BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Ingeniería del Software

Sprint 1 Sunken
2021-2022
...
Titulación:
Grado en Informática de Gestión y Sistemas de Información
...
2º Curso(2º Cuatrimestre) ...
`https://github.com/Strawberryai/Sunken`

Alan García, Álvaro Díez-Andino,
Danel Alonso, Gorka Crespo, Adrián López

31 de marzo de 2022

Índice

1. Informe MVC

El MVC es el protocolo que se encarga de los observers y los observables que utilizamos para actualizar los visuales de nuestro juego.

En el aspecto gráfico, se crea una ventana principal que se encargará de visualizar el estado actual de la aplicación. La vista se comunica con el modelo (el juego) a través del controlador (*ControladorVentanaPrincipal*) que se encarga de gestionar todas las acciones relacionadas con el mouse y el selector de orientación ya que implementa MouseListener e ItemListener. Además, en la vista se crean dos tableros de 10x10 cada uno, el panel de barcos donde estarán los botones para seleccionar el tipo de barco, el panel de misiles (de momento solo bomba) y además el seleccionador de dirección para colocar los barcos (Norte,Sur,Este,Oeste).

Al pasar por encima de alguna casilla con el cursor, estas cambiarán de color mientras el cursor se encuentre en su area. El controlador gestiona la posición del mouse y le indica a la casilla correspondiente que cambie su color al color del hover.

Cuando se selecciona un barco, misil, orientación... Se actualiza el registro del controlador con la información que se ha seleccionado para posteriormente enviarla a Gestor del Juego un formulario con todos los datos necesarios.

Al hacer 'click' en una casilla se produce la llamada al Gestor del Juego y se envía el formulario previamente mencionado:*notificarCasillaPresionada()*.

Una vez el Gestor del Juego ha realizado los cambios en el estado de la aplicación se actualizan los estados de las casillas en la vista realizando una llamada a *setChanged()* y *notifyAll()* del patron Observer: el Gestor es el Observable y las casillas de la vista son los Observers.

2. Diagramas

En esta sección incluimos todos los diagramas, tanto de secuencia como el de clases del modelo, que se han solicitado para esta entrega.

2.1. Diagrama de clases del modelo

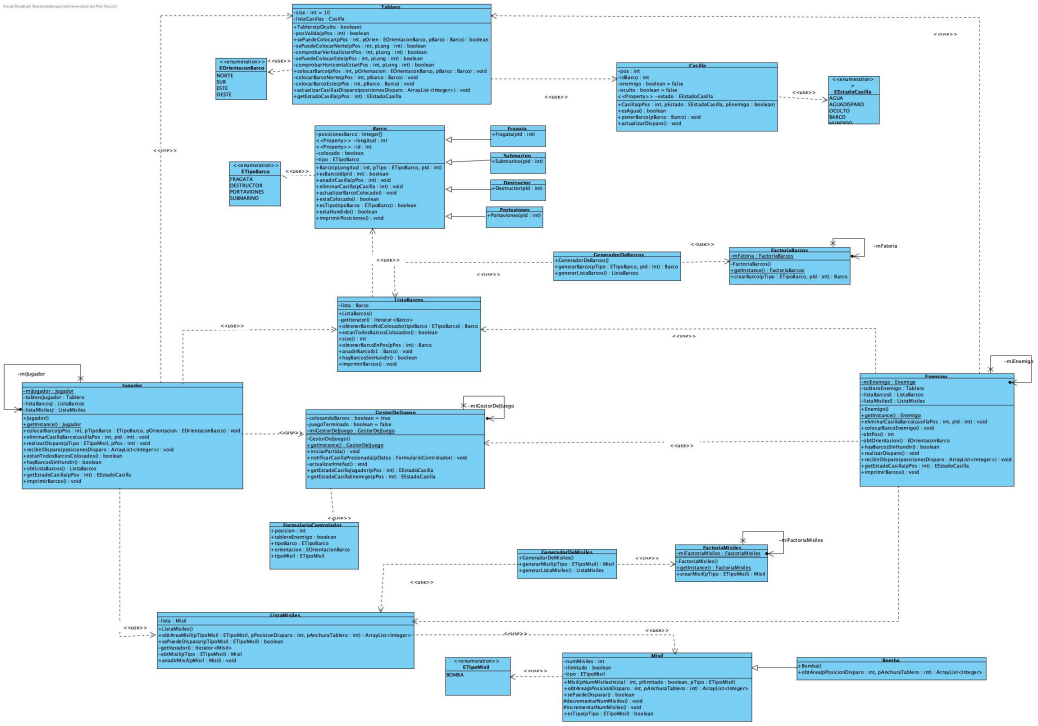

Figura 1: Diagrama de clases

2.2. Diagramas de secuencia de disparo

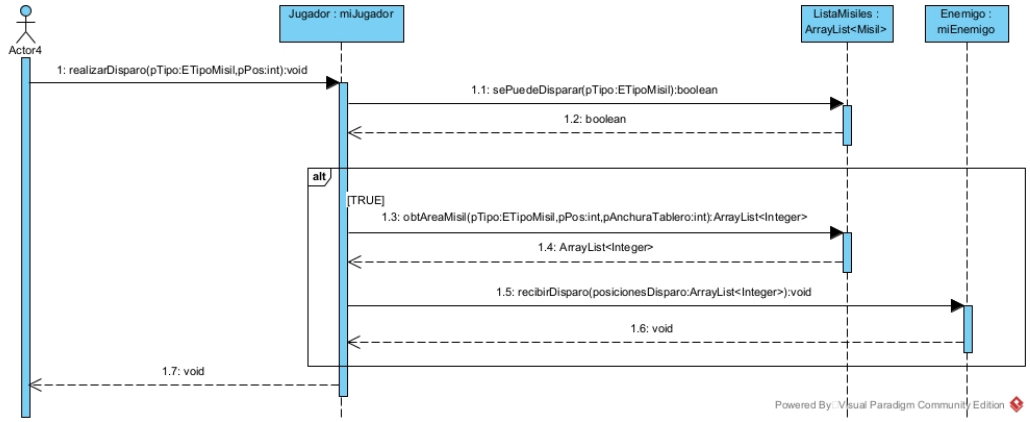

Figura 2: Realizar disparo jugador

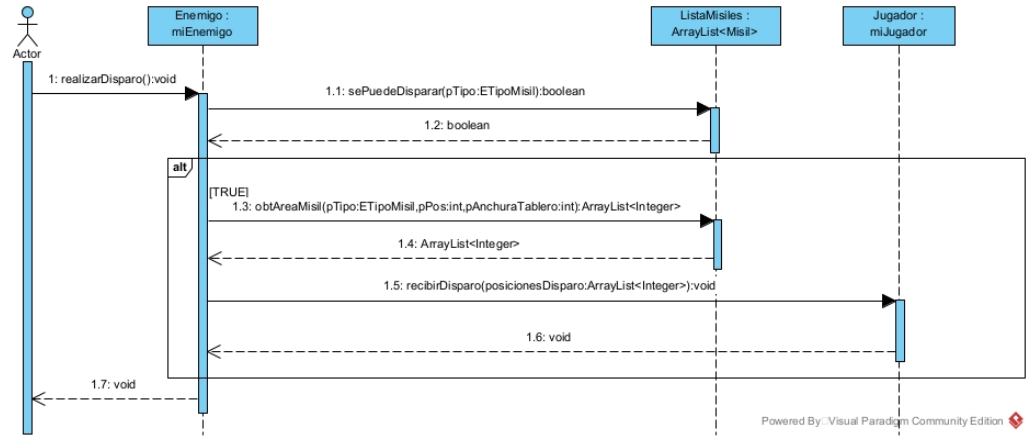

Figura 3: Realizar disparo enemigo

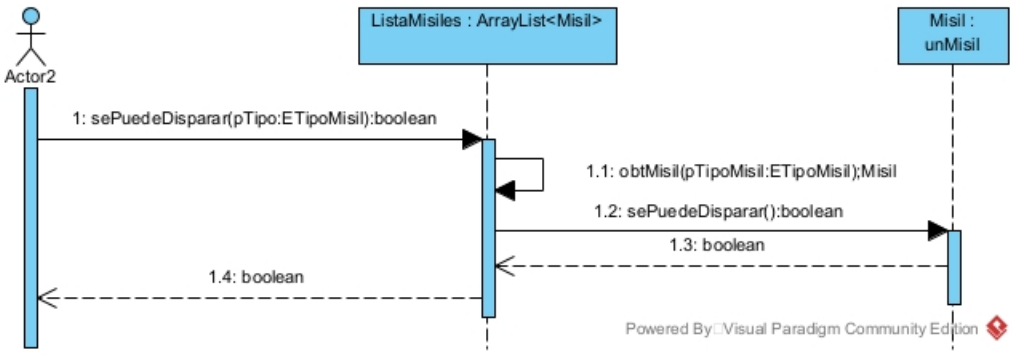

Figura 4: Se puede disparar de ListaMisiles

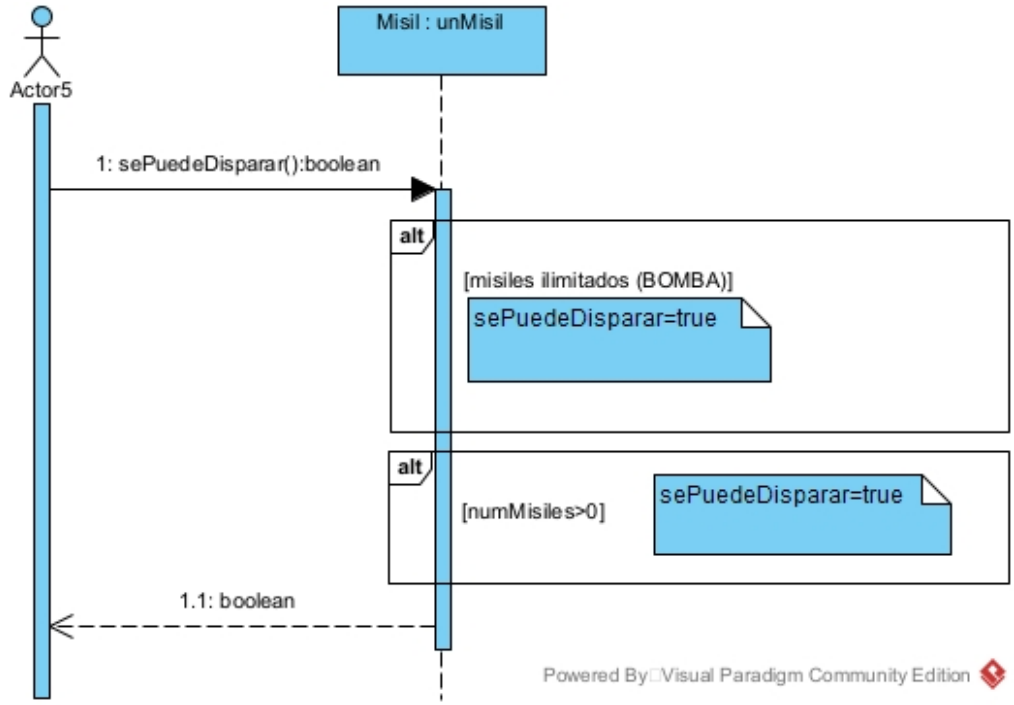

Figura 5: Se puede disparar de Misil

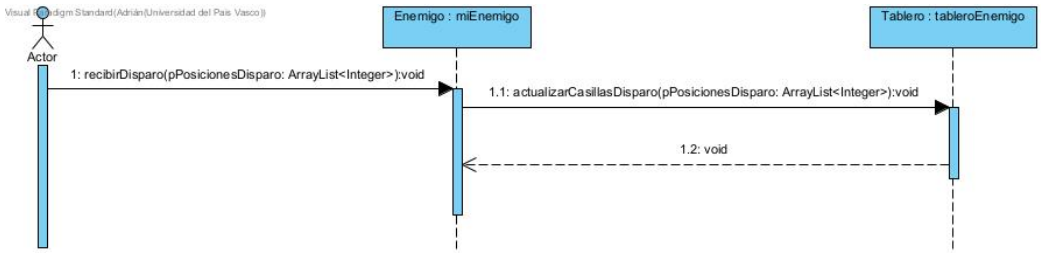

Figura 6: Recibir disparo enemigo

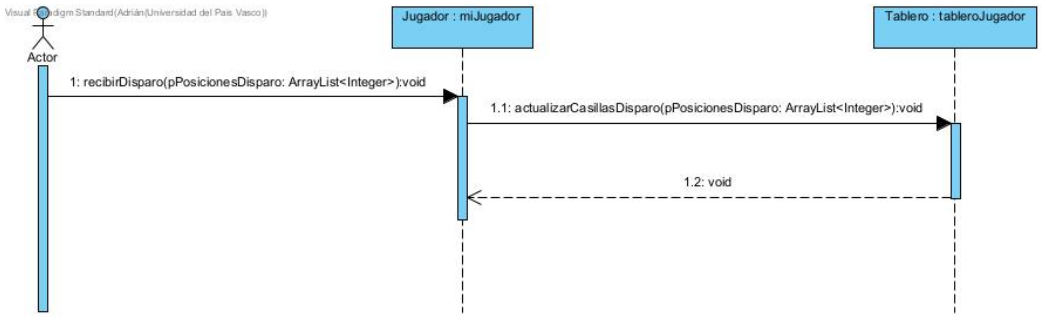

Figura 7: Recibir disparo jugador

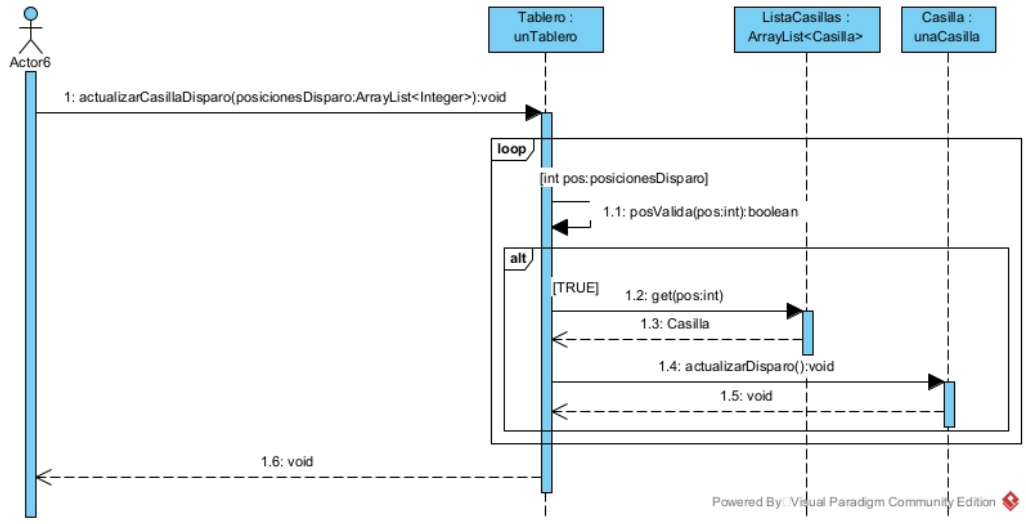

Figura 8: Actualizar casilla disparo

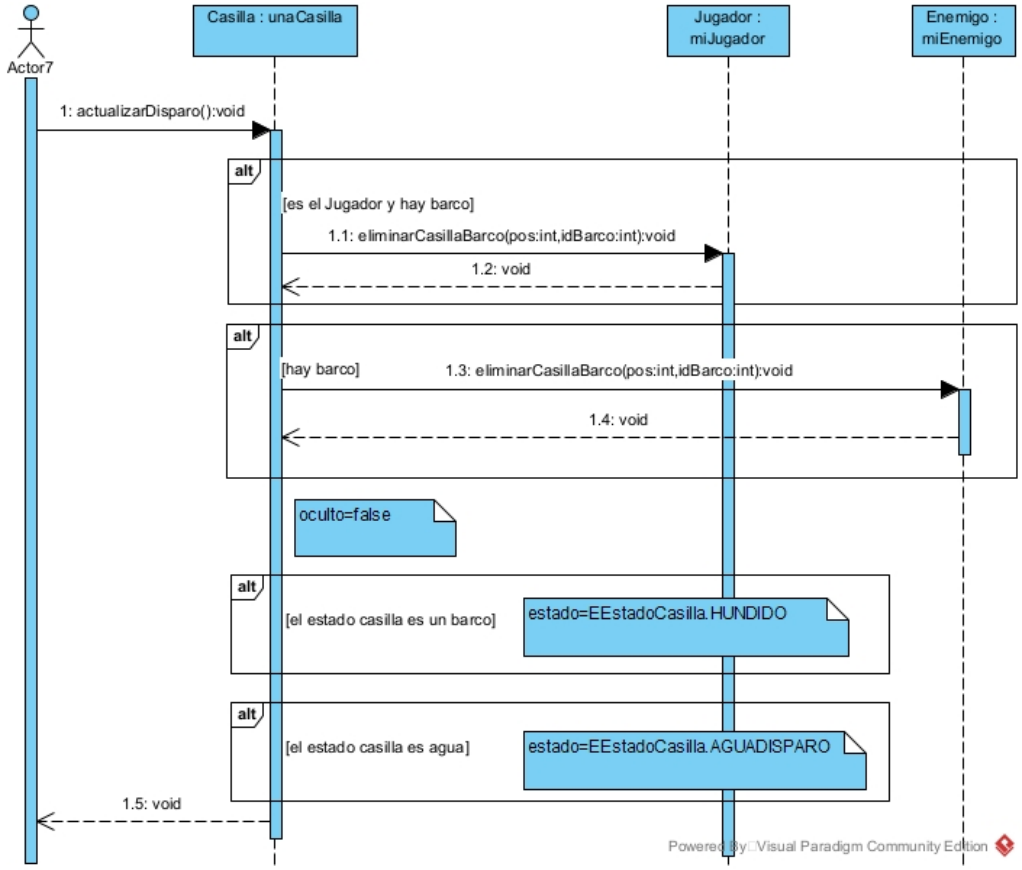

Figura 9: Actualizar casilla disparo

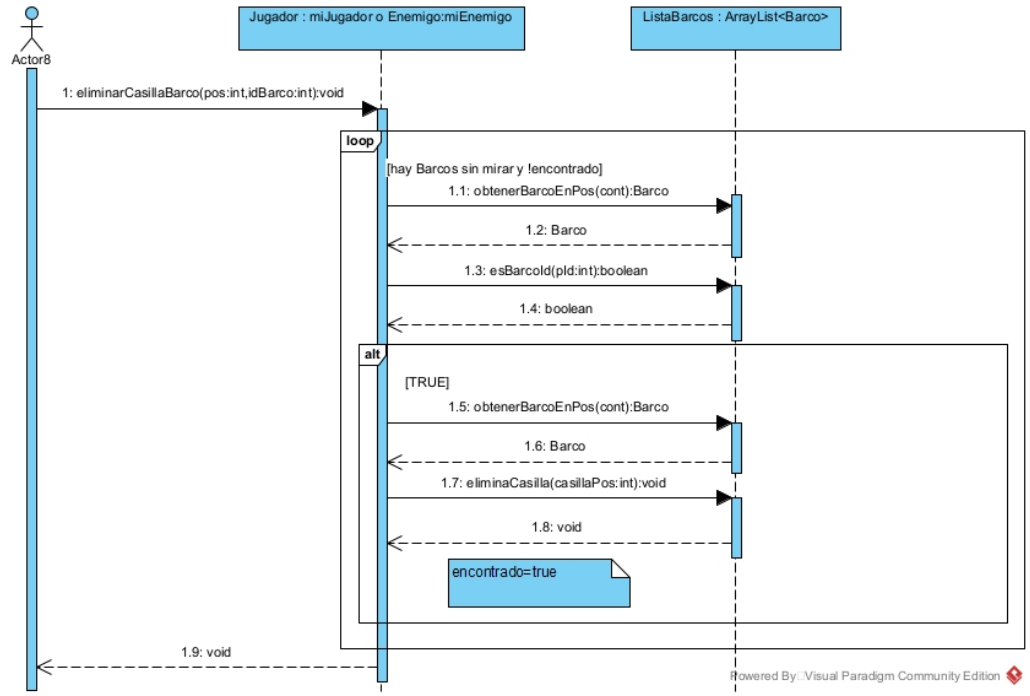

Figura 10: Eliminar casilla barco

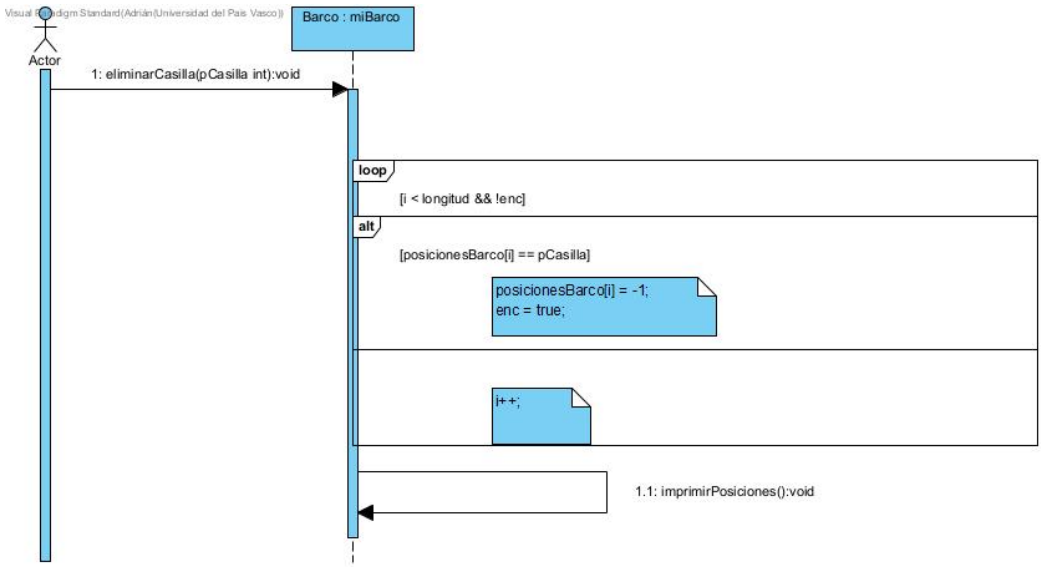

Figura 11: Eliminar casilla

3. Actas de reunión

Las actas de reunión están incluidas como un pdf adjunto.

Primera Reunión

Asistencia	Adrián Alan Danel Álvaro
Fecha Inicio	@March 11, 2022 10:00 AM
Fecha Fin	@March 11, 2022 11:00 AM
Lugar	Discord
Tags	Virtual

Conclusiones principales

Punto de partida del proyecto. Se elaboró un diagrama muy básico y se discutió cómo realizar ciertas tareas.

Acuerdos tomados

- Realizar el programa principal en primer lugar y luego ocuparse de la interfaz gráfica.
- Tener una MAE GestorDeJuego encargada de asumir el papel principal de modelo en el patrón modelo-vista-controlador.
- Distinguir dos clases: Jugador y Enemigo.
- Cada jugador y enemigo tienen un tablero que a su vez tiene una colección de casillas.
- Cada casilla tiene un estado: AGUA, BARCO, HUNDIDO
- Tanto el enemigo como el jugador tienen una lista inicial de barcos y se encargarán de colocarlos en el tablero.
- Hay varios tipos de barcos que se diferencian en el número de casillas que ocupan.

Tareas asignadas

Pensar en cómo extender el diagrama.

Segunda Reunión

Asistencia	Completa
Fecha Inicio	@March 14, 2022 11:30 AM
Fecha Fin	@March 14, 2022 1:45 PM
Lugar	Aula de estudio Bilbao
Tags	Presencial

Conclusiones principales

En esta reunión hemos continuado con la elaboración del diagrama de clases y hemos realizado un diagrama de secuencia inicial del método colocarBarcos() desde Jugador y Enemigo.

Además, hemos planteado una posición del proyecto en la cual ya comprendemos cuales son las tareas principales a realizar.

Acuerdos tomados

- Continuar trabajo en el diagrama de clases de forma conjunta
- Plantear en la próxima reunión cómo realizar el proyecto de forma colaborativa.

Tareas asignadas

Pensar en cómo extender el diagrama de clases y de secuencias además de explorar cómo realizar el proyecto de manera conjunta.

Tercera Reunión

Asistencia	Completa
Fecha Inicio	@March 15, 2022 5:00 PM
Fecha Fin	@March 15, 2022 7:00 PM
Lugar	Clase
Tags	Presencial

https://github.com/utnfrrojava/eclipse-git-tutorial

Conclusiones principales

En esta reunión hemos continuado con la elaboración del diagrama de clases y hemos diseñado el sistema de misiles y un primer diagrama de secuencias del método recibir disparo. Se ha empezado a estructurar de forma colaborativa, se han contemplado varias opciones como GitHub (en progreso), Visual Paradigm online (descartada) y Notion para realizar las actas de reunión de forma conjunta.

Acuerdos tomados

- Continuar trabajo en el diagrama de clases de forma conjunta
- Finalizar la vinculación con GitHub desde Eclipse e IntelIJ.
- Iniciar la implementación del código.

Tareas asignadas

Investigar como vincular GitHub con Eclipse.

Cuarta Reunión

Asistencia	Completa
Fecha Inicio	@March 16, 2022 10:30 AM
Fecha Fin	@March 16, 2022 1:00 PM
Lugar	Discord
Tags	Virtual

Conclusiones principales

Se ha decidido que eclipse da muchos problemas en cuanto a gestionar proyectos se refiere, luego hemos decidido usar Maven como gestor de proyectos para que se encargue de la configuración del Classpath y el JRE. Además hemos conseguido vincular el GitHub con Eclipse e IntelIJ por lo que ahora somos capaces de realizar commit y push desde estos IDE manteniendo el proyecto actualizado.

Se ha generado el esqueleto de las clases del modelo desarrollado en Visual Paradigm para poder repartirnos la implementación de las clases diseñadas

Acuerdos tomados

- Realizar un fichero LaTex del proceso de vinculación con GitHub.
- Realizar la documentación en GitHub con LaTex.
- Explorar la forma de realizar ficheros LaTex conjuntamente mediante GitHub.
- Repartir e implementar las clases

Tareas asignadas

- Álvaro —> Paquete Misil.
- Danel —> Jugador y Enemigo.

- Alan —> Paquete Barcos.
- Gorka —> ListaBarcos y poner en marcha Intellj con GitHub.

Quinta Reunión

Asistencia	Alan Danel Gorka Álvaro
Fecha Inicio	@March 18, 2022 11:00 AM
Fecha Fin	@March 18, 2022 1:45 PM
Lugar	Aula de estudio
Tags	Presencial

Conclusiones principales

Se ha llevado a cabo la implementación de las clases referentes a barcos, misiles y parte de la de Jugador y Enemigo. También hemos incluido el patrón de diseño Factory con su correspondiente implementación. Finalmente, hemos actualizado el fichero de Visual Paradigm.

Acuerdos tomados

- Arreglar el método colocarBarcos() de la clase Jugador y Enemigo.
- Implementar las clases Casilla, Tablero y GestorDeJuego.
- Realizar una interfaz grafica inicial para visualizar lo que hace.
- Realizar un esquema de como va a ser la interfaz grafica.

Tareas asignadas

- Alan —> Implementar clase Tablero.
- Adrián —> Implementar clase GestorDeJuego.
- Gorka —> Implementar clase Casilla.

- Danel y Álvaro —> Actualizar diagrama de clases.

Sexta Reunión

Asistencia	Completa
Fecha Inicio	@March 21, 2022 11:00 AM
Fecha Fin	@March 21, 2022 1:00 PM
Lugar	Discord
Tags	Virtual

Conclusiones principales

Se han aclarado todos los procesos hechos hasta ahora y se ha propuesto trabajo para la próxima semana. Se han mostrado todos los problemas existentes en la implementación actual, se han corregido algunos y se ha distribuido el trabajo.

Acuerdos tomados

Cosas a añadir:

1. ID de barcos en casilla y en barco para identificar qué barco ha recibido un disparo
2. Implementar todos los métodos relacionados con realizar disparo
3. Arreglar gestorDeJuego y otros métodos relacionados.
4. Implementar colocarBarco() en tablero.
5. Implementar una vista inicial de los tableros.
6. Crear un controlador de la ventana principal y agregar los botones necesarios para testear lo implementado.

Tareas asignadas

- Gorka → Implementación de código restante y arreglos.

- Adrián → Implementación de código restante y arreglos.
- Álvaro → Implementación de código restante y arreglos.
- Danel → módulos de una vista inicial: Botones y ControladorVentanaPrincipal(MAE).
- Alan → módulos de una vista inicial: Tableros.

Séptima Reunión

Asistencia	Completa
Fecha Inicio	@March 26, 2022 11:20 AM
Fecha Fin	@March 26, 2022 6:00 PM
Lugar	Discord
Tags	Virtual

Conclusiones principales

Se finaliza la elaboración del método colocar barco tanto en el modelo como en la vista incluyendo todas las direcciones.

Acuerdos tomados

Cosas a añadir:

1. Corregir los disparos tanto de enemigo como de jugador.
2. Terminar la versión preliminar para el primer sprint.
3. Diagrama de clases final del primer sprint
4. Diagrama de secuencias del método disparo
5. Manual MVC

Tareas asignadas

Alan→ Corregir los disparos, terminar la versión preliminar y diagrama de clases

Adrian→ Diagrama de secuencia de disparo

Gorka→ Manual MVC

Danel→ Diagrama de secuencia de disparo

Alvaro→ Manual MVC

Octava Reunión

Asistencia	Completa
Fecha Inicio	
Fecha Fin	
Lugar	Clase
Tags	Presencial

- Preparar todo para entregar el primer sprint

Conclusiones principales

Avanzar con la entrega del primer sprint puesto que la versión preliminar ya ha sido finalizada.

Acuerdos tomados

Cosas a añadir:

1. Diagrama de clases final del primer sprint
2. Diagrama de secuencias del método disparo
3. Manual MVC

Tareas asignadas

Todos→ Pensar en nuevos patrones y preparar el entregable