Informe del Bot

Segunda entrega

Ingeniería en informática

Programación 2

Grupo 3

Integrantes del grupo:

Benjamín Rielli

Federico Ferreira

Juan Cabrera

Nicolas Abreu

Docentes: María Parapar, Matías Olave

Fecha de informe 20/06/2022

Contenido

1. Decisiones Principales

UML

En cuanto al UML, y con el objetivo de facilitar la lectura de este, tomamos una serie de decisiones sobre la manera en la que representamos distintas clases. La clase “Coord” no esta conectada con sus colaboradores en el diagrama ya que colabora con una gran cantidad de clases, las clases con las que coord colabora son:

Barco, Tablero, ControladorJuego, CoordenadasNoAlineadas, LeerCoordenadas y Jugador.

La otra clase que no representamos en el diagrama dada su presencia en varias otras clases es Ident, clase simplemente utilizada para simular la función de Id presente en telegram, esta clase colabora con las siguientes clases:

ControladorJuego, JugadorIncorrecto, HistóricoEstadisticas, Jugada, Jugador, Usuario y JuegoConsola.

Otra libertad que nos tomamos en el diagrama UML fue el cómo representamos los handlers, ya que estos usan distintas partes de la lógica de todo el programa decidimos mostrar como se conectan entre ellos, pero se puede inferir sus colaboraciones por sus nombres. Todos los handlers utilizan la clase message, esto no es representado en el diagrama por simplicidad.

Tuvimos que generar dos archivos distintos para los UML ya que la pagina que utilizamos tenia un limite de carga que fue superado,

Código

En cuanto a código, preferimos justificar algunos patrones en el informe en vez de en el código por practicidad, este es el caso de los handlers, todos cumplen con SRP,

tienen la única responsabilidad de responder a cierto mensaje, cumplen con LSP, ya que todos son intercambiables con todos sin representar ningún tipo de problema, también cumplen con Chain of Responsability y con Polimorfismo, responden a la misma llamada (Un mensaje de un Usuario) de manera distinta dependiendo de la naturaleza de dicha llamada.

Otra decisión que tomamos fue hacer la clase Jugadas para facilitar la creación de los bots, generando estas jugadas vamos a poder comunicar por una interfaz única con el controlador de juego.

Los casos de prueba de los handlers no fueron implementados en esta entrega dada la caracterización opcional de los mismos, dichos casos de testeo serán implementados para la entrega final.

2. Cambios

Como primer cambio decidimos separar el usuario del jugador, teniendo en cuenta que cada vez que se crea una nueva partida un nuevo objeto jugador es creado utilizando la id del usuario, nos pareció más efectivo remover algunos atributos de jugador y dárselos a la clase usuario, una clase cuyos datos, serán guardados (utilizando persistencia), cada usuario tendrá un perfil predeterminado con su nombre y estadísticas.

Otro cambio importante es que en esta entrega implementamos los handlers y creamos la máquina de estados necesaria para utilizarlos. Estos fueron implementados pensando en cada posible decisión que un usuario podría tomar en el Bot y creando un handler para la misma.

Un cambio notable fue a la funcionalidad de las excepciones, en un principio y como fueron pensadas para la primera entrega estas no podían decirnos *exactamente*

donde se encontraba el error, por ejemplo, para colocar un barco el programa requiere que el usuario ingrese dos coordenadas, en el caso de que una de ellas no fuera valida, en nuestra idea previa el usuario solamente obtendría un mensaje de “Coordenadas invalidas”, luego de un poco de pienso determinamos que sería mejor que el usuario sepa exactamente en qué coordenada cometió el error, en el código actual el mensaje seria “Coordenada <coord.> invalida”

En cuanto al funcionamiento del juego, eliminamos el concepto de “Turnos” de la lógica, en vez de mantener un control directo de quien es el turno, simplemente lo manejamos en la lógica del juego, en la clase ControladorJuego, si un jugador trata de jugar fuera de su turno este recibirá una excepción desde dicha clase que le indica que no es su turno de jugar.

3. Nuevas Funcionalidades

Implementamos una serie de nuevas funcionalidades para facilitar el funcionamiento del Bot, para cumplir con algunos patrones y para implementar las nuevas funcionalidades prometidas. Estas nuevas funciones son: Persistencia, Handlers, Usuarios, Maquina de Estados, Temporizador de Turnos, Excepciones.

Persistencia

La persistencia fue implementada en colaboración de Juan y Nicolas, esta consiste en una clase encargada de guardar todas las instancias existentes en un archivo Json, cuando el programa se inicia estos archivos Json pueden ser levantados por el código para restaurar la información de las instancias guardadas en los mismos.

Handlers

Los handlers fueron implementados por Federico y ligeramente modificados por Juan, estos implementan el patrón Chain of Responsability y utilizan código de la

lógica para actuar acorde a los mensajes del usuario recibidos, devolviendo respuestas para indicarle al usuario como proceder.

Como indicamos en la sección de cambios, decidimos separar al Usuario del Jugador por motivos de practicidad, esta clase fue implementada por Nicolas y ligeramente modificada por Federico, en esta clase mantenemos los datos permanentes del Usuario como su nombre y estadísticas.

Reloj

El temporizador de turnos, o reloj, fue una de las funciones extra prometidas por nuestro equipo, fue implementada por Juan y es utilizada para controlar cuanto tiempo demora un jugador en efectuar su turno, dándole un tiempo límite y otorgándole la victoria al oponente en caso de agotarse.

Otra adición fue la clase Ident, esta clase fue implementada por Juan y es una adición temporal para simular las Ids de los usuarios de telegram mientras la conexión no está hecha.

Una serie de excepciones fueron añadidas por Benjamín para contemplar los distintos tipos de errores que un usuario podría cometer y cómo manejarlos, las excepciones añadidas son; BarcoYaExiste, JugadorIncorrecto, RadaresAgotados, BarcosSuperpuestos, CoordenadaFormatoIncorrecto, CoordenadasNoAlineadas, EstadoPartidaIncorrecto.

BarcoYaExiste

Esta excepción detecta cuando el jugador esta tratando de colocar un barco que ya colocó previamente y le indica que ya realizó esa acción.

JugadorIncorrecto

Esta excepción detecta cuando el jugador esta tratando de realizar una acción cuando no es su turno y le indica que es el turno del oponente.

RadaresAgotados

Esta excepción detecta cuando el jugador esta tratando de usar un radar cuando ya no tiene más de este recurso y le avisa que sus radares han sido agotados.

BarcosSuperpuestos

Esta excepción detecta cuando el jugador esta tratando de colocar un barco superpuesto a otro y le indica que esta acción no es posible.

CoordenadaFormatoIncorrecto

Esta excepción detecta cuando el jugador manda un set de coordenadas en un formato incorrecto, revisa cual de las coordenadas en el set es incorrecta y devuelve al usuario la coordenada que escribió mal.

El return del atributo “Razón” es un enum que esta incluido en la misma clase.

CoordenadasNoAlineadas

Esta excepción detecta cuando el jugador manda un set de coordenadas para colocar un barco y estas no están alineadas, le dice al usuario que no es posible colocar un barco con dicha configuración.

EstadoDePartidaIncorrecto

Esta excepción detecta cuando el jugador intenta realizar una acción en un momento no adecuado y le indica que el estado de la partida no es el correcto para esa acción.

4. Futuras Funcionalidades

Bot Oponente.

Una futura funcionalidad que vamos a agregar es la capacidad de jugar contra un Bot automático en vez de contra otra persona, esta funcionalidad no fue implementada en esta entrega en parte por falta de tiempo y en parte por la duda de que tanto tendríamos que modificarla una vez implementada la conexión con telegram.

Mejora a visuales del tablero.

En esta entrega, al todo funcionar en consola, la implementación de como se imprime el tablero es, aunque funcional, bastante rustica. Para la próxima entrega tenemos planeado mejorar como se ven estas impresiones del tablero, ya sea con caracteres mas bonitos o con imágenes, dependiendo de cuánto tiempo tengamos.

A esta altura sentimos que casi toda funcionalidad pedida ha sido implementada, futuros cambios incluirán la conexión con telegram, los ítems mencionados arriba y cualquier cambio pedido por los docentes en la corrección.