distintivos de si acertó o erró a una nave enemiga mediante el símbolo correspondiente.

Mientras estén jugando atacarán por turnos, en la próxima entrega se añadirá también que exista un tiempo límite para cada turno, es decir, un jugador no podrá demorar más de un tiempo prefijado ya que automáticamente pierde la instancia de ataque y se le da el turno a su oponente. Además tendrán la posibilidad de enviarse mensajes a través de la nueva función quick chat o solicitar una lista con todos los comandos disponibles para el juego. Estas últimas dos funcionalidades no fueron incluidas en esta entrega, ya que aún no contábamos con el bot para implementarlo y que éste espere en todo momento los comandos enviados desde telegram para realizar las distintas interacciones durante el juego.

Todas las interacciones entre los jugadores y el juego se darán a través del chat bot quien ejecutará la acción solicitada por el jugador. Al día de hoy se cumple con el hilo conductor principal que permite crear y registrar un usuario, comenzar una nueva partida e ir cambiando de turnos entre los dos jugadores para realizar un ataque, luego los tableros de los mismos se verán afectados y constantemente se puede acceder a ellos para visualizarlos, por otro lado también se está chequeando constantemente cuál va a ser el jugador que gane la partida y al finalizar se les asignan las estadísticas ya calculadas a cada jugador correspondiente.

En el momento que se hayan derribado todas las naves de alguno de los jugadores, la partida finalizará y mostrará por pantalla quién fue el ganador, también se le sumará una victoria al jugador correspondiente y se guardarán sus estadísticas. Luego se los enviará al menú principal donde podrán elegir para jugar una nueva partida y buscar oponente o salir del juego. Al momento de retirar a ambos jugadores al menú principal se elimina la instancia de juego generada.

Ya explicado el juego “clásico”, como mencionamos anteriormente existen algunas modificaciones o modos de juego añadidos a este para que el usuario pueda seleccionar entre ellos cual prefiera.

Entre los nuevos modos de juego encontramos el modo “bomba” el cual es bastante particular porque como lo dice su nombre, afecta en el ataque del usuario, es decir, el modo de juego es totalmente idéntico al “clásico”, pero al momento de atacar, en caso que el usuario no haya acertado determinada cantidad de disparos (esto se realizará con un contador que cada vez que la respuesta del chequeo de si golpeó o no a un barco sea negativa irá aumentando, y luego de ejecutar el disparo especial se fijará nuevamente en 0), se le asigna un disparo “bomba” el cual afectará una dimensión de 3x3 celdas en el tablero enemigo, brindando de esta forma una posibilidad mucho mayor de acertar en el disparo a un barco enemigo.

Por otro lado también nos encontramos con otro modo de juego llamado “TimeTrial”, el cual también se basa en el modo de juego “clásico” pero en esta modalidad el tiempo es un factor muy importante debido a que se establece una partida con un tiempo que ya esté predefinido y será limitado, al momento que comienza la partida el tiempo comienza a correr y si al finalizar el tiempo ninguno de los dos jugadores ganó, se establece como ganador el jugador que haya acertado más disparos en las naves del oponente. Esta es una modalidad de

juego bastante interesante porque el juego se puede tornar un poco largo o extenso en caso de que los jugadores no acierten los disparos en las naves de los oponentes, en cambio con esta nueva funcionalidad se pueden hacer partidas más cortas y rápidas de jugar.

Finalmente se optó por agregar otra nueva funcionalidad a este juego la cual es de “challenge”, esta es bastante interesante o entretenida para el usuario ya que permite realizar varios juegos consecutivos con el mismo usuario, y gana el mejor de 3 partidas, esta modalidad también se basa en el juego “clásico” por lo cual todos los ataques y funciones del juego normal son iguales. La única diferencia es que habrá un contador de partidas ganadas para cada jugador y cuando uno de los dos alcance el valor 2, se indicará por pantalla quién fue el ganador del Challenge.

Test Cases

En esta sección queremos aclarar los problemas encontrados a la hora de realizar Tests para la segunda entrega. El problema que tuvimos es que nuestro código si bien funciona correctamente corriendo con el comando “dotnet run”, la gran cantidad de métodos no realiza returns. Esto generó un gran problema a la hora de realizar los Tests ya que nos era imposible realizar Asserts y testear las historias de usuario solicitadas para la entrega.

Luego de hablar con María, nos indicó que hagamos tests básicos con la finalidad de demostrar que sabemos realizar tests en C# y que el funcionamiento será verificado mirando todo el código y verificando la ejecución del mismo.

Roles

En cuanto al diagrama UML que se realizó, se pudieron identificar diferentes roles en las clases. Identificar los roles que hay en la solución provista es muy importante ya que ayuda a identificar el propósito de cada clase o darle una importancia no menor a cada una de estas.

Por otro lado también es útil identificar cada clase por su rol ya que se puede observar de una forma más clara que responsabilidad tiene cada clase, que en algunas ocasiones es bastante complicado dar una responsabilidad a cada clase sin antes haber definido su propósito.

Cabe destacar que las clases no pueden tener más de un rol, y todas deberían tener uno, ya que de esta forma nos ayuda a implementar nuestra clase a través de su finalidad.

A medida que se desarrolle el proyecto existe la posibilidad de que se añadan nuevas clases junto a sus roles, pero en el estado temprano actual del mismo identificamos los siguientes roles:

-Information Holder asignamos a las clases: User, Ship & Statistics ya que su funcionalidad es almacenar y proveer información.

-Structurer se le asigna a la clase Ranking que vincula distintos objetos y almacena información sobre dicho vínculo.

-Service Provider la clase Board, ya que realiza trabajo y ofrece información y servicios a otra clase.

-Coordinator a las clases Administrator, QuickChat y Commands cuyas funciones son de reaccionar y delegar tareas a otras clases.

-Controller Game ya que se encarga de tomar decisiones y dirigir otras clases.

-Interfacer las clases Menu, Bomb, TimeTrial y Challenge ya que son clases que cambian información entre clases alterando el funcionamiento del juego principal.