Funciones extras del juego:

1- Seleccionar la id del usuario y poder acceder al registro y resumen de las partidas jugadas por este.
2- Que los barcos posean habilidades especiales, por ejemplo, el barco acorazado debe ser golpeado dos veces, el destructor pueda cambiar su ubicación y el portaviones pueda sacar una foto satelital de un cuadrado de 2x2 del tablero de las naves del enemigo.
3- Poder elegir entre tres modos de juegos, la diferencia entre ellos es el tamaño del tablero, y por ende el tiempo que demorará la partida.
4- Al comienzo de cada partida, cada jugador tiene 3 comodines (WildCard) que pueden ser utilizados solamente una vez:
    a. Tener dos turnos consecutivos
    b. Tener una bomba para disparar a 4 casilleros en 1 turno (los casilleros deben formar un cuadrado, o sea estar unidos)
    c. El tablero se divide en 4 sectores (izquierdo arriba, izquierdo abajo, derecho arriba, derecho abajo). Con esta habilidad el jugador puede obtener información sobre en cuál de los sectores existen más puntos de impactos
5- Modo de juego extra: Poder colocar un límite de tiempo para ingresar una coordenada a disparar, si se pasa el tiempo pierde el turno. Similar al sistema de reloj en los juegos de ajedrez. (TimeLimitedGame)

6- * Modo de juego donde el jugador tiene un barco que repara (otros barcos, cualquiera), pero no puede repararse a sí mismo y si lo destruyen ya no se puede reparar los barcos. Reparar un barco equivale a gastar un turno.
7- * Modo de juego predictivo. Cada tiro del jugador atacante analizará un rango de un cuadrante a su alrededor y en vez de decir “agua” dirá “casi le diste” o algo similar.

* Las funcionalidades que contiene un asterisco equivalen a ideas de reserva, las realizamos para los casos en que no logremos por alguna razón aplicar alguna de las ideas principales (1 a la 5). Cabe destacar que el UML está realizado respetando solamente las funcionalidades principales (1 a la 5).

Clases:

ChatBot

Rol: Coordinator, Controller

El ChatBot será el encargado de interactuar con los jugadores, imprime comandos, y en base a la respuesta del usuario, realizará distintas actividades.

Menu

Rol: Information Holder

Contendrá cadenas, que corresponderán a los menús que aparecerán en la pantalla de los usuarios durante distintos momentos.

GameSummary

Rol: Information Holder

Al final de cada partida, se guardará un resumen de esta, dicho resumen será una string. El creator es la clase GameSummariesRegister, la cual se detallará más adelante.

GameSummariesRegister

Rol: Information Holder, Structurer

Se encarga de contener todos los resúmenes de juegos de un usuario en específico. Se crea automáticamente cuando se crea un nuevo usuario.

Es una lista de GameSummary.

Gracias a esta clase y la anterior, se logrará llevar un registro de las partidas jugadas (función extra-1).

User

Rol: Information Holder, Structurer

Cada vez que alguien quiere jugar, debe crearse un usuario. Contiene determinados datos, y entre ellos un registro de los resúmenes de las partidas jugadas por este. Al crear al usuario, se crea el registro de resúmenes de partidas (vacío). Los usuarios se encuentran y son creados en UserRegister, que se detallará más adelante.

UserRegister

Rol: Information Holder, Structurer

Es un registro, el cual contiene, crea y elimina a los usuarios. Funciona como si fuera la base de datos principal.

<abtract> Game

Rol: Service Provider

Game es una clase base, en la cual derivaran los modos de juegos planteados en la funcionalidad extra 3 y 5. Aquí se encontrará la lógica necesaria para realizar el juego.

Se utiliza el método DoGame para iniciar el juego, el cual una vez invocado creara dos player asociados a los usuarios ingresados, los players se aclaran más adelante. Llevará el juego adelante y contendrá un while que verifique que el juega continue hasta que los barcos de uno de los dos jugadores sean eliminados.

Attack es un método que se va a encargar de realizar los disparos entre los usuarios.

ShipHability es el método encargado de invocar las habilidades especiales de los barcos, cumpliendo el requisito extra 2.

SaveResume guarda el resumen, accede a UserRegister. Utilizando el usuario que Game ya posee, accede al usuario. Y desde el usuario accede al atributo GameSummariesRegister y crea un nuevo GameSummary.

Contiene la lógica de las WildCard. Que son parte de la función extra 4 sobre los comodines.

BasicGame: Game

Rol: Service Provider

Es subclase de Game, se caracteriza por tener un tablero de tamaño original.

LongGame: Game

Rol: Service Provider

Es subclase de Game, se caracteriza por tener un tablero de mayor tamaño al original.

QuickGame: Game

Rol: Service Provider

Es subclase de Game, se caracteriza por tener un tablero de menor tamaño al original.

Nota: Cabe aclarar que los tamaños de los tableros son predeterminados, y no se pueden modificar, solamente seleccionar.

TimeLimitedGame: Game

Rol: Service Provider

Es subclase de Game, se caracteriza por tener un tiempo límite para realizar los disparos.

Player

Rol: Information Holder

Cuando la partida inicia, se crea un player por cada usuario. La función de este es la de tener una instancia apartado (al usuario) que sea utilizada explícitamente para el juego, teniendo los conocimientos necesarios para el mismo.

Lo asociamos a User a través de la id.

El atributo WildCard solamente se utiliza para llevar el registro de que Wildcard ya se utilizó o no.

Se crea con dos tableros (Board), que se utilizará para colocar sus naves, y otro que llevará el registro de los disparos efectuados hacia el tablero del enemigo. La clase Board de explicará más adelante.

Board

Rol: Information Holder, Structurer

Es el tablero sobre el cuál se jugará, para cada jugador se crearán dos instancias, una para el tablero con las naves y otra para el registro de disparos.

Lleva el registro de barcos que posee.

Crea los barcos (Ship) y los posiciona. Más adelante se hablará de la clase Ship.

Contiene un método que verifica si un impacto acertó o no en un barco.

<abstract> Ship

Rol: Information Holder

Es una superclase, de la cual derivaran los barcos. Se decidió crear, ya que cada barco va a tener una habilidad que lo identifica. Y se va a ver reflejada en el método Hability el cual en esta clase es un método abstracto.

Todavía estamos analizando si la clase debería ser a su vez Service Provider, ya que por el momento estamos planeando que cada barco tenga la lógica de su habilidad.

Carrier: Ship

Rol: Information Holder

Subclase de Ship

Battleship: Ship

Rol: Information Holder

Subclase de Ship

Cruiser: Ship

Rol: Information Holder

Subclase de Ship

Submarine: Ship

Rol: Information Holder

Subclase de Ship

Destroyer: Ship

Rol: Information Holder

Subclase de Ship

<interface> IPrint

Rol: Interfacer

Contiene las operaciones de imprimir en el medio que sus subtipos lo implementen. Creamos una interfaz para que un futuro se pueda agregar distintos medios sin necesidad de modificar la clase.

Por ahora, se imprime los menús, los tableros y el mensaje de si un disparo acertó o no en un barco.

ConsolePrint: IPrint

Rol: Interfacer

Implementa la interfaz IPrint, y se dedica estrictamente a imprimir en consola.